Addis Ababa University College of Natural Sciences

Department of Computer Science

Bidirectional Long-Short Term Memory Based Text to Speech Synthesis for Amharic Language

Mahlet Awel Temam

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

December 7, 2020

Addis Ababa University

College of Natural Sciences

Department of Computer Science

Mahlet Awel Temam

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by *Mahlet Awel Temam*, titled: *Bidirectional Long-Short Term Memory Based Text to Speech Synthesis for Amharic language* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signature of the Board of Examiners for Approval

Name	Signature	Date
Advisor: Yaregal Assabie (PhD)	_____	_____
Examiner: Mesfin Kifle (PhD)	_____	_____
Examiner: Ayalew Belay (PhD)	_____	_____

Abstract

Text-to-speech (TTS) synthesis is the automatic conversion of written text to spoken language. TTS systems show an imperative character in natural human-computer interaction. The aim of this work is to develop a bidirectional long-short term based TTS system for the Amharic Language. The system has two phases, the training and synthesis phases. In the training phase, first the text normalization is done and then from the normalized text linguistic features are extracted by using festival tool and the extracted features are used as input for the BLSTM based duration model. Then after that, duration model training is done and the model adds duration information on the extracted linguistic features and feeds for the BLSTM based acoustic model. The world vocoder extracts many acoustic frames composed of features which describe the signal in a more convenient way and used as an input for the acoustic model. Aco5ustic model training is done to map the input linguistic features and the associated duration features into acoustic features. We have prepared 600 speech their corresponding text transcription from Amharic audio bible by a male speaker. For this work the open source merlin speech synthesis toolkit, festival speech synthesis tool as a frontend and world vocoder are used. We have also prepared a pronunciation dictionary (lexicon) of 2500 words, phone set, letter to sound rule and question file set for frontend text processing based on the phonetic structure of Amharic language. In order to test the performance of the system we have performed subjective and objective evaluation. The evaluation with a listening test by 10 volunteers gave a score in MOS of 3.8 for intelligibility and 3.9 for naturalness to our BLSTM model and 3.65 for intelligibility and 3.7 for naturalness to our DNN model and MCD of BLSTM and DNN is 4.68 and 4.7 respectively.

Keywords: Deep learning, Recurrent Neural Networks, long-short term memory, duration model, acoustic model, vocoder, linguistic features, acoustic features.

Dedicated to

My Family

Acknowledgements

I am humbly grateful to my Lord for guiding me and helping me all the way through. Everything is done by the will of God, he gives me strength and health to finish my work. Then I would like to thanks my advisor Dr. Yaregal Assabie, for having patience till I have finished and for his guidance to help me.

I would like to thank my family and especially my mom they always support me on everything as much as they can. I would like to sincerely thank all my friends, the people who were volunteer to do the experiment and classmates for their support, help, and inspiration during the research. It is difficult to mention all person's name who supported me during the completion of the thesis.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

SPSS Statistical Parametric Speech Synthesis

DL Deep Learning

RNN Recurrent Neural Network

DNN Deep Neural Network

LSTM Long-Short Term Memory

BLSTM Bidirectional Long-Short Term Memory

MLPG Maximum Likelihood Parameter Generation

LTS Letter to Sound

TTS Text to Speech

HMM Hidden Markov Model

HTK Hidden Markov Model Tool Kit

HTS Hidden Markov Based speech synthesis tool kit

SPTK Speech Signal Processing Toolkit

MCD Mel Cepstral Distortion

MOS Mean Opinion Score

IPA International Phonetic Alphabet

SAMPA Speech Assessment Methods Phonetic Alphabet

NSWs Non Standard words

F0 Fundamental Frequency

Chapter One: Introduction

Language technologies are information technologies that are specified for dealing with the most compound information medium human language. Human language happens in spoken and written form. While speech is the eldest and best natural mode of language communication, complex information and most of human knowledge is preserved and communicated in written texts. Speech and text technologies process language in these two manners of realization [1].

Text-to-Speech system is a system that can convert a given text into speech signals. The aim of an ideal text to speech system is to be able to process any text that a human can read. Text-to-speech synthesis systems are an important component of modern human–machine communication systems [2].

There are different techniques that are used to develop a text to speech system. The first one is formant based which is based on the source-filter-model of speech. The second one is concatenative based which is based on breaking down the spoken sentence into different words and syllables within it there are different techniques which are Unit selection based, Diphone based, Domain Specific. Articulatory Synthesis is the third one which is based on the modeling of the human speech production system. The fourth one is hidden markov model (HMM) based speech synthesis is also known as a statistical parametric synthesis [3].

Deep learning is a set of learning approaches trying to model data with difficult architectures merging dissimilar non-linear transformations. Deep learning is developing area of machine learning (ML) research. It encompasses numerous hidden layers of artificial neural networks. The deep learning practice put on model abstractions of high level in big databases and nonlinear transformations. Deep learning entails an abstract layer examination and hierarchical methods. Nevertheless, it can be practical in numerous real-life applications [4]. Deep learning models are decent at learning features from data. Deep learning based text to speech models can be built without prior knowledge of a language when generating speech. Deep text to speech models can be easily built, which require carefully designed features, require (speech, text) pair data [5].

Deep neural network is one of the family of deep learning architecture and it is used by so many researchers to do the text to speech synthesis. The main limitation of the DNN-based acoustic modeling is that the sequential nature of speech is unnoticed. Even though, indeed there are correlations between consecutive frames in speech data, the DNN-based approach assumes that each frame is independent. It is necessary to incorporate the sequential nature of speech data to the acoustic model itself. DNNs do not naturally model the temporal structure in speech and text. Using DNNs as acoustic models will restrict usage of context to a few phones in past and future and lead to discontinuities in the predicted parameters [6].

Recurrent Neural Networks (RNNs) is the also the family of DL that are well-suited for pattern classification tasks whose inputs and outputs are sequences, for example tasks such as speech recognition, speech synthesis, named-entity recognition, language modelling, and machine translation [6].

RNN provides sophisticated method to model speech-like sequential data that represents associations among bordering frames. It can also practice all the accessible input features to forecast output features at each frame. Particularly, the RNN model is different from the DNN since it operates not only on inputs but also on network internal states that are updated as a function of the entire input history. In this case, the recurrent connections are able to map and remember information in the acoustic sequence, which is important for speech signal processing to enhance prediction outputs [6].

Training RNN incorporates backpropagation. Consequently, parameters are shared through all time steps, the gradient at every output relies upon now no longer best at the contemporary time step, but too on the earlier ones and vanishing gradients and exploding gradients occur.

Long short-term memory networks (LSTM) are a class of recurrent networks composed of units with a certain structure to manage better with the vanishing gradient problems during training of recurrent neural network and maintain potential long-distance dependencies [7].

1.1 Motivation

Amharic language is national language of Ethiopia. Amharic language is a Semitic language which is developed in the horn of Africa and it also spoken in somewhere the world. It is the second-most spoken Semitic language in the world (25 million speaker), next to Arabic [8]. There are many people who have learning disabilities, with visual impairment, who have literacy difficulties, speak the language but do not read it, who are multitask, and so on. These people need text to speech system for different purpose but they can't utilize it since it is difficult to find and use text to speech products easily. This has motivated us to work on Amharic text to speech using bidirectional long short term memory.

1.2 Statement of the Problem

A speech synthesis system that generates natural sounding and intelligible speech is essential for many application areas. In addition to this a speech synthesizer to be useful for any one, it should generate speech with appropriate melody and prosody. A number of researches on the development of text to speech system have been conducted for different languages such as English [9], Arabic [10], Afaan Oromo [11]. Different research works have been carried out to develop Amharic text to speech [50-59] using different techniques. Deep neural networks are used by many researchers to develop text to speech synthesis for different languages, since they have better ability to capture the dependencies across the features. However, there is the absence of the ability to capture the relations that are spread across time. Therefore, the text and acoustic features are mapped frame-wise, assuming that each frame is independent of the other [12].

Therefore, bidirectional long short term memory is found to be very effective to comprise contextual constraints, it was used in [13] to articulate TTS as a sequence to sequence mapping problem that is to map a sequence of linguistic features to the conforming sequence of acoustic features. In [14] LSTM with a recurrent output layer was recommended to incorporate contextual constraints. In [15] LSTM and gated re-current unit (GRU) based RNNs are joined with mixture density model to forecast a sequence of probability density functions and the works indicates that the recurrent neural network provides promising results over the deep neural network.

Thus, we hypothesize that bidirectional long short term memory to overcome the problems for the development of Amharic text to speech synthesis system and generate a natural and intelligible speech for Amharic language.

1.3 Objectives

General objective

The general objective is to develop bidirectional long-short term memory based text to speech synthesis for Amharic language.

Specific objective

The specific objectives which are needed to realize the general objective are:

- To conduct the literature review on different speech synthesizer techniques regarding their advantage and disadvantage.
- To conduct the literature review concerning on Amharic speech synthesizer which are done till currently.
- To collect a corpus of speech recording and their corresponding text transcriptions for the language which are used for the training and testing purpose.
- To study the characteristics of Amharic linguistic features.
- To develop the Amharic text to speech synthesizer using bidirectional long short-term memory.
- To test the developed speech synthesizer.
- To judge the proposed Amharic speech synthesizer.

1.4 Application of Result

Text-to-speech technology offers numerous advantages on behalf of content owners and publishers in addition to their content consumers. Content consumers can be, online learner's mobile application users, website visitors and more. Text to speech lets content owners to reply to the different wants and requirements of each user in case of how they interact with the content.

For people with visual impairment – Text to speech can be a precise valuable tool for the moderately visually impaired. Even for people with the visual ability to read, the process can

frequently cause too much stress to be of any use or pleasure. With text to speech, people with visual impairment can take in entire way of content in well-being instead of stress.

For people with learning disabilities – Reading large amounts of text in some people have a trouble due to dyslexia and other learning disabilities. Therefore, giving them an easier possibility for undergoing website content is a countless way to involve them.

For people who have literacy difficulties – Often getting frustrated for trying to browse the internet is happened in some people have basic literary levels. By giving them a choice to hear the text instead of reading it, they can acquire valued information in a way that is easier for them.

Educational application: Teaching 24 hours a day and 365 days a year can be practiced by a computer with a capability of speech synthesizer. The machine can be programmed for many responsibilities like spelling and pronunciation education for diverse languages.

People who speak the language but do not read it: Several people may have difficulty reading in a second language to speak and understand the native language effectively.

People who multitask: A tiring life sometimes means that people do not have time to do the reading they would like to do online. Taking a chance to listen to the content instead of reading it permits them to do somewhat else thing at the same time.

1.5 Methods

Literature review

A literature review is conducted on text to speech system area to have a clear idea about the work and in order to have an understanding on what kind of techniques we have to use for our speech synthesis and the techniques those are used by different researchers for Amharic speech synthesis and for other different languages. To pick out the advantage and disadvantage of the existing speech synthesis techniques literature review is done. Deep learning method for speech synthesis is also reviewed and different types of deep learning based techniques and their advantage and limitation is also reviewed in order to use the better technique from those through ours work.

Data Collection

The main objective of the data collection activity is to prepare a dataset which can be used for the training and testing purpose to develop the desired system a speech data of Amharic language were collected. The collection of the data is parallel corpus (speech and text). To begin our data collection, we have selected a portion of an Amharic audio bible which is found on YouTube and then we have selected a part of speech with a good recording condition to reduce the availability of noise. The segmentation of speech is done manually to select the speech without noise. Then the corresponding text transcription of the recorded speech is also prepared manually.

Prototype development

We have developed a prototype for Amharic language text to speech synthesis that converts both standard and non-standard words (NSWs) in to speech using bidirectional long-short term memory based recurrent neural network.

Evaluation

The quality of a TTS-system is judged using two evaluation methods those are subjective and objective evaluation. The subjective evaluation is conducted in terms of intelligibility and naturalness describes the quality of the audio generated. Naturalness is the ability for the reader to perceive the original message and the clarity of the audio is considered within the intelligibility criterion. Naturalness describes the quality of the speech generated. Linguistic features such as pronunciation, timing and emotion all falls under naturalness. The evaluation of the performance of the proposed system were done using subjective in terms of naturalness and intelligibility evaluation. Mean opinion square (MOS) is used to conduct the subjective evaluation. Evaluation is also performed for different neural network configurations to perform comparisons between them. Objective evaluation is also conducted by using melcepstral distortion (MCD).

Development tools

To develop the text to speech synthesis we have used different development tools for different purpose.

HTS Toolkit (HTS 2.0): is used for building for speaker dependent and speaker adaptive synthesizers and more or less voices for the Festival Speech Synthesis System.

Merlin

The Merlin toolkit for neural network based speech synthesis [16] is an integrated and extensible toolkit for training synthetic voice. Merlin is an open source neural network based speech synthesis system for statistical parametric speech synthesis. Merlin allows the user to configure the number and type of layers in the neural network voice model, and also allows for different choice of vocoder.

World

WORLD vocoder [17] is used for acoustic feature extraction and vocoding, to generate speech signal.

Festival Speech Synthesis System

Many TTS systems consist of two components: front end (text normalization, grapheme-to phoneme conversion, linguistic feature extraction) and back end (waveform generation). The BLSTM based speech synthesis performs the back-end part only. So, it is important to combine the BLSTM based speech synthesis module with a front end module in other software packages to build a complete TTS system. The Festival Speech Synthesis System is mostly used for general multilingual speech synthesis system in C and Scheme [18].

1.6 Scope and Limitation

This research focuses on single speaker Amharic text to speech synthesis. It doesn't employ speaker adaptation and it is speaker dependent.

1.7 Organization of the Thesis

This thesis is organized as follows. Chapter 2 studies concepts related to TTS, overview of speech synthesis, the history and development of text to speech synthesis, different techniques of speech synthesis with their advantage and disadvantage and discusses about Amharic language phonologies and writing system and also the approaches of text to speech synthesis reviewed. Chapter 3 review works done to design TTS system for Ethiopian language and some of non-Ethiopian language descriptions about the techniques and metrics used in the thesis. Chapter 4 describes the design and phases of BLSTM based Amharic text to speech synthesis. Chapter 5 describes the experiments conducted to evaluate the design of TTS for Amharic language. And how the data selection and different techniques plays a significant

role on the performance. Chapter 6 summarizes and conclude the key concepts of the thesis and provides directions for future work.

Chapter Two: Literature Review

2.1 Introduction

In this chapter, the theoretical basis of speech synthesis, models of speech production and how these are used in text-to-speech conversion will be reviewed. In the first section of the chapter the introduction and overview of text to speech and its components will be presented. In the second section of the chapter the historical development of text to speech synthesis will be discussed. The third part presents the different methods which can be used to do a text to speech synthesis and the advantage and disadvantages of those methods also will be discussed. Different related works which are done on Amharic and other languages will also be reviewed.

2.2 Text to Speech Synthesis

Speech is the leading means of communication among people. Speech synthesis is a manner of automatic generation of speech via machines/computers. The aim of speech synthesis is to develop a machine having an intelligible, natural sounding voice for passing on information to a user in a preferred accent, language, and voice. A system that is employed for this purpose is understood as a speech synthesize. A system that is employed for this purpose is understood as a speech synthesizer, and might be applied in code or hardware. The process of converting text into speech comprises generally two stages those are text analysis and production of speech signal [19].

Text analysis module contains normalization of the text where in the numbers and symbols become words and abbreviations are replaced by their whole words or phrases etc. The main challenging duty in the text analysis block is the linguistic analysis which means the syntactic and semantic analysis and intentions at understanding the context of the text. The statistical ways are used to get the greatest likely meaning of the utterances. This is significant because the pronunciation of a word may depend on its meaning and on the context [19].

Phonetic analysis changes the orthographical symbols into phonological ones by means of a phonetic alphabet. For e.g. the alphabet of the International Phonetic Association contains phoneme symbols, their diacritical marks and other symbols related to their pronunciation, other phonetic alphabets such as SAMPA (Speech Assessment Methods Phonetic Alphabet), Worldbet and Arpabet are also available [19].

Prosody is a concept which comprises the rhythm of speech, stress patterns and intonation. At the perceptual level, naturalness in speech is credited to some properties of the speech signal correlated to noticeable variations in pitch, loudness and syllabic length, cooperatively known as prosody. Acoustically, these changes parallel to the distinctions in the fundamental frequency (F0), amplitude and duration of speech units [19].

Speech synthesis block lastly produces the speech signal. This can be accomplished either founded on parametric representation, in which phoneme comprehensions are created by machine, or by choosing speech units from a database. The subsequent short units of speech are combined together to generate the final speech signal. The figure below shows a block diagram of T-T-S synthesis [19].

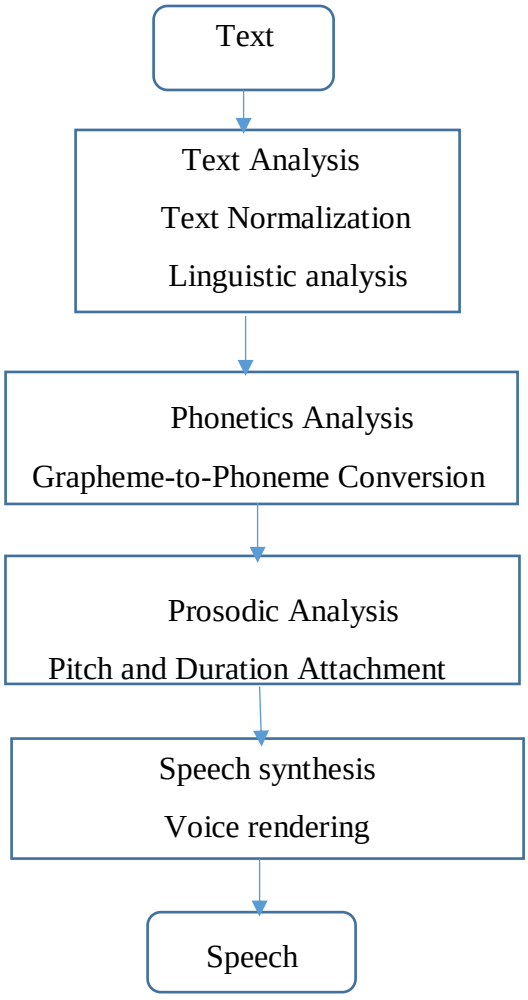

Figure 2.1 Block diagram of TTS [19]

History and development of text to speech system

In this section we will see the historical development of text to speech system, how they have changed from the first mechanical system to the electrical systems and how they have advanced to their present form will be discussed.

From Mechanical to Electrical Synthesis

The initial efforts to produce synthetic speech were made before two hundred years ago [20, 21, 22]. In St. Petersburg 1779 Russian Professor Christian Kratzenstein defined physiological changes among five long vowels (/a/, /e/, /i/, /o/, and /u/) and made device to generate them artificially. He made acoustic resonators like to the human vocal tract and activated the resonators using vibrating stems like in music instruments. The sound /i/ is created by carrying into the lower tube without a reed affecting the flute-like sound.

In 1791, Kempelen presented acoustic mechanical speech Engine after few years later [20] [21]. In fact, Kempelen started his work beforehand Kratzenstein, in 1769, and afterward 20 years of research he also delivered a book in which he defined his studies on human speech generation and investigates with his speaking engine.

In about middle 1800's Charles Wheatstone built his famous version of von Kempelen's speaking machine. It was a bit more complex and was proficient to yield vowels and most of the consonant sounds. Certain sound combinations and even full words were also probable to produce.

In late 1800's Alexander Graham Bell with his father, encouraged by Wheatstone's speaking machine, built same kind of speaking machine. The research and investigations with mechanical and semi-electrical analogs of vocal system were made till 1960's, however with no remarkable achievement. The mechanical and semi-electrical investigations made by well-known scientists, such as Herman von Helmholtz and Charles Wheatstone [22].

Development of electrical synthesizers

In 1922 the initial full electrical synthesis device was presented by Stewart [23]. The mahine had a buzzer as excitation and two resonant circuits to model the acoustic resonances of the vocal tract. The machine was able to generate only static vowel sounds with two nethermost formants, however not any consonants or connected utterances. Equivalent kind of synthesizer was ended by Wagner [24]. The device entailed of four electrical resonators connected in

corresponding and it was excited by a buzz alike source. The outputs of the four resonators were shared in the proper amplitudes to produce vowel spectra. In 1932 Japanese researchers Obata and Teshima revealed the third formant in vowels [20].

Mechanical speaking machines were exchanged with the advent of electrical technology by electrical devices, the principal device measured as a speech synthesizer and one distinguishable initial version of which was the VODER (Voice Operating Demonstrator), first demonstrated at the world fair of 1939 in New York by Homer Dudley. The VODER functioned through inspiring a set of fixed filters, which proceeded as resonators, which taken to produce a particular sound were organized by a human operator. The VODER comprised of wrist bar for choosing a voicing or noise source and a foot pedal to govern the fundamental frequency. The supply signal was stirred over 10 band pass filters whose output levels were organized by fingers. It acquired substantial ability to play a sentence on the device. It was lastly presented that intelligible speech can be produced falsely. Certainly, the main structure and idea of VODER is very equivalent to the present systems which are stucked on source-filter-model of speech. Though, the speech quality and intelligibility were distant from good but the potential for producing artificial speech were well revealed and it was only with the invention of a mode1 of speech production that speech synthesizers as they are nowadays seemed [20, 21, 23].

Nearby a decade later, in 1951, Franklin Cooper and his friends developed a Pattern Playback synthesizer at the Haskins Laboratories [21, 23]. The four first formants are usually reflected to be enough for intelligible synthetic speech. The head formant synthesizer, PAT (Parametric Artificial Talker), was presented by Walter Lawrence in 1953. The earlier articulatory synthesis engine was DAVO (Dynamic Analog of the Vocal tract), presented in 1958 by George Rosen at the Massachusetts Institute of Technology, M.I.T.

In 1968 Japan, the major full text-to-speech system for English was established in the Electro technical Laboratory, by Nariko Umeda and his buddies Klatt [21]. It was founded on an articulatory model and comprised a syntactic analysis module with refined heuristics. The speech was reasonably intelligible but boring and far-off from the quality of present systems. In 1979 Mi-Talk works text-to-speech system developed at M.I.T by Klatt Hunnicutt, and Allen showed the [21].

Modern speech synthesis technologies comprises reasonably complex and refined methods and algorithms. Neural networks have been experienced in speech synthesis for about ten years and the latest results have been somewhat encouraged. Nonetheless, the possibility of using neural networks have not been adequately discovered [22].

2.3 Evaluation of Speech Synthesis

Speech synthesis evaluation is in itself another research topic and is somewhat essential when benchmarking the performance of a system. A brief overview of the objective metrics that are used in this thesis work and subjective methodologies that have been used for evaluating listening tests is given below.

2.3.1 Objective Evaluation

In conventional SPSS systems, a vocoder world is commonly used to reconstruct the final waveform from the predicted acoustic parameters. Hence, objective evaluation is used to match the predicted parameters with reference parameters mined from natural speech. Most of the metrics in general use are distance measures. The underlying assumption is that a model with relatively smaller distance performs the best, leading to a better speech synthesis system [46].

Acoustic measures:

Mel-Cepstral Distortion (MCD) [46] measures the distortion between predicted and extracted (from natural speech) Mel-Cepstral Coefficients and is defined as:

$$\text{MCD} = \frac{\alpha}{t} \sum_{t=1}^{T} \sqrt{\sum} (\text{xd(t)} - \hat{x}\text{d(t)})^2 \quad \text{.....(2.3.1)}$$

where T is the total number of frames in the test data and D is the dimensionality of the mel-cepstral coefficients extracted at each frame

2.3.2 Subjective Evaluation

Objective measures are not only used to optimize hyper parameters during training but also often used as an indication of the quality of synthetic speech. However, subjective listening tests still remain the standard method for evaluation of synthetic speech. Since, the primary focus of this thesis is to improve the overall naturalness of the SPSS by enhancing the prosody model, subjective listening tests will be of paramount importance in evaluating our systems.

They tend to be costly and at times require huge investment in terms of both time and resources, as they require human listeners [47].

Mean Opinion Score (MOS):

One of the most common and conventional ways to evaluate speech synthesis is via a mean opinion score (MOS) test. And has become standard practice when evaluating speech synthesis systems. Listeners are asked to listen to one speech sample at a time and rate it on a scale of 1 to 5 in terms of quality or naturalness, where 1 indicates bad and 5 indicates excellent. When evaluating MOS, listening tests are usually balanced to remove potential effects of repeated listening of the same utterance under different conditions by the same participant [47].

Transcription System

Transcription is needed since written text for Amharic languages does not correspond to its pronunciation. Hence, in order to describe the correct pronunciation some kind of symbolic presentation is needed. There were some efforts made to construct language independent phonemic alphabets during the last decades. Among these, IPA (International Phonetic Alphabet) and SAMPA are some to list. IPA, which is one of the best-known language-independent phonemic alphabets, consists of a huge set of symbols for phonemes, suprasegmental, tones/word contours, and diacritics. On the other way, SAMPA is considered to map IPA symbols to 7-bit printable ASCII characters to ease the difficulty and the use of Amharic symbols that make IPA alphabets inappropriate for computers which usually require standard ASCII as input. Even if there are several other phonetic representations and alphabets used in present TTS systems, there is no single generally accepted phonetic symbol [23].

2.4 Types and Approaches of Text to Speech Synthesis

Synthesized speech can be produced by several methods and those methods have their own advantages and disadvantages.

2.4.1 Formant Synthesis

In this type of speech synthesis, the main notion is that the vocal tract transfer function is modeled by using formant frequencies and formant amplitudes. Thus the synthesis involves the artificial rebuilding of the formant characteristics to be created. By moving a set of resonators by a voicing source or noise generator to realize the wanted speech spectrum, and by governing the excitation source to pretend either voicing or voicelessness which can be done. The addition of anti-resonators additionally permits the imitation of plosives nasal tract effects and fricatives. The requirement of about 20 or more such parameters can lead to a satisfactory restoration of the speech signal. The benefit of this method is that its parameters are extremely connected with the creation and transmission of sound in the oral tract. The major drawback of this method is that automatic methods of requiring formant parameters are still mainly inacceptable, and that accordingly, the majority of parameters must still be improved [25].

The formant synthesis doesn't use any human speech samples but relies on rules written by linguists to generate the parameters that will permit the synthesis of speech, and to agree with the conversion from one phoneme to another, that is, the co-articulation. To write the rules, linguists have studied spectrograms and derived the rules of evolution of formants. However we do not yet know the optimal rule to do this [23]. Moreover, the speech waveform is naturally produced in such a complex process that, currently, rules can only model the features of the speech waveform.

When memory and processing costs are limited, such as in embedded systems, these synthesizers are more fascinating because they don't have a database of speech samples [26].

2.4.2 Articulatory Synthesis

Articulatory synthesis generates speech by direct modeling of the human articulator behavior, thus in principle it's the foremost satisfying technique to supply top quality speech. Actually, it is one of the most difficult method to implement. The articulatory control parameters consist of lip aperture, lip protrusion, tongue tip position, tongue tip height, tongue position and

tongue height. There are two complications in articulatory synthesis. The primary difficulty is getting data for the articulatory model. This data is usually derived from X-ray photography. X-ray data do not depict the masses or degrees of freedom of the articulators. Another trouble is to discover a balance among an extremely accurate model and a model that is easy to design and regulate. Totally, the results of articulatory synthesis are not as good as the results of formant synthesis or the results of concatenative synthesis [26].

2.4.3 Concatenative Synthesis

The concatenative speech synthesis methodology may be a corpus primarily based methodology that customs certain quantity of pre-recorded speech samples (words, syllables, half-syllables, phonemes, diphones or triphones) in an exceedingly information and produces the output speech by combining appropriate units based on the arrived text utterances. The comfort of the model and highly natural speech production quality marks it suitable for its use in designing human computer interactive systems for different areas. The quality of the synthesized speech is exaggerated by the unit length in the database. The naturalness of the synthesized speech rises with longer units while by using longer units less concatenation points are there reducing the generation of unnatural sections at concatenation points. Though, more memory is wanted and the amount of units stored in the database comes numerous. On the other hand, with shorter units, the memory prerequisite is less but the complexity of sample collection and labeling techniques rises. The concatenative technique may generally be classified into the following three groups based on the unit type stored in its database [27].

A. Unit Selection Synthesis

Unit selection synthesis practices a huge database of recorded speech. During database construction, each recorded utterance is divided into individual phones, syllables, morphemes, words, phrases, and sentences. Typically, the division of segments is prepared using specifically modified speech recognizer set to a forced alignment manner with certain labor demanding correction later, consuming graphical representations such as waveform and spectrogram. The directory of the units in the speech database is formerly produced depend on the dissection and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and bordering phones [27].

At runtime, the desired target utterance is created by defining the finest chain of nominee units from the database. This process is characteristically done using a specially weighted decision tree. In the unit selection order, by using the target cost and the concatenation cost, speech units are nominated from the entire speech database, and concatenated in run-time. In this scheme, a heuristic distance is well-defined among contexts to measure the target cost. To avoid this, a clustering-based outline may be used which groups the contexts in improvement, and chooses each unit from a cluster [27].

As only a small amount of digital signal processing is applied to the recorded speech, unit selection technique produces extremely natural speech segments. But, in order to get maximum naturalness unit-selection method needs speech databases to be very large. Similarly, unit choice algorithms are known to decide on segments from an area that leads to not good synthesis. For instance, minor words become unclear, even when a better choice exists in the database. Lately, researchers have proposed plentiful computerized methods to notice unnatural segments in unit selection speech synthesis systems [27].

B. Diphone Synthesis

In this kind of synthesizers, all the diphones (sound-to-sound transitions) happening in a language are enclosed in the speech database. A diphone is made of two connected half phones and captures the conversion between two phones by starting in the middle of the first phone and finish in the middle of the second one. The amount of diphones relies on the acoustics of a language: for example, for Spanish and germany 800 and 2500 diphones respectively. Identifying the number of diphone units in a language is a difficult task. Only one instance of every diphone is enclosed in the speech database and at runtime, the prosody goal of a sentence is put under these nominal units by the way of digital signal processing practices such as LPC, PSOLA or MBROLA [28].

In diphone synthesis, only 1 of the speech unit is accessible, overall manner of speaking changes have to be compelled to be applied to urge an honest quality of speech.. Diphone synthesis experiences from the robotic sounding synthesized speech. Even though due to a number of freely accessible software implementations, it remains to be used in research but its use in commercial applications is diminishing. TTS systems depend on diphone synthesis require prosodic models to generate good speech output. The prosodic analysis for these

models entails a database of speech interpreted with linguistic and prosodic labels. Tools also are needed to get applicable linguistic data essential to predict prosody from text [28].

C. Domain Specific Synthesis

The domain specific synthesis stores recorded speech samples of some mostly used words or phrases for certain domains and combination of those segments is done to produce full utterances. It is typically used in applications like announcement of transit schedules, weather reports, railway inquiries where the variability of texts the system will output is limited to a particular domain. This expertise is in marketable use for a long time for its humble and calm to implement features. It can be realized in devices like calculators, talking clocks, etc [28].

The foremost advantage of this method is the level of naturalness. These systems can have a very high naturalness as the variety of sentence varieties are restricted, which narrowly matches the prosody and intonation of the recorded original speech. The main disadvantage of this type of techniques is the limitation of words and phrases in the databases. These categories of systems can only generate speech that associates the words and phrases in the database with which they are preprogrammed. Therefore, these kinds of TTS systems are specific purpose. The mixing of words within naturally spoken language can still basis problems if not the many dissimilarities are taken into account [28].
There are several problems in concatenative synthesis when compared with the other methods [23].

- Distortion from discontinuities in concatenation points, it will be condensed using diphones or certain distinct methods for flattening signal.
- The memory necessities are generally very high, particularly when long concatenation units are used, such as syllables or words.
- Data collecting and labeling of speech samples is typically time-intense. In notion, all possible allophones should be comprised in the material, however trade-offs among the quality and the number of samples must be made.

2.4.4 Hidden Markov Model Based Synthesis

HMM-based synthesis is a statistical parametric speech synthesis based on Hidden Markov Models (HMM). In this scheme, the range of frequencies in a signal, fundamental frequency, and duration of speech are modeled all together by HMMs. Speech waveforms are generated from HMMs themselves established on the maximum likelihood standard. Figure 2.3 shows the general HMM based speech synthesis process. It involves of two phases, the training phase and the synthesis phase. From the speech database, spectrum and excitation parameters created and shaped by context dependent HMMs within the coaching stage. A model commonly comprises of three states that characterize the beginning, the middle and the end of the phone. The synthesis stage deals with generation of speech signals by concatenating the context dependent HMMs according to the text to be synthesized [29].

The topmost advantage of the HMM-based speech synthesis approach is its voice characteristics can be simply adapted and can be practical to numerous languages with trivial modification. A changeability of speaking styles or emotional speech can be synthesized using the small amount of speech data. Correspondingly, the techniques developed in Automatic Speech Recognition (ASR) can be simply functional to it and its footprint is comparatively small [30].

The key disadvantage of the HMM-based synthesis approach contrary to the unit selection approach is the quality of synthesized speech. There appears to be three factors which degrade the quality: vocoder (an artificial sounds from an analysis of speech input), modeling accuracy, and over-smoothing. The speech produced by the HMM-based generation synthesis method sounds not clear as it is built on the vocoding technique. To ease this problem, a high-quality vocoder such as multi-band excitation scheme or STRAIGHT have been integrated. Several groups have recently applied LSP-type parameters as a replacement for mel-cepstral coefficients to the HMM-based generation synthesis approach. The Hidden Semi-Markov models (HSMMs), trajectory HMMs, and stochastic markov graphs are some other variations to obtained enhanced modeling accuracy. More, while sharing of clustering and asynchronous-state model structures are implemented for joined acoustic and articulatory features advance the accuracy of acoustic parameter prediction and the naturalness of synthesized speech [30].

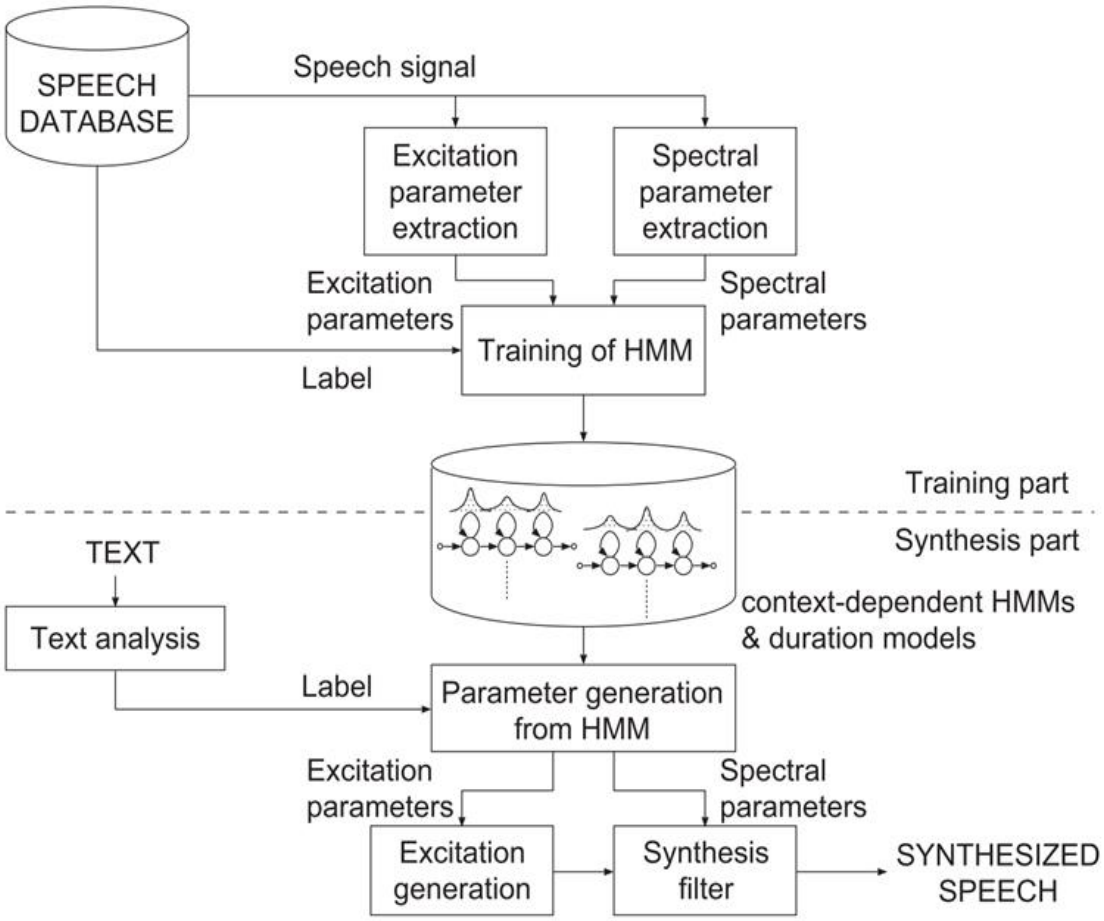

Figure 2.2 The HMM based statistical speech synthesis system [29]

Additionally, fundamental frequency, or F0 modeling, to get peech which is both natural and accurate HMM-based synthesis is important. Continuous F0 modeling produces better synthesized F0 trajectories and offers considerable enhancement to the naturalness of synthesized speech [26].

2.4.5 Deep Learning

Deep learning (DL) is another investigation in the machine learning area in recent years. It can successfully capture the hidden internal structures of data and use more powerful modeling capabilities to characterize the data [31]. Meanwhile 2006, deep structured learning, or usually known as deep learning or hierarchical learning, has emerged as a new area of machine learning investigation. Through the past over many years, the techniques developed from deep learning analysis have already been impacting a large vary of signal and

information processing work inside the old and the new, widened scopes together with key features of machine learning and artificial intelligence [32].

Until as of late, most machine learning and signal processing techniques had abused shallow-structured models. These models usually contain at most one or two layers of nonlinear feature transformations. Shallow architectures have been accessible operational in resolving many simple or well-constrained difficulties, nevertheless their inadequate modeling and representational power can cause problems when dealing with more complex real-world applications including natural signals for instance human speech, natural sound and language. Human information handling devices (e.g. audition), advocate the need of deep models for extracting complex structure and constructing internal representation from rich sensory inputs. For instance, human speech construction and opinion systems are both provided with clearly layered hierarchical structures in transforming the information from the waveform level to the linguistic level [32].

Deep Learning in Speech Synthesis

The impact of deep learning has recently unfold to text to speech synthesis, meant to beat the constraints of the traditional approach in applied mathematics constant synthesis supported Gaussian-HMM and decision-tree-based model clustering. Speech sounds generated by this standard approach are often muffled (lowered) when compared with natural speech. The insufficiency of acoustic modeling based on the narrow structured HMM is indirect to be one of the causes. Numerous very new studies have implemented deep learning methods to over whelm such deficiency. One important advantage of deep learning techniques is their sturdy ability to represent the intrinsic correlation or mapping relationship between the units of a high-dimensional random vector [32]. The family of deep learning approaches have been rising progressively richer, covering those of neural networks, hierarchical probabilistic models, and a range of unsupervised and supervised feature learning algorithms.

Deep Neural Network for Speech Synthesis

Deep neural network is in the family of deep learning architecture and it is a conventional multilayer perceptron (MLP) with many (often more than two) hidden layers. Deep neural networks (DNNs) have been used as acoustic models on behalf of statistical parametric speech synthesis (SPSS) [33].

In DNN-based speech synthesis, DNN pretends human speech construction through a layered hierarchical structure to change linguistic text information into its final speech output. The inadequacy of decision-tree based contextual state clustering, which is used to connect states of long contexts into generalized ones to forecast unnoticed contexts in testing better than the HMM-based counterpart, is overcome in DNN. DNN too can embody high dimensional and interrelated features well and model highly complex mapping function efficiently. Though, DNN is still suffering from using short unit, e.g., state or frame, as a main modelling unit. To capture co-articulation result and imitator natural pitch, very rich contexts are used as input features. To synthesize smooth speech parameter trajectories, the dynamic model parameters are essential to be used together with their static equivalents to produce smooth parameter trajectories [34].

The sequential nature of speech is discounted and this is one of the inadequacy of the feed-forward DNN-based acoustic modeling is. While indeed there are correlations among successive frames in speech data, the DNN-based method undertakes that each frame is independent. It is necessary to integrate the sequential nature of speech data to the acoustic model itself.

DNNs do not naturally model the sequential structure in speech and text. Using DNNs as acoustic models will limit usage of context to a few phones in past and future and lead to discontinuities in the predicted parameters. The first problem can be addressed by means of recurrent neural networks, which make use of unrestricted and adaptive context (for the reason of the capacity of RNNs to memorize the past). The earlier problem arises because of the training technique engaged for DNNs. Every frame of textual and acoustic representations is mapped frame wise in an independent manner. This leads to incoherence in predicted parameters from frame to frame, which is not the case with the real speech parameter trajectories [11].

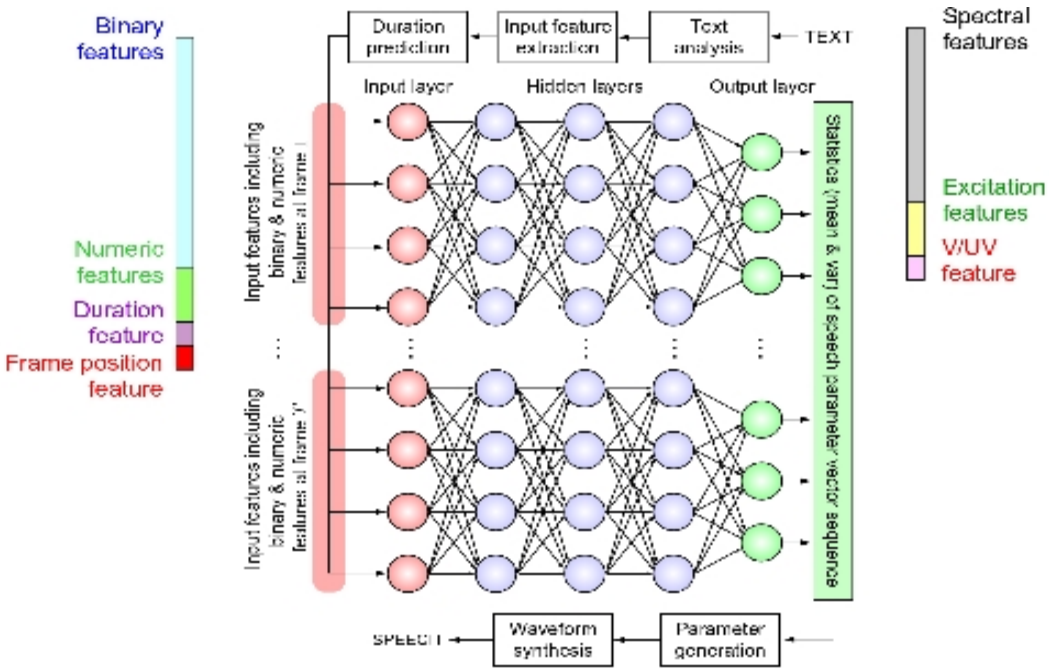

Figure 2.3 Typical DNN based SPSS with linguistic features and acoustic features [33].

DNN based speech synthesis works as, a given text to be synthesized is first changed to a sequence of input features {xtn}, where xtn represents the n-th input feature on frame t. The input features contain binary answers to questions about linguistic contexts (e.g. is-current-phoneme-aa?) and numeric values (e.g. the number of words in the phrase, the comparative location of the current frame in the current phoneme, and durations of the current phoneme).

Formerly the input choices are plotted to output choices by a trained DNN victimization forward propagation, wherever ytm represents the m-th output feature at frame t. The output features embrace spectral and excitation parameters and their time derivatives (dynamic features). The weights of the DNN can be trained with pairs of input and output features extracted from training data. In the same manner as the HMM-based method, it is possible to produce speech parameters; by means of setting predicted output options from the DNN as mean vectors and pre-computed variances of output options from all coaching information as variance matrices, speech parameter generation rule will manufacture swish trajectories of speech parameter options that fulfill each the statistics of static and dynamic features. To end, a waveform synthesis part outputs a synthesized waveform given the speech parameters. The HMM-based one can share modules of the DNN primarily based system that square measure

the text analysis, speech parameter generation, and wave shape synthesis. But the mapping module from context-dependent labels to statistics essentials to be substituted [33].

Recurrent Neural Network for Speech Synthesis

Recurrent neural networks is alternative class of deep learning for unsupervised as well as supervised learning, where the depth of the model can be as large as the length of the input data sequence. In the unsupervised learning mode, the RNN is used to forecast the data sequence in the future using the previous data samples, and no extra class information is used for learning. One of the ability of recurrent neural networks it can model time-series. RNN is very influential for modeling sequence data (e.g. speech, text) [35].

They vary from deep neural networks in their capability to send information over time-steps. Recurrent neural networks accept each vector from a sequence of input vectors and model them one at a time. This lets the network to retain state while modeling each input vector through the window of input vectors. Modeling the time dimension is a symbol of recurrent neural networks. Recurrent neural networks add the concept of recurrent connections. The adjacent time-steps spans and give the model the concept of time through recurrent edges. The expectable connections do not comprise cycles in recurrent neural networks. Yet, recurrent connections can form cycles encompassing connections back to the original neurons themselves at upcoming time-steps. At each time-step of sending input through a recurrent network, nodes receiving input beside recurrent edges receive input activations from the current input vector and from the hidden nodes in the network's earlier state. The output is calculated from the hidden state at the given time-step. The previous input vector at the earlier time step can affect the current output at the current time-step over the recurrent connections. [35].

Recurrent neural network exploits the sequential nature of their input. Such inputs could be text, speech, time series, and whatever where the existence of an element in the sequence is reliant on the elements that appeared beforehand of it [36].

Recurrent neural networks which can process sequences of inputs and yields sequences of outputs. Mostly, the RNN model is different from the DNN in the following method: the RNN operates not only on inputs (like the DNN) but likewise on network internal states that are updated as a function of the whole input history. In this case, the recurrent connections are

capable to map and recall information in the acoustic sequence, which is significant for speech signal processing to improve prediction outputs [6].

Recurrent neural networks (connectionist models) can capture the long-range time dependencies in the input file. They achieve this since their hidden state captures information from randomly long context window and does not have the restriction of the other methods. Furthermore, the number of states they can model is represented by the hidden layer of nodes, and these states raise exponentially with the number of nodes in the layer. This marks them special at capturing a lot of time-dimension relevant information through many input vectors [35].

Vanishing and Exploding Gradients

Training RNN integrates backpropagation. The dissimilarity in this case is that as the parameters are shared by all time steps, the gradient at each output hang on not only on the current time step, but likewise on the earlier ones. This process is named as backpropagation through time (BPTT). The consequence of vanishing gradients is that the gradients from steps that are far away do not give whatever to the learning progression, therefore the RNN ends up not learning long range dependencies. Vanishing gradients can arise for traditional neural networks as well, it is just more visible in case of RNNs, and since RNNs have a tendency to have several layers (time steps) over which back propagation must occur. Exploding gradients measure a lot of simply detectable, the gradients can become terribly massive so change into not variety (NaN) and therefore the coaching method can crash. Exploding gradients can be controlled by cutting them at a predefined threshold [36].

However there are a few methods to diminish the difficult of vanishing gradients, such as proper initialization of the matrix, using a ReLU instead of tanh layers, and pre-training the layers using unsupervised means, the furthermost popular solution is to use the LSTM or GRU architectures. These architectures have been designed to agree with the vanishing gradient problem and absorb long term dependencies more efficiently [36].

Long Short-Term Memory

Long short-term memory networks (LSTM) are a class of recurrent networks composed of units with a certain structure to manage better with the vanishing gradient problems during training of recurrent neural network and maintain potential long-distance dependencies. This makes LSTM appropriate to learn from history for the purpose of classify, process and predict

time series.Unlike the standard repeated unit that overwrites its content at whenever step, LSTM have a special memory cell with self-connections within the repeated hidden layer to keep up its states over time, and 3 gating units (input, forget, and output gates) that area unit accustomed management the data flows in and out of the layer similarly as once to forget and recollect previous states [37].

The structure of an LSTM cell is presented in figure below. There are 3 gates during this structure:

Input gate: control the flow of information coming in.

Forget gate: control which components of the cell state are forgotten (i.e. multiplying by zero to delete from memory).

Output gate: control the flow of information going out.

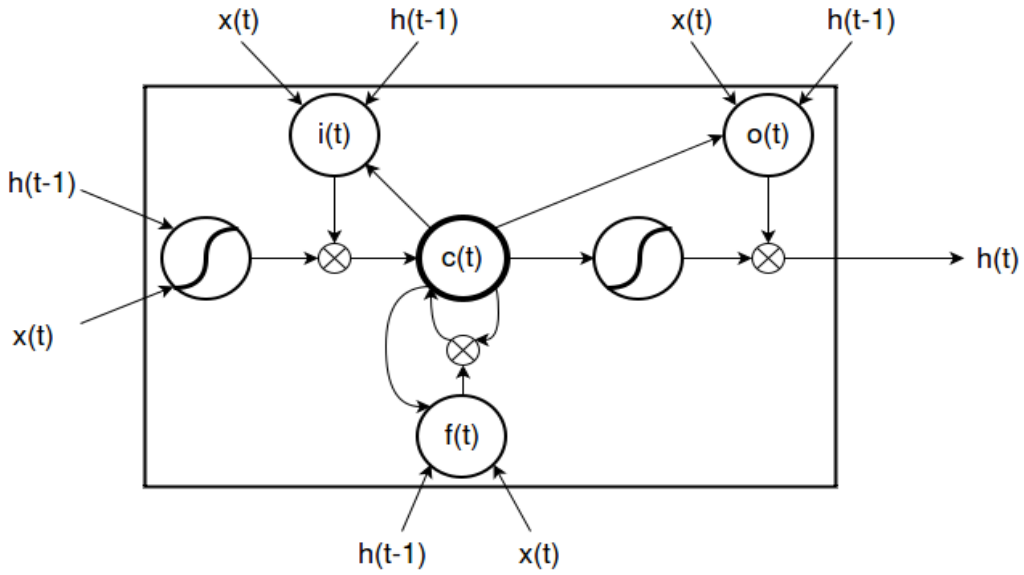

Figure 2.4 Architecture of an LSTM cell [37]

i(t): input gate at time-step t. o(t): output gate at time-step t. f(t): forget gate at time-step t. c(t): cell state at time-step t.

2.5 Orthography and Phonology of Amharic Language

This chapter presents the nature of Amharic language. The first part presents the overview of Amharic language and discusses the orthography of the language. The second deals about the phonology (Amharic vowels and consonants) of the language.

Overview of Amharic Language

Amharic (አማርኛ) is an Afro-Asiatic language of the Semitic division. Ethiopia uses this language as official working working language. Amharic is the second furthermost broadly spoken Semitic language in the world afterward Arabic. The language is spoken by 22 million native speakers in Ethiopia. Furthermore, from the outdoor of Ethiopia 3 million emigrants speak this language [38]. Its written form is left-to-right using Ethiopic Fidel, ፊደል, which developed out of the Ge'ez knowns as abugida, in Ethiopian Semitic languages, ፊደል (Fidel) writing system, letter, or character. The word አቡጊዳ (abugida) taken from the first four Ethiopia letters, which gave rise to the recent linguistic term abugida. The scripts are more or less orthographic representation of the phonemes in the language [39].

Amharic Orthography

The script of Amharic is abugida, and the grapheme of the Amharic writing system is known as Fidel [38]. Amharic orthography contains 276 distinct symbols. Furthermore, there are twenty numbers and eight punctuation marks. The script has 33 main characters and 32 of them are consonants having seven orders to denote the seven vowels. Out of the seven derivatives six of them are CV (Consonant vowel) mixtures whereas the sixth is the consonant itself. Every character characterizes a consonant and vowel sequence, nevertheless the basic shape of each character is determined by the consonant, which is adapted for the vowel. Certain consonant phonemes are written by more than one series of characters. This is because these Fidel initially represented different sounds, however phonological changes fused them [39].

The reference form for each series is the consonant form, i.e. the first column of the fidel. Amharic script is encompassed in Unicode, and glyphs are involved in fonts accessible with foremost operating systems. The writing system of Amharic is now derived from Ge‟ez. Amharic do not distinguish in implementing the Ge‟ez fidel; it acquired all of the symbols and added some new ones that represent sounds not found in Ge‟ez. These extra alphabetic characters are, ጨ, ጀ, ኘ, ቨ, ሸ, ኸ, and ዠ. Presently, the orthographic representation of the language is structured into seven orders. Amharic language's orthography includes thirty four base characters every of that happens in a very basic type and 6 alternative forms called orders. The seven orders represent syllable combinations containing of a consonant succeeding

vowel. Out of the seven derivatives six of them are CV (Consonant Vowel) combinations whereas the sixth is the consonant itself. Consequently, having these orthographic variations for each of the 33 core letters, totally the language has more than 230 orthographic symbols; in case of Amharic character ሀ, it is represented by "ha" rather than mistreatment "he" not like different Amharic character sets/orthographies since the sound is that the same as its fourth order [39].

Phonology of Amharic Language

Phonology is that the study of the distribution and patterning of speech sounds during a language and of the implicit rules leading pronunciation. In phonology, phoneme is the important unit that defines how speech delivers linguistic meaning. The phoneme characterizes a class of sounds that express similar sense. The meaning of a word is reliant on the phoneme that it comprises [40]. Nonetheless, all of the letters of the Amharic script are not crucial for the pronunciation patterns of the spoken language; certain were simply inborn from Ge'ez without having any semantic or phonetic difference in recent Amharic. There are many circumstances where many symbols are used to represent a single phoneme, as well as words that have exceptionally dissimilar orthographic form and somewhat dissimilar phonetics, but with the similar meaning [41].

For instance, the Amharic characters ሀ, ሐ and ኀ have diverse orthographic arrangement, but still having similar meaning and signifying similar phoneme. Nonetheless, in Ge'ez there are cases where to use which symbol as it takes variance in its meaning. Commonly, the Amharic phonemes are designed from two wide-ranging groups: vowels and consonants. Semi-vowels are considered to be part of consonants [42].

Amharic vowels

Vowels are the types of sound, which make the slightest obstacle to the flow of air. The tongue shape and locating in the oral cavity do not form a major restriction of air flow through vowel articulation [43]. Nevertheless, differences of tongue settlement offer each vowel its different character by altering the resonance, just as different sizes and shapes of bottles give increase to different acoustic effects when hit. This also decides the length of the vowel to be produced [44].

In Amharic there are seven vowels, these are ኧ, ኡ, ኢ, ኣ, ኤ, እ and ኦ. All are voiced and spoken sounds. These vowels can be originate in each letter, that is, each letter in Amharic is not a single sound relatively it is a mixture of two sounds, one from vowel and one from consonant. Amharic languages sounds of letters are a mixture of a vowel and consonant [39]. The table below displays the combination of each of the vowels and the consonants (በ and ሙ).

Table 2.1 IPA maps of the Amharic vowels

Order	1st	2nd	3rd	4th	5th	6th	7th
V / C	E	U	I	A	Ie	Ix	O
M	መ	መ፡	ሚ	ማ	ሜ	ም	ሞ
B	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ

Amharic Consonants

In Amharic language there are 27 consonants from those the total of 34 phonemes. Each of them mostly varies in the way of articulation, place of articulation, and vibration level. When the air pressure produced by the lung passes over closely closed vocal cords, vibration is produced as the air vibrates the vocal folds. Accordingly, vibration level is one way of categorizing consonants. Consonants that are obstructed by the vocal cord whereas they are produced are called voiced sounds and others are voiceless sounds. Based on this, Amharic consonants are categorized into two: voiced sound and voiceless sound. Afterward the air passes the vocal cord, it has two choices. One is to go through the nasal cavity and the other is to go through the mouth and this is determined by velum. As the air passes through either the mouth or the nose, the sounds which will be produced are dissimilar [38].

Altogether, consonants that are generated by nasal cavity are called nasal sounds. Whereas, those who are created orally/mouth cavity are referred to as oral sounds. These differences in consonants occur because of the difference in place of articulation. The place of articulation of a consonant is the point of interaction, where an impediment happens in the vocal tract between an active articulator and a passive articulator [38]. Even for nasal sounds the active

and passive articulator are needed. The way the air is generated from the lung behaves based on the way active articulators behave. This is referred as manner of articulation.

There are six types of consonants in the language: stops, fricatives, nasals, affricates, semivowels and liquids, which are classified based on manner of articulations. Nasals, liquids and semivowels are always voiced; stops, fricatives and affricates can be voiced or unvoiced. They can also be classified as labials, alveolar, palatals, velars, labiovelar and glottal based on place of articulation [45].

A. Stop consonants are generated when the air is blocked and immediately released by articulatory organs. Hence, the stop consonants are distinguished by which articulatory organ the air is blocked, that is, whether the air is blocked by lips, tongue and velars, or tongue and alveolar.

B. Fricatives comprise allowing the air slide through a thin opening in the mouth. They can be extended for certain time and the air is not totally blocked.

C. Nasal sounds cannot be generated if the air is blocked from passing through the nasal cavity. Hence, there ought to be associate air flow through the cavum to provide them. The air will be directed to the nose since different parts of the mouth blocks it from passing through the mouth.

D. The affricates begin as stops and slide into fricatives, and hence are represented as a stop followed by a fricative.

E. The top of the tongue closes the vocal tract leaving a sideways route for the air flow.

Table 2.2 Consonants with their feature [45]

Manner of Art/n	**Voicing**	**Place of Articulation**											
		Labials		***Alveolar***		***Palatals***		***Velars***		***Labiovelar***		***Glottal***	
Stops	Voiceless	P	ፕ	T	ት			K	ክ	kx	ኪ		
	Voiced	B	ብ	D	ድ			G	ግ	gx	ጊ		
	Glottalized	Px	ጵ	Tx	ጥ			Q	ቅ	qx	ቋ		
Fricatives	Voiceless	F	ፍ	S	ስ	Sx	ሽ					H	ህ
	Voiced	V	ቭ	Z	ዝ	Zx	ዥ						
	Glottalized			xx	ጽ							Hx	ኋ
Africative	Voiceless					C	ች						
	Voiced					J	ጅ						
	Glottalized					Cx	ቾ						
Nasals	Voiced	m	ም	N	ን	Nx	ኝ						
Liquids	Voiced			L	ል								
				R	ር								
semi-vowels	Voiced	W	ው			Y	ይ						

Chapter Three: Related Work

3.1 Introduction

Text to Speech had been the attention of furthermost of the investigations from last decade. Text to Speech was for various specialization fields (Android, Embedded System, and Education etc.) had been developed numerous languages. Approach to give a text as an input and a technique can play an important role to convert into a speech in an effective manner. Based on the area of application, suitable means to be accepted. Following are the papers reviewed for the understanding of the topic. Different researchers develop a speech synthesizer for different languages such as for English, Arabic, AffanOromo a, Tigrigna and Amharic and so on and some of them are reviewed here.

3.2 Text to Speech Synthesis for Non-Ethiopian Languages

An Arabic text-to-speech system based on artificial neural networks is developed by [9]. There are different phases the first one is text pre-processing in this phase before the words enter the neural network, a series of preliminary processing are satisfied first, the punctuation marks are removed, then the numbers are identified and also the abbreviations are expanded into full words. Then the words are prepared as input vectors for the neural network. Though, only numerical inputs are recognized by neural networks, thus, the ASCII code of each character is taken and substituted with its equivalent binary representation. The second phase is text to speech conversion: In this phase they have created three models to hold dissimilar sizes of units those are word, diphone and triphone model. The third stage is training the word, training the diphone and training the triphone. Since their database of speech doesn't contain complete words, they have constructed each word out of its diphone sequence. To train the words of the dictionary, each word is altered into its diphone arrangement previously passed to the pre-processing unit. Since neural networks need that all inputs are of the same length, they have selected a vector length of 154 with respect to the longest word in the dictionary. Accordingly, words generating a vector smaller than 154 are expanded with trailing zeros.

The fourth stage is synthesizing words, diphones and triphones. In this process the input text is tokenized into single words and each word is processed individually to produce the feature vector and the output pattern. Then the pattern is compared with the patterns which are found in the look-up table and classified by the Euclidean distance metric. The words are

automatically broken down to their diphone sequence in order to convert the input words into speech, each diphone will be transformed into a feature vector then the feature vector is trained using the network to produce the pattern lastly. Training the triphone is additionally identical because the one won't to train diphones, by a variance of the dimensions of the input and output units. A triphone is accessible by three characters producing a feature vector of 21 elements. Finally, the recognized word, diphone and triphone is mapped to the corresponding sound and output as a speech. Finally, the proposed system is evaluated using matlab7 and the average of the recognized words by the listeners was 92.26%. The average accuracy of the recognized sentences which is produced by the neural network is 99%.

A Bangla text-to-speech system using deep neural networks is developed in [48]. They have created their own dataset. They hired two professional voice artists (one male and one female, both providing 20 hours. The speech database contained more than 40 hours of speech. The database consists of 12,500 utterances. They have also prepared a pronunciation dictionary (lexicon) of 1, 35,000 words for front-end processing. Before training the neural network they have developed a text normalizer that converts non-standard words (NSWs) into pronounceable forms and a normalized text is produced as output. As a frontend text processor to get the linguistic features from the input text, they have utilized two open source front-end tools Ossian and Festival. HTS labels with state level alignment is produced by the front-end outputs. From these labels, a vector of linguistic features are generated by Ossian (or Festival). This feature vector is then fed to the duration model and acoustic model. Duration Model. The back-end of their Bangla TTS system consists of two deep neural networks. They have employed feed forward networks for both duration and acoustic modeling. The first one, duration DNN, takes linguistic features generated from front-end processing as input, and learns the proper duration information by updating weights.

They split the training data into three sets: training (94%), testing (3%) and validation (3%). Their duration DNN consists of 3 layers of hidden units where each layer contains 512 neurons. The network uses Gradient Descent optimizer with the learning rate of 0.002. The output of this network is then fed to the acoustic model. The acoustic DNN was trained to map the input linguistic features and the associated duration features into acoustic features. Linguistic features (sequence binary vectors) are normalized in the range of [0.01, 0.99] before

passing to input layers of DNN. Finally, acoustic features, the output of acoustic modeling DNN are normalized appropriately so that they can be used by a vocoder. They are reduced to zero mean and unit variance and then acoustic features are sent to a vocoder for synthesizing waveform they have used world vocoder. For objective evaluation they have choose the Perceptual Evaluation of Speech Quality (PESQ) score, and mean opinion score for their evaluation. Their evaluation shows that they got a good result.

3.3 Text to Speech Synthesis for Ethiopian Languages

Syllabification design and text to speech system for AffanOromo using unit selection technique is developed by [10]. Syllabification of any words with the exception of abbreviations and acronyms were considered in their system. In order to rigger the module, AffanOromo text is used. Subsequently normalization is followed. Grammatically wrong characters that appear in the word or text were deleted using devised rule in the normalization module. Representing the whole phonemes and digraphs is carried out, and if digraphs exist re-mapping is carried out. The normalized text formerly energies to the germination module. Syllabification is carried out by the using the syllabifier module. This is done victimization associate rule to syllabicate any input words in their legal sequence. The final output will be syllable boundary marked AffanOromo text. A word could also be syllabified in numerous structure however the formula selects the legal structure sequence for the input word. Here linguistic division implementation principles particularly the maximum-onset principle and timber hierarchy principle are enforced.

Diphone based Amharic speech synthesis is done by [49]. Di-phones were used as the basic concatenation units and the author used the linear predictive coding technique to yield the synthesized speech and likewise used interpolation pattern to diminish the discontinuous nature of the synthesized speech, Pascal and matlab tools were used, and the evaluation reported as good.

The unit selection voice is developed for Amharic using Festvox is discussed by [45]. The authors defined a transliteration scheme to work with Amharic scripts and incorporated constriction Amharic phone set, syllabification rules, letter to sound rules into Festvox. To evaluate the standard of Amharic synthesizer, they conducted sensory activity tests on eleven faculty students (2 females and nine males) twenty to thirty years' previous native speakers

of the Amharic language and therefore the average score of the projected. Amharic synthesizer was obtained 2.9. Though, the unit selection techniques do not permit for adjustment of the TTS system to varied speaking styles and speaker characteristics and needs databases of broad sizes.

Formant based speech synthesis for Amharic vowels is developed by [50]. The model defined in this work has two main parts. The first one is the analysis part that handles the text analysis (transcription of the input word) and extraction of the speech parameters. The second part is the synthesis part that generates the artificial speech. During this model, smaller speech units like phonemes and therefore the like don't seem to be hold on within the info rather the speech parameters are hold on. This extremely reduces the memory necessity of the speech synthesizer than other synthesizing methods. The synthesizer which was developed produces vowels according to their context in a given word. The technique models the human speech production system in the form of source and filter, in which the source is completely independent from the filter. The source is identified by the air flow from the lung to vocal cord and the filter represents the resonance of the vocal and nasal tracts, which are also known as the formant that varies from time to time. The resonance is due to the vocal tract while generating different sounds. In the evaluation the author have performed two round during the first round of the perception test, 84.68% of the vowels were correctly recognized and 93.01% is achieved in the second round. On average 88.85% of the vowels are identified in this test.

A synthesizer which follows a formant synthesis approach to generate a speech for a given Amharic input text was developed by [51]. The developer collected speech for voiced sounds and extracted parameters such as formants, bandwidth, pitch, etc. from the collected speech. The unvoiced sounds were too hold on by dividing them from all Amharic syllables. The author finally synthesized the speech by first generating the voiced sounds using the parameters from the inventory data and concatenating both the voiced and unvoiced sounds.

Hidden Markov Model based speech synthesizer for Amharic language is attempted by [52]. The vocalization structure generated by competition and festvox along with the parameters extracted from the raw wave information were used for coaching the model. The speech parameters used for training the model are Mel-spectrum coefficients and fundamental frequencies.In this analysis work the text that's planning to be synthesized was assumed to be

normalized, that is, all of the preprocessing activities are done before it's given to the synthesizer. Finally, the synthesized speech is generated from the trained model supported the input text. Technique which was used to test the performance of the system: namely Mean Opinion Score. In this technique, respondents got speech synthesized by HTS-FA and information driven approach (concatenation method) and so they gave a rank for every sentence for various criterion Based on the worth of the MOS, HTS-FA performs better than that of knowledge driven approach for both naturalness and intelligibility criteria. The evaluation showed that HTS Analysis for both female and male have an average 3.6 and standard deviation 0.21 for naturalness and average 4.16 and 0.16 for intelligibility.

A generalized Amharic Text-To-Speech (TTS) synthesis based on diphone unit concatenation synthesis to handle both Amharic standard and non-standard word is attempted by [53]. The author used Residual Excited Linear Predictive (RELP) coding method. The author made Diphone info used as a base store for Amharic information, like phone-sets, text utterances and recorded diphone sounds. The performance of the system shows on the typical an accuracy level of 73.75% for Amharic text containing both non-standard word and standard word. Additionally, the performance of the system was assessed by adopting the Mean Score Opinion (MOS) and achieved 3 and 2.8 MOS score for intelligibility and naturalness respectively. The limitation of this work is it uses the rule-based mapping process to convert non-standard words to their equivalent standard words. The author used non-standard words (NSWs) and standard words (common words and proper names) to build the system.

A concatenation-based speech synthesis for Amharic using unit selection method is developed by [54]. In this work the author tried to address epenthesis and germination in having as many allophones as possible and identifying contexts that determine allophonic variations. During this work, a minimum of 2 phoneme variations for every phone within the Amharic alphabet were built; and articulation, gemination and interrogative prosody modeling rules to form acceptable choice of those variations from context are also used. The performance of the system was evaluated by using the MOS techniques and generally the system achieved cumulative result of the naturalness of the system was 3.63 and its intelligibility was 3.53.

A syllable-based concatenation speech signal synthesizer on behalf of Amharic language by means of TD-PSOLA algorithm is supposed by [55]. The author applies syllable-based

concatenation speech synthesis approach to design TTS for Amharic language. He used Time Domain Pitch Synchronous Overlap and Add (TD-PSOLA) algorithm for the prosodic modification and speech waveform analysis/synthesis purpose and diphones and syllables were used as the basic concatenation units to synthesize sample. The system was obtained seventy three.75% and 89.58% victimization ORT check result for understandability for transcribed and syllabified texts and mean average score of three.45 victimization MOS check result for the naturalness for transcribed and syllabified texts.

A syllable-based speech synthesis system for Amharic language using Hidden Markov Model is developed by [56]. The author used the text and speech corpus with the size of 600 and split it to 550 (90%) for training and the rest 50 (10%) for testing data sets. The author thought-about solely the characteristics and method of creation of Amharic phonemes and used Mean Opinion Score (MOS) analysis technique, the understandability of the planned synthesizer was achieved, the overall performance of 75.56% for syllable based and 77.78% for phone-based system.

Hidden Markov Model (HMM) based speaker freelance Amharic text to speech system is developed by [57]. A speaker-independent modeling methodology were utilized victimization HMM-based text to speech technique on a browse speech info of 726 sentences verbalized by three feminine and three male speakers with various speaking styles. The speech information is first modeled by context Dependent HMMs. The corresponding three state left to right HMMs was mechanically trained by construct speaker freelance model as initial model, then speaker adjusted model is calculable by exploitation speaker independent model using one male speech data. A decision-based clustering procedure was functional in isolation to the dispersals of Mel-cepstral, log F0, and state durations of context-dependent phoneme HMMs. Finally, to improve the voice quality, trajectory HMM and mixed excitation model was included by applying parameter generation rule supported milliliter mistreatment dynamic options to the Gaussian Mixture Model. Objective analysis was conducted to gauge the speaker freelance or speaker adaptation coaching demo exploitation spectral analysis and preference score. Finally, subjective mean opinion score (MOS) evaluation was conducted to evaluate the overall performance of the adapted models and developed system. The developed Amharic speech synthesizer attains seventy four comprehensibility and seventy nada

naturalness MOS result from fifty subject's mother tongue Amharic speakers. As well the intelligibility test, a unit test is performed on the text normalizer. The performance of text normalizer is eighty fifth for Amharic numbers, punctuation marks and abbreviation.

Text to speech synthesis using found data for low-resource languages is developed by [58]. In the first part of the work, they have examined several kinds of found speech data in contrast with data collected for TTS. Their TTS works has been based on three main corpora. Amharic data contains 25 corpora with both male and female speakers. The data consists of about 40 hours from 300 different speakers, and about 10 hours of speech from 230 different speakers, a single male speaker and about 55 hours of recorded speech and Amharic Read Speech (ARS) corpus it consists of 20 hours of transcribed speech with 44 female speakers and 56 male speakers. Their initial test voices were four-hour subsets chosen using one of the following: f0, energy, speaking rate and level of articulation For each feature, they sorted utterances by feature value, and so designated 3 sub-sets and initialized 3 empty subgroups low, mid, and high, therefore future which are extra lowest vocalization analysis to the low set until set was four hours long, added the next highest scoring utterance to the high subset until subset was four hours long, and added the "median-scoring" utterance to the mid set, and then until their set was four hours extending the set altogether orders, interchanging the direction one communication at a time,. They trained a Merlin voice on every of the 3 subsets for all the higher than mentioned options, keeping all alternative factors identical among the voice models. They have got 87.3% preference for the naturalness and adapted voice.

3.4 Summary

From the above works we can see that different works was tried by different researchers to develop text to speech synthesis for different languages. Likewise, different researchers develop text to speech synthesis for Amharic language using different text to speech synthesis techniques and get different evaluation result. From other works we have seen that still the works need improvement by using another technique. The limitation of the works is the intelligibility, the sound quality and naturalness still remain a major problem. Some of the works does not consider abbreviated words, numbers, and punctuation marks. So far, there is no any research work that develops a speech synthesizer for Amharic using bidirectional long

short term memory. However, the BLSTM is used for other languages. Thus, we proposed to use Bidirectional long-short term memory to get a better result for Amharic speech synthesis.

Chapter Four: Design of Amharic Text to Speech Synthesis

4.1 Introduction

This chapter will discuss the design of the Amharic text to speech synthesis using bidirectional long-short term memory (BLSTM). This chapter describes the overall architecture and discusses the preparation details of the data and files which are necessary for the development of text to speech synthesis and explain how our text to speech synthesis works in detail.

Defining a Phone Set of Amharic

We have defined a phone set which consists Amharic symbols which are defined in terms of features, like whether it is consonant or vowel, what is the type of consonant, what is place of articulation for consonants, vowel frontness and also it includes the kind of vowel, consonant phoneme descriptions sideways with their length, height, place of articulation and other vital elements for apiece of the phonemes. Therefore, all the phonemes, including phones which are representing silences are prepared. The defined Amharic phone set is found in **Appendix A** and within the phone-set the above features are defined based on the language structure since defining phone set is language dependent task.

Pronunciation Lexicon and Letter to Sound Rules

The method for finding the pronunciation of a word we use either a lexicon (a large list of words and their pronunciation) or by using some method of letter to sound rule. And also, we will train letter-to-sound models from such a lexicon to predict the pronunciation of unknown words. Such resources may not be available in many languages. Thus, we have prepared a pronunciation lexicon which contains a certain number of Amharic words with their pronunciation which is used for the text analysis purpose and found in **Appendix E**. Amharic pronunciation lexicon is mainly used for determining the pronunciations of these words but preparing a lexicon for all of the existing Amharic words or it is impossible to list all words in a natural language for general text to speech we need to provide something to pronounce out of vocabulary words thus if the word is not found in the lexicon we have used a letter to sound rule for unknown words. The letter to sound rules is used as a backup when the word is not explicitly listed within the lexicon.

The lexicon structure that is basically available in festival takes both a word and part of speech tagging to find the given pronunciation, but in our case the part of speech is not considered and represented by nil. The prepared letter to sound rule is one to one since Amharic language is a phonetic language, which means the way Amharic orthographic characters written is very similarly to the way they are spoken and the rule is found in **Appendix B**. For instance to traumatize the pronunciation of the letters "ch" word at the start in English we have a tendency to might right two rules like this

(# [ch] r => k) (# [ch] => ch)

Defining a Set of Question File for Amharic

Our question set is prepared based on the similarities between the place and manner of articulation for segmental context. Since it is language dependent, we cannot use a question file which is prepared for another language. The question set is prepared based on the language structure for Amharic and it has yes/no answer found in **Appendix C**. In this work our question set consists five positions the positions are, LL (left of the left phone to the current phoneme), L (left to the current phoneme), C (the current phoneme), R (right to the current phoneme), RR (right of the right phone to the current phoneme).

4.2 System Architecture

We have designed the architecture of a text to speech synthesizer for Amharic. Our architecture has training and synthesis phase. The training phase a text corpus is used passes through the text analysis process (tokenization, normalization and linguistic features extraction) and the extracted features are used as an input for the duration model, from the speech corpus acoustic features are extracted and used as an input for the acoustic model with the linguistic features and duration information generated by the the duration model and the output of the acoustic model is used as input for the vocoder to generate the final speech. The synthesis phase the given input text passes through the text analysis process (tokenization, normalization, linguistic features extraction) and linguistic features are extracted. The extracted features are then used as an input for duration model and the model is trained, acoustic model training, speech parameter generation and lastly reconstructing waveform.

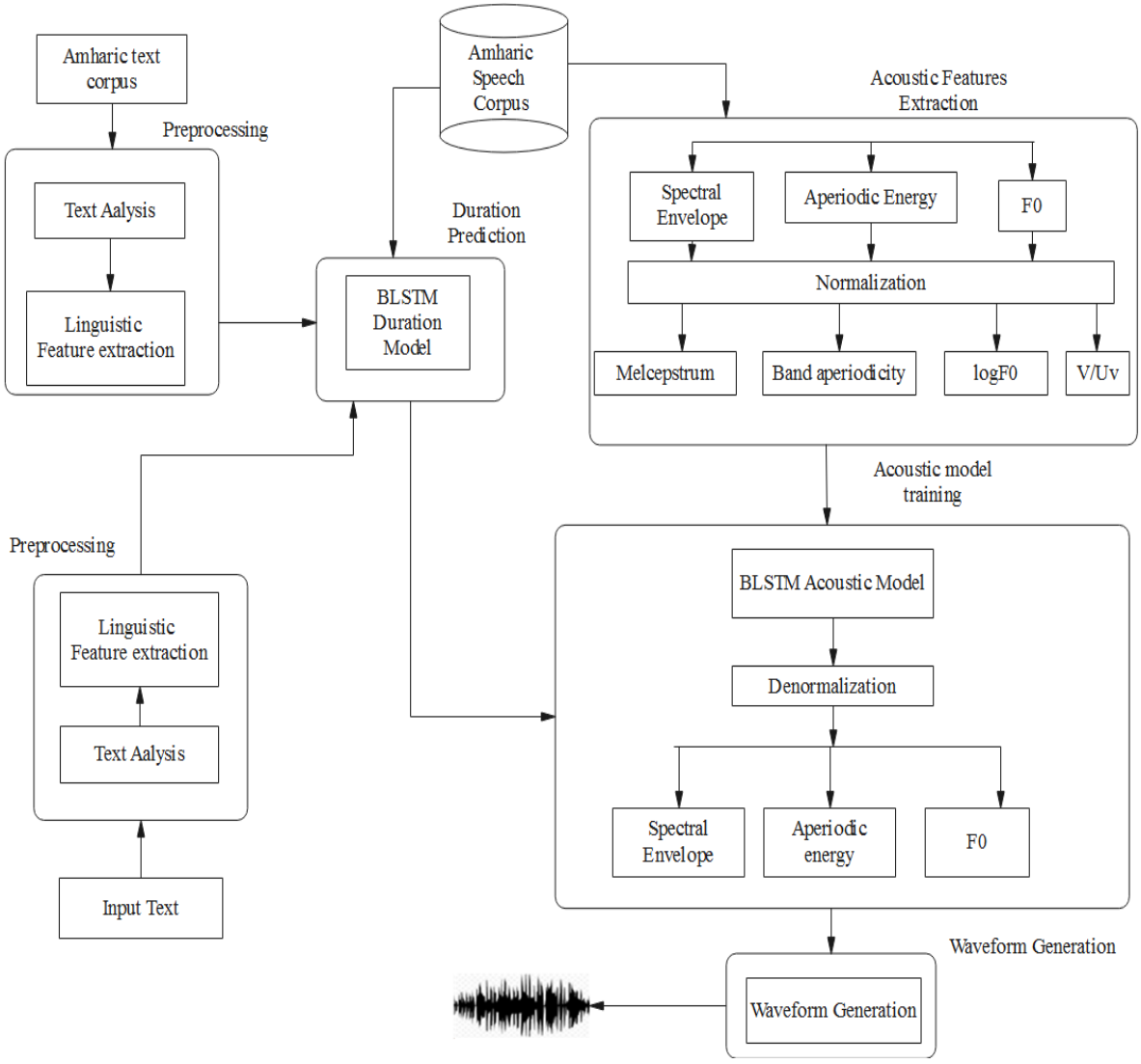

Figure 4.1 The overall architecture of the proposed system

4.3 Description of the Overall Architecture

In this work there are two main phase which are the training and synthesis stage. Both the training and the synthesis stage contains their own processes.

4.3.1 The Training Stage

Before we have trained the models (the duration model and the acoustic model) defining the structure of the models is necessary. Therefore, for this work we have defined a bidirectional long short term memory based acoustic and duration model based on the following value the configuration is found in **Appendix F**.

The training stage have different processes which includes the text analysis, the linguistic feature extraction, acoustic feature extraction, duration model training, acoustic model training, speech parameter generation and waveform generation. Each of the processes will be discussed below.

Preprocessing

Text to speech synthesis is a multipart process that must be broken down into modules. Preprocessing is one of these modules. Preprocessing enables a sequence of texts to be converted into linguistic features. Preprocessing has three goals: tokenization normalization and linguistic features extraction. Consequently, preprocessing is essential, since it affords a good representation of Amharic texts which can be used for the text to speech synthesizer.

Text Analysis

For our BLSTM based TTS systems we've got different modules and also the text analysis module is one in all them and that we have used this module to preprocess the text data and to urge a preprocessed tokenized and normalized text. But so as to make a voice for Amharic language the building of a pronunciation lexicon and building a phone set for Amharic language is required, since the phonological structure of one language is differ from the opposite language, we've prepared the pronunciation dictionary and also the phone set supported the structure of the language.

Tokenization and Text Normalization

The vital stage in text processing is that the initial tokenization of text and text normalization. Thus, our text analysis modules contains the tokenization and text normalization. Since in Amharic language whitespace and punctuation marks are used, we have used them as an honest feature for the tokenization process. Our tokenization process splits the sentence into a token by employing a white space as a delimiter to know the boundary which is found between the words.

After doing the tokenization process the following step is to try and do text normalization. For this work we've prepared a text normalizer supported the language, which convert them into their full orthography. The text analysis module is additionally accountable for converting

input text into their full orthographic form. For instance, numbers and acronyms should be converted into words sequence. The limitation here is just the numbers up to 1000 are considered.

```
Input: Amharic text or sentence
Output: Amharic tokenized words
Let new be a string which set to be null
Let tokens be a list which set to be null
Read list
If file in read endswith '::', '!', or '?'
Put each sentence in new
For file in new
Divide each word using white space
Put each word in tokens
return list of tokenized words
```

Algorithm 4.1 Amharic sentence tokenization

```
Input: Amharic text or sentence
Output: Amharic normalized words
Read file
A) when the number length equal to one or when one number is
present, between [0 9]synthesis the corresponding token.
B)when the number length equal to two or when two numbers meet
together less than one hundreds,
when 1 and 0 come together, list aser, else list asera plus the
next number.
when 2 and 0 come together, list haya, else list haya plus the
next
when 3 and 0 come together, list selasa, else list selasa plus
the next number.
when 9 and 0 come together, list zetena, else list zetena plus
the next number.
C) when the number length equal to three or when three numbers
meet together less than thousand ,if 1 plus just zeros, list
meto, else list A plus meto plus B.
D) when the number length equal to four or when four numbers
meet together less than ten thousand ,if 1 plus just zeros ,
list "Shi", else list A plus Shih plus C.
return list of normalized words
```

Algorithm 4.2 Text normalization for Amharic

Linguistic Feature Extraction

After doing the method text analysis, the normalized text were accustomed extract linguistic features and processed into a more convenient or suitable representation called label. During this stage the normalized word strings from the text analysis module are taken and produce a pronunciation for every word. The foremost important component here may be a pronunciation dictionary. During this work, the letter to sound rule is employed additionally to urge the pronunciation of unknown Amharic words since our dictionary alone end up to be insufficient, because running text always contains words that don't appear within the dictionary.

In order to extract the linguistic features, we have used the phone set, the question file set and the pronunciation dictionary those are already prepared for the language specifically since they have specific information about the language.

In order to teach our BLSTM based mostly TTS synthesis, wealthy contexts are also are used as input options, that contain the binary options for categorical contexts, e.g. phone labels and numerical features for the numerical contexts. Therefore, we have generated contextual features by analyzing the textual transcriptions of the corresponding recorded speech signal in order to represent the linguistic complexities that must be derived into speech signal. The extracted linguistic features are composed of a set of contextualized prosodic and phonetic features. These linguistic features includes phoneme, syllable, word, phrase and sentence-level features. For our work the extracted features are information about stressed syllables, position of the phoneme inside the current syllable, type of the current phoneme, type of the next phoneme, the position of the syllable in the word, position of word in phrase, number of syllables in word, etc. are included in these features.

The extraction of linguistic feature is done on both the training and synthesis phase. Then extracted linguistic features are used as input for our duration model. Since the outputs are with state-level alignment. We have to processes it further in order to make those input features to be suitable for our bidirectional long short term memory based model features because the neural networks require numeric features. We have used the min-max normalization to normalize the extracted linguistic features to the range of [0.01 0.99]. We have used our question file to convert the full context labels into binary and/or numerical

features for vectorization. Therefore, the labels are converted into vectors of binary and continuous features for the input of the bidirectional long short term memory.

```
Input: Amharic tokenized and normalized words
Output: Linguistic features
for each phoneme in each sentence
define the position of current phoneme
define the position of phoneme next to current phoneme in syllable
define the position of phoneme previous to current phoneme within syllable
define the position of phoneme next of next to current phoneme within syllable
define the position of phoneme previous of previous to current phoneme within syllable
define the stress of phoneme previous of previous to current phoneme within syllable
define the accent of phoneme previous of previous to current phoneme within syllable
define the stress of current phoneme
define the accent of current phoneme
end for
for each syllable in each sentence
define the position of syllable within word
define the position of syllable within phrase
define the stress of syllable within word
define the accent of syllable within word
define name of the vowel of syllable
define length of syllable
define length of word
define number of syllable within phrase
define number of word within phrase
define number of total syllable
define number of total word
define number of total phrase
end for
add all the defined values
return list of linguistic features
```

Algorithm 4.3 Linguistic feature extraction for Amharic

Acoustic Feature Extraction

During the training phase we have a speech database which contains the speech recordings in a wave file format. Out of the recordings of the database we have extracted different acoustic features by using a world vocoder. By taking the raw speech and, by windowing it, the vocoder extracts many acoustic frames composed of features which describe the speech signal in a more convenient way, which means having good mathematical properties. The extracted raw features are spectral envelope, aperiodic energy and F0. Before the extracted acoustic features are given as an input for bidirectional long short term memory based acoustic model we have normalized them to zero mean and unit variance normalization. After normalization the normalized features are melcepstrum, band aperiodicity and logF0. In addition, information whether the currently observed frame is voiced or unvoiced, is also added as one binary feature V/UV and their corresponding delta and delta-delta features. Dynamic features (both delta and delta-delta coefficients of all the three categories of static acoustic features) are also calculated and used in target feature vector. Finally the normalized acoustic features are used as input for our BLSTM based acoustic model.

Bidirectional Long-Short Term Memory Based Duration Model Training

Our Amharic TTS system consists of two BLSTM based models. The first one is, BLSTM based duration model and the second one is BLSTM based acoustic model. In this work a separate durational model must be trained first, followed by acoustic model training. The bidirectional long short term memory neural network has two forward-backward recurrent layers topped with one full-connected layer. For the recurrent layers, there are 512 forward neurons and 512 backward neurons.

Our BLSTM based duration model takes linguistic features generated by the text analysis and linguistic feature extractor modules as input, and learns the proper duration information by updating weights. By using the speech corpus speech labeling is done using the HTK tool. Then the extracted speech label is aligned with the linguistic features using the duration model. Since to train the bidirectional long short term memory based acoustic model, the time information is added to linguistic features set, which consists of beginning time and ending time of the phoneme. Thus to do this, the linguistic features and speech have to be aligned during training. Sample labeled speech is found in **Appendix D.**

The output of this network is then feed to the acoustic model with the linguistic feature which was generated by the front end.

Bidirectional Long-Short Term Memory Based Acoustic Model Training

Our BLSTM acoustic model is trained to map the input linguistic features and the associated duration features into acoustic features which was generated by the world vocoder. The linguistic features produced through the frontend. Linguistic features (sequence binary vectors) were normalized in the range of [0.01, 0.99] before passing to input layers of BLSTM. We use the BLSTM to realize linguistic to acoustic feature mapping. The BLSTM neural network has two forward and backward recurrent layers covered with one full connected layer. For the recurrent layers, there are 512 forward neurons and 512 backward neurons.

The linguistic features and their duration information with their acoustic features is generated by the trained acoustic model they will used as an input the parameter predictor to get a generated smoothed parameter trajectories.

Here, the outputs of BLSTM acoustic model are normalized appropriately to make it suitable for our vocoder so denormalization were done. By using the zero mean and unit variance we have change the acoustic features back to their previous raw vocoder features. That means the extracted features are converted into their previous features form. The melcepstrum with delta and delta-delta converted into spectral envelope, the Band aperiodicities with delta and delta-delta converted into Band aperiodicities into and Interpolated log (F0) and voiced/unvoiced with delta and delta-delta logf0. For our work we have used the neural network based parameter prediction. Parameter generation is used to forecast acoustic feature parameters founded on the result of the text analysis module and the trained acoustic model. Thus, we applied the maximum likelihood parameter generation (MLPG) algorithm applied to the output features to produce smoothed parameter trajectories. By setting the anticipated output options from the BLSTM as mean vectors and pre-computed variances of output options from all coaching information as variance matrices, the speech parameter generation algorithmic program generates sleek trajectories of speech parameter options that satisfy each the statistics of static and dynamic options.

Waveform Generation

In the process of waveform generation the input is the acoustic features and output is the speech waveform. Once the acoustic features are predicted by the bidirectional long short term

memory acoustic models. The final normalized acoustic features are taken as input to the WORLD vocoder to synthesize the waveform.

```
Input: Denormalized acoustic features
Output: Amharic speech waveform
While acoustic model generates the acoustic features
Then do speech parameter generation
if speech parameter is generated then
end for while
return a speech waveform
```

Algorithm 4.4 Waveform generation

4.3.2 The Synthesis Stage

At synthesis time we have only an input text, the input text is inserted, during the speech synthesis stage, we first do text analysis to generate linguistic features. The text analysis module is used to do tokenization and text normalization and then the normalized text is used to extract linguistic features and the extracted features are used as an input for the duration model. The durations for each phone are first predicted using the duration model using the pre-trained model. Then after, the duration model is used to estimate the timestamps of each phoneme using the knowledge of the duration of each phone to be synthesized. The linguistic features alongside timestamps are then used because the input for the BLSTM acoustic model to get corresponding compressed acoustic features, which include MC, log F0, BAP. And then the acoustic features are converted back to their previous form of acoustic features SP, F0, and BAP. Maximum likelihood parameter generation using pre-computed variances from the training data is applied to the output features for synthesis. Lastly, the acoustic features are taken as an input for the vocoder to generate speech signal.

```
Input: Amharic word or sentence
Output: Amharic speech waveform
Read file
Extract linguistic features
Predict the durations for each phone
```

```
if duration of each phone is predicted and used as an input
for acoustic model
then generate corresponding compressed acoustic features
Reconstruct speech parameters
Convert the acoustic features to their previous form
End
return a speech waveform
```

Algorithm 4.5 The Synthesis Stage

Chapter Five: Evaluation Result and Discussion

5.1 Introduction

Speech synthesis is assessed both objectively and subjectively. In objective evaluations, synthetic speech is usually compared to reference speech, or features derived from synthetic speech are used to measure synthesis quality. In subjective evaluations, volunteers are asked to listen two systems, and rate or compare them on a variety of metrics. Consequently for our work we have conducted objective evaluation and subjective evaluation. BLSTM and DNN based duration and acoustic model are trained from the data, so we have conducted the evaluation for both.

As we have discussed before, our system uses deep learning based model, specifically bidirectional long-short term memory using the open source merlin, world and festival tool and we have used operating system called Ubuntu 18.04 as a working environment. We have installed the tools on Ubuntu 18.04 using Virtual machine and we have trained and tested our corpus on it after some preparation. We have used Notepad++ editor for writing C++, python and the C codes, GCC 6 compiler for compiling a code, free mp3 cutter for speech segmentation, Festival speech tool for tokenization and normalization and for linguistic feature extraction, world vocoder for acoustic feature extraction and merlin speech synthesis toolkit for the training and synthesis purpose. By using the above tools we have prepared the necessary data.

5.2 Data Collection and Preparation

Corpus preparation is one of the furthermost significant procedures to make a high-quality speech synthesis system. To have a good training dataset, first we need to collect sufficient amount of data. Speech corpus is only available for Amharic automatic speech recognition purpose but there is no any dataset which can be used for text to speech synthesis purpose. The available datasets are not suitable for speech synthesis since the datasets are mostly noisy. For our work we found that an audiobook of Amharic bible read by a male Amharic speaker online from YouTube and the recorded audio bible have relatively a good recording condition. Thus, to begin our data collection, we have selected a portion of the audio bible. The database contains a sound spoken by a male speaker, to build the TTS system. Apiece wave file comprises one spoken sentence. We have used 600 audio files and they have already changed

into waveform and their corresponding text file is also prepared. The corresponding text files are prepared manually. The collected data corpus will be used for training, testing and validation. From the total of 600 sentence, 90% have used to train the neural network model and 5% sentences will be used for testing and 5% for validation purpose. When we prepare the data, we have tried to include all of Amharic phonemes in order to make the synthesis better but there are some Amharic phonemes which are not included in the audio bible.

5.3 Test Result

Speech synthesis can be evaluated using subjective and objective evaluation and experiments were conducted using both of these evaluation methods. In subjective evaluations, humans are asked to listen to one or multiple systems, and rate or compare them.

5.3.1 Objective Evaluation and Discussion

We first conducted the objective measure to evaluate the performance of the two models. To evaluate the performance of bidirectional long short term memory based and deep neural network based speech synthesis system objectively, Melcepstral Distortion (MCD) (dB). Mel cepstral distortion is a weighted root mean square error between the predicted speech and the original speech.

Table 5.1 Comparison of objective results using Mel-Cepstral Distortion (MCD)

Model	MCD (db)
BLSTM	4.68
DNN	4.7

From the objective evaluation we can see that the bidirectional long short term memory outperforms the deep neural network. In this objective evaluation case the low value means better result. The melcepstral distortion shows that the bidirectional long short term memory synthesized the more speech compared with the original recording than the deep neural network based one.

5.3.2 Subjective Evaluation and Discussion

Our subjective evaluation evaluates two systems the BLSTM based text to speech synthesis and the DNN based speech synthesis. To assess and conclude which one is better for our text to speech synthesis.

For the evaluation of the synthesized voice quality, we carried out formal listening tests for the subjective evaluation. The tests were requires the listeners to rank the voice quality in terms of naturalness and intelligibility using a MOS like scoring to assess the quality of the synthesized speech produced from our system. We have used to assess the synthesized speech quality in terms of intelligibility and naturalness. The test was carried out by synthesizing a set of 10 sentences that have been selected from the speech corpus that has been set aside isolated from the training set used for constructing synthesis database. The wave files were at 16 kHz and 16 bits. The intention for choosing the sentences for which we have also the original speech waveforms spoken by our speaker is that we also use the original recordings in our tests to ensure the reliability of our test results. The same training data set as BLSTM is used for DNN training.

In the MOS test, 10 volunteer were selected using voluntary response sample**.** The subjects were instructed to rate the sentences on a scale of 1 to 5 where 1 is very poor and 5 is excellent. The subjects listened the sentences using headphones. In the MOS test we evaluated the quality of the two systems which are BLSTM based speech synthesis and DNN based speech synthesis. The acoustic DNN consists of 4 layers of hidden units where each layer contains 512 neurons. The result of subjective evaluation is shown below in figure 5.1.

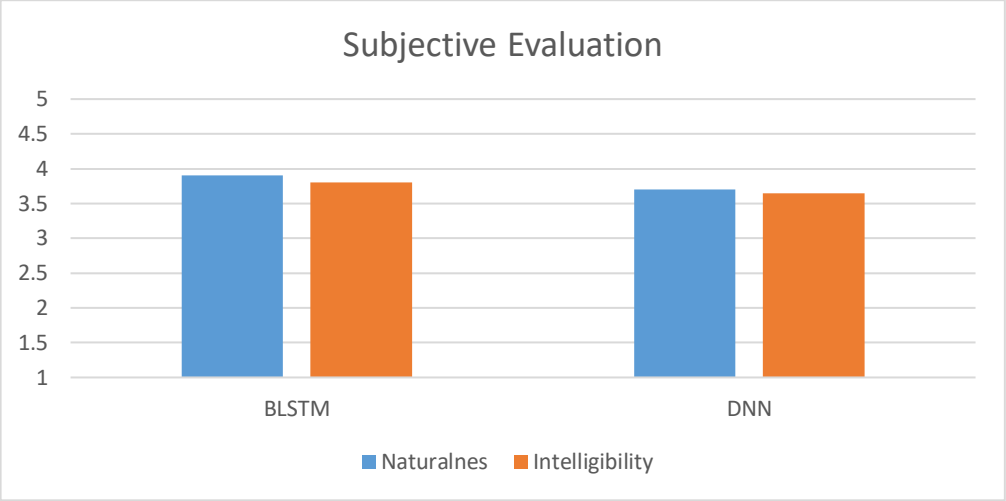

Figure 5.1 Subjective evaluation of the proposed system and DNN system

Sentences used to do subjective evaluation

ወደሚያድሩበትም ወደ ሚካ ቤት መጡ

እስራኤላውያንም በእርሱ መሪነት ከኮረብታው ላይ አብረውት ወረዱ

የፍልስጤማውያን ገዢዎችም በሙሉ በዚያ ነበሩ

ከርሱ ጋር የነበሩ ሰዎች መነሳቱን ይመሰክሩ ነበር

ይህም ቢሆን ከአለቆች መካከል እንኳን ሳይቀር ብዙዎች በርሱ አመኑ

ለአምልኮ ወደ በሃሉ ከወጡት መካከል የግሪክ ሰዎችም ነበሩ

ሰለሞን ባለ ብዙ ምሰሶ አዳራሽ የተባለ ቤት ሰራ

በልክ በልኩ በተቆረጠና አምሮ በተስተካከለ ምርጥ ድንጋይ የተሰራ ነበር

በእቃ ማስቀመጫው ጫፍ ላይ ግማሽ ክንድ የሆነ ዙሪያ ክብ ነበረበት

አናቱ ላይ ያሉት ጉልላቶች የሱፍ አበባ ቅርፅ ነበራቸው

Chapter Six: Conclusion and Recommendation

6.1 Conclusion

This paper proposed to use bidirectional long short term memory to improve the quality of synthesized speech and to get speech that is both intelligible and fairly natural sound. The Amharic speech synthesis system was initially trained using Amharic audio bible data by a male speaker with a good recording quality. The TTS synthesis system developed in this work exhibited an ability to synthesize understandable speech though it had no associated part-of-speech tagger, post-lexical rules and word stress information but only depended on manual lexicon entries, letter-to-sound rules simple syllabification rules, and the phone set. Both the objective and subjective evaluation shows that the bidirectional long short term memory based one is both intelligible and fairly natural sounding than the deep neural network based one. Therefore, from our work we can conclude that bidirectional long short term memory system outperforms the deep neural network due to its ability to capture sequential information and can adaptively memorize the context in a sentence for Amharic language.

6.2 Contribution of This Work

- We have developed a lexicon and letter-to-sound rules for Amharic language.
- We have prepared our own corpus which can be used for text to speech synthesis to create voices with good naturalness and intelligibility.
- Because of the sequential nature of text dependent features, we proposed BLSTM based model to get better result.
- We have addressed the issues of acoustic modeling, by using of bidirectional long short term memory for acoustic modeling.

6.3 Future Work

The issues listed below are some among many that should be looked into in the direction of developing better TTS synthesis system or improving on the already built Amharic TTS synthesis system:

- Associating part-of-speech tagger, post-lexical rules and word stress information to check whether they can enable to get the most understandable speech.
- Including homograph disambiguation since Amharic language have many disambiguate words.

- Preparing an Amharic corpus which can be used for Amharic text to speech synthesis.
- In this work the TTS system used a mono-lingual single speaker data. However, sometimes, it is more required to model several variations in a single TTS system. These variations may be within the kind of multiple speakers, dialects, languages, expressions, voice qualities, etc.
- Checking whether the number of hidden layer may or not affect the synthesized speech.
- Trying to use the other deep learning based models in order to assess their performance whether they give better result or not for Amharic language.
- Adding speaker adaptation and voice conversion for Amharic based on bidirectional long short term memory.
- Besides, our experiments were conducted only on male data, thus additional experimentation is needed to determine which results generalize across gender as well.

References

[1] R. Cole, J. Mariani, H. Uszkoreit and a. G. Varile, "Survey of the State of the Art in Human Language Technology," Cambridge University Press and Giardini., 1997.

[2] Mustafa, Hussain and M. Hamad, "Arabic Text-To-Speech Synthesizer," in *Student Conference on Research and Development*, Khartoum, Sudan, 2011.

[3] S. Lukose, Savitha and S. Upadhya, "Text to Speech Synthesizer-Formant Synthesis," in *International Conference on Nascent Technologies in the Engineering Field (ICNTE-2017)*, 2017.

[4] R. Vargas, A. Mosavi and L. Ruiz, "Deep learning: A Review," in *Advances in Intelligent Systems and Computing*, 2017.

[5] Y. Lee, T. Kim and Soo-Young Lee, "Voice Imitating Text-to-Speech Neural Networks," Korea Advanced Institute of Technology, Seoul, Korea, 2018.

[6] M. S. Al-Radhi, T. G. Csapo and G. Nemeth, "Deep Recurrent Neural Networks in Speech Synthesis Using a Continuous Vocoder," in *International Conference on Speech and Computer*, Hatfield_UK_S, 2017.

[7] R. W. Amanda Wimsatt et al, "Amharic Language and Culture Manual," Texas State University, 2011.

[8] Mustapha, Oloko-Oba, T. Ibiyemi and E. O. Samuel, "Text-to-Speech Synthesis Using Concatenative Approach," *International Journal of Trend in Research and Development,*, vol. 3, pp. 459-462, 2016.

[9] Al-Said, Ghadeer and a. M. Abdallah, "An Arabic Text-To-Speech System Based on Artificial Neural Networks," *Journal of Computer Science,* vol. 3, pp. 207-213, 2009.

[10] Hailemariam, A. Korssa and Sebsibe, "Syllabification Design and TTS System for Afaan Oromo Using Unit Selection," *HiLCoE Journal of Computer Science and Technology,* vol. 2, pp. 1-8, 2013.

[11] Achanta, Sivanand, Gangashetty and S. V, "Deep Elman Recurrent Neural Networks for Statistical Parametric Speech Synthesis," *Speech Communication,* 2017.

[12] Y. Fan, Y. Qian, F. Xie and a. F. K. Soong, "TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks," in *INTERSPEECH Microsoft Research*, Asia, Beijing, China, 2014.

[13] H. Zen and a. H. Sak, "Unidirectional long short-term mem-ory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[14] B. X. W. Wang and and Shuang Xu, "Gating recurrent mix-ture density networks for acoustic modeling in statisti-cal parametric speech synthesis," in *in Proc. IEEE Int. Conf.on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[15] Z.Wu, O.Watts and S. King, "Merlin: An open source neural network speech synthesis system," in *Proc. SSW*, Sunnyvale, USA, 2016.

[16] M. Morise, F. Yokomori and K. Ozawa, "WORLD: A vocoder-based high quality speech synthesis system for real-time applications," in *IEICE Transactions on Information and Systems*, 2016.

[17] Black and Lenzo, "Building Voices In The Festival Speech Synthesis System," 2002. [Online]. Available: http://Festvox.org/bsv. [Accessed 17 9 2019].

[18] A. Balyan, S. S. Agrawal and A. Dev, "Speech Synthesis: A Review," *International Journal of Engineering Research & Technology (IJERT),* pp. 57-75, 2013.

[19] F. J, "Speech Analysis, Synthesis, and Perception," in *Springer-Verlag*, Berlin-Heidelberg-New York, 1972.

[20] F. J, R. L. (editors), Dowden and H. a. Ross, " Speech Synthesis," Pennsylvania,, 1973.

[21] S. M, " A Brief History of Synthetic Speech. pp.," *Speech Communication,* vol. vol. 13, pp. 231-237, 1993.

[22] K. D, "Review of Text-to-Speech Conversion for English," *Journal of the Acoustical Society of America, JASA , pp.*, vol. vol. 82 (3), pp. 737-793, 1987.

[23] S. Lemmetty, "Review of Speech Synthesis Technology," Espoo, March 30, 1999.

[24] D. Klatt, "Review of text-to-speech conversion for English," *Journal of the Acoustical Society of America,* vol. vol 82(3), 1987..

[25] T. Styger and a. E. Keller, "Formant Synthesis," *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges,* pp. 109-128, 1994.

[26] "Chapter 2: A Survey on Speech Synthesis Techniques," 2011, pp. 17-31.

[27] S. Panda, A. Nayak and S. and Rai, "A survey on speech synthesis techniques in Indian languages," *Multimedia Systems 26,* p. 453–478 , 2020.

[28] A. Indumathi and D. E. Chandra, "Survey On Speech Synthesis," *Signal Processing: An International Journal (SPIJ)*, vol. Volume (6), no. issue (5), 2012.

[29] S. Kayte, M. Mundada and a. J. Gujrathi, "Hidden Markov Model based Speech Synthesis: A review," *International Journal of Computer Applications (0975 – 8887)*, Vols. Volume 130-No.3, November 2015.

[30] A. W. Black, H. Zen and a. K. Tokuda, "STATISTICAL PARAMETRIC SPEECH SYNTHESIS," in *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Honolulu, HI, USA, 2007 .

[31] Y. J, A. Wang, Y. Liu, H. Li, J.H. and L. J, "Deep learning theory and its application in speech recognition," *Commun. Countermeas*, pp. 1-5, 2014.

[32] Yu, L. Deng and Dong, Deep Learning: Methods and Applications, Redmond, 2014.

[33] H. Zen, A. Senior and M. Schuster, "STATISTICAL PARAMETRIC SPEECH SYNTHESIS USING DEEP NEURAL NETWORKS," in *In Proc. ICASSP*, 2013.

[34] K. Tokuda, T. Kobayashi, T. Masuko, T. Kobayashi and T. Kitamura, "Speech parameter generation algorithms for HMMbased speech synthesis," *In Proc. ICASSP*, pp. 1315-1318, 2000.

[35] Gibson, J. Patterson and Adam, Deep learning: A PRACTITIONER'S APPROACH, Highway North, Sebastopol: O'Reilly Media, 2017.

[36] G. Antonio and P. Sujit, Deep learning with keras: Implement Neural Networks with Keras on Theano and Tensorflow, BIRMINGHAM- MUMBAI: Packt Publishing, 2017.

[37] A. Karpov, R. Potapova and L. Mporas, "Speech and Computer," in *19th International Conference, SPECOM*, Hatfield, 2017.

[38] "Ethnologue: Languages of the World," 2016. [Online]. Available: http://www.ethnologue.

[39] ፕሮፉሰር and ጌ. አ. ረ/, Modern Amharic Grammer, ዘመናዊ የአማርኛ ሰዋሰው በቀላል አቀራረብ, አዲስ አበባ፣ ኢትዮጵያ: አልፋ አሳታሚዎች ታተመ, 2001 ዓም.

[40] P. Roach, "A little encyclopedia of phonetics," University of Reading, UK, 2002.

[41] Gamback, H. Seid and Bjorn, "A Speaker Independent Continuous Speech Recognizer for Amharic," in *INTERSPEECH*, Lisbon Portugal, 2005.

[42] Gamback, S. Eyassu and Bjorn, "“Classifying Amharic News Text Using Self Organizing Maps," *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, p. 71–78, June 2005.

[43] X. Huang, A. Acero and H. Hon, "Spoken Language Processing. Prentice Hall," Upper SaddleRiver, New Jersey, 2001.

[44] S. Myers and B. B. Hansen, "The origin of vowel-length neutralization in vocoid sequences," *Phonology 22*, p. 317–344, 2005.

[45] Sebsibe, Kishore, Black, Kumar and Sangal, "Unit Selection Voice for Amharic using Festvox," in *5th ISCA Speech Synthesis Workshop,*, Pittsburgh,, 2005.

[46] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assesment," *In Proceedings of IEEE Pacific Rim Conference on Communications*, vol. volume 1, p. pages 125–128., 1993.

[47] S. a. K. King, "The Blizzard Challenge," in *In Proceedings of Blizzard Challenge Workshop*, 2016.

[48] R. S. Raju, P. Bhattacharjee, A. Ahmad and M. S. Rahman, "A Bangla Text-to-Speech System using Deep Neural Networks," in *International conference on Bangla speech and language speech processing*, Saylhet.Bangladesh, 2016.

[49] Laine Berhane, "Text-to-Speech Synthesis of the Amharic Language," unpublished, Addis Ababa University, Ethiopia, , 1998.

[50] Nadew Tademe, "Formant-based speech synthesis for Amharic vowels," unpublished, Addis Ababa University, Ethiopia, 2008.

[51] Yibeltal Tefera, "Formant-Based Speech Synthesis: A Case of Amharic Words," unpublished, Addis Ababa University, Ethiopia,, 2008.

[52] Bereket Kasaye, "Developing a Speech Synthesizer for Amharic using Hidden Markov Model,," unpublished, Addis Ababa University, Ethiopia,, 2008.

[53] Alula Tafere, "A Generalized Approach to Amharic Text-To-Speech (TTS) Synthesis System," Addis Ababa University,Ethiopia,, 2010.

[54] Eyob Bayou, "Concatenative speech synthesis for amharic using unit selection method," unpublished,, Addis Ababa University, Ethiopia,, 2011.

[55] Mulat Shiferaw, "Syllable-Based Text-To- Speech Synthesis (TTS) For Amharic," unpublished, Addis Ababa University, Ethiopia,, 2012.

[56] Bahiru Demessie, "Syllable-Based Amharic Speech Synthesis (TTS) Using HMM,," unpublished, Addis Ababa University, Ethiopia, , 2017.

[57] Habtamu Abate, "A Speaker Independent Text-to-Speech Synthesis (TTS) For Amharic Language Using Hidden Markov Model," unpublished, Addis Ababa, Ethiopia, 2018.

[58] E. Cooper, "Text-to-Speech SynthesisUsing FoundData for Low-Resource Languages," 2019.

[59] R. Weide, "The CMU pronunciation dictionary release 0.6," 1998.

Appendix A: Amharic Phone Set

```
(defPhoneSet
aau
;;; Phone Features
(;; vowel or consonant
(vc + -)
;; vowel length: short long dipthong schwa
(vlng s l d a 0)
;; vowel height: high mid low
(vheight 1 2 3 0)
;; vowel frontness: front mid back
(vfront 1 2 3 0)
;; lip rounding
(vrnd + - 0)
;; consonant type: stop fricative affricate nasal lateral approximant
(ctype s f a n l r 0)
;; place of articulation: labial alveolar palatal labio-dental
;;
dental velar glottal
(cplace l a p b d v g 0)
;; consonant voicing
(cvox + - 0)
)
;; Phone set members
(
(pau  - 0 0 0 0 0 0 -)
(he   - 0 0 0 + f v -)
(le   - 0 0 0 + l a +)
```

(me - 0 0 0 + n l +)

(se - 0 0 0 + f a -)

(re - 0 0 0 + l a +)

(sxe - 0 0 0 + f p -)

(qe - 0 0 0 + s v -)

(be - 0 0 0 + s l +)

(te - 0 0 0 - s a -)

(ce - 0 0 0 - a p -)

(ne - 0 0 0 - n a +)

(nxe - 0 0 0 - n p +)

(e + s 2 2 - 0 0 0)

(u + s 3 3 + 0 0 0)

(ii + s 3 1 - 0 0 0)

(a + s 1 2 - 0 0 0)

(ie + s 2 1 - 0 0 0)

(ix + s 3 2 - 0 0 0)

(o + s 2 3 - 0 0 0)

(ke - 0 0 0 – s v -)

(we - 0 0 0 – f l +)

(ze - 0 0 0 – f a +)

(zxe - 0 0 0 – f p +)

(ye - 0 0 0 – l p +)

(de - 0 0 0 – s a +)

(ge - 0 0 0 – s v +)

(je - 0 0 0 - a p +)

(ju - 0 0 0 – s v +)

(jii - 0 0 0 – s v +)

(ja - 0 0 0 – s v +)

(jie - 0 0 0 – s v +)

(jix - 0 0 0 - s v +)

(jo - 0 0 0 - s v +)

```
(jua  - 0 0 0 – s  v +)
(txe  - 0 0 0 – s  a -)
(cxe  - 0 0 0 – a  p -)
(pxe  - 0 0 0 – s  l -)
(xxe  - 0 0 0 - f a -)
(fe   - 0 0 0 - f b -)
(pe   - 0 0 0 - s l -)
(h#  - 0 0 0 0 0 0 -)
(brth - 0 0 0 0 0 0 -)
)
)
(PhoneSet. Silences '(pau h# brth))
(define (aau_amharic_amharic_voice::select_phoneset)
  "( aau_amharic_amharic_voice::select_phoneset)
Set up phone set for aau_amharic"
  (Parameter. Set 'PhoneSet 'aau_amharic)
  (PhoneSet.select 'aau_amharic)
)

(define (aau_amharic_amharic_voice::reset_phoneset)
  "(aau_amharic_amharic_voice::reset_phoneset)
Reset phone set for aau_amharic."
  t
)

(provide 'aau_amharic_amharic_voice_phones)
```

Appendix B: Sample Amharic Letter to Sound Rule

```
(lts.ruleset
; Name of ruleset
amharic)
(set! allowables
     '((e _epsilon_ e)
       (ii _epsilon_ ii)
       (o _epsilon_ o)
       (u _epsilon_ u)
       (ix _epsilon_ ix)
       (ie_epsilon_ ie)
       (he _epsilon_ h)
       (le_epsilon_ l)
       (me _epsilon_ m)
       (se _epsilon_ s)
       (re _epsilon_ r)
       (sxe _epsilon_ sx )
       (qe _epsilon_ q)
       (be _epsilon_ b )
       (te _epsilon_ t)
       (ce_epsilon_ c)
       (ne _epsilon_ k )
       (nxe _epsilon_ nx)
       (ke _epsilon_ k)
       (we_epsilon_ w)
       (ze _epsilon_ z)
       (zxe _epsilon_ zx )
       (ye _epsilon_ y)
       (de _epsilon_ d)
       (je _epsilon_ j)
       (ge _epsilon_ g)
```

```
(txe _epsilon_ tx)
(cxe _epsilon_ cx)
(pxe _epsilon_ px)
(xxe _epsilon_ xx)
(fe_epsilon_ f)
(pe _epsilon_ P)
(# #)))
```

Appendix C: Sample Amharic Question Set

QS "LL-Vowel" {a^*,e^*,ii^*,ix^*,ie^*,o^*,u^*}

QS "LL-Stop"
{be^*,bu^*,bii^*,ba^*,bie^*,bix^*,bo^*,de^*,du^*,dii^*,da^*,die^*,dix^*,do^*,pxe^*,pxu^*,pxii^*,pxa^*,pxie^*,pxix^*,pxo^*,ge^*,gu^*,gii^*,ga^*,gie^*,gix^*,go^*,ke^*,ku^*,kii^*,ka^*,kie^*,kix^*,ko^*,pe^*,pu^*,pii^*,pa^*,pie^*,pix^*,po^*,te^*,tu^*,tii^*,ta^*,tie^*,tix^*,to^*,txe^*,txu^*,txii^*,txa^*,txie^*,txix^*,txo^*,qe^*,qu^*,qii^*,qa^*,qie^*,qix^*,qo^*}

QS "LL-Nasal"
{me^*,mu^*,mii^*,ma^*,mie^*,mix^*,mo^*,ne^*,nu^*,nii^*,na^*,nie^*,nix^*,no^*,nxe^*,nxu^*,nxii^*,nxa^*,nxie^*,nxix^*,nxo^*}

QS "LL-Fricative"
{fe^*,fu^*,fii^*,fa^*,fie^*,fix^*,fo^*,xxe^*,xxu^*,xxii^*,xxa^*,xxie^*,xxix^*,xxo^*,ve^*,vu^*,vii^*,va^*,vie^*,vix^*,vo^*,se^*,su^*,sii^*,sa^*,sie^*,six^*,so^*,sxe^*,sxu^*,sxii^*,sxa^*,sxie^*,sxix^*,sxo^*,ze^*,zu^*,zii^*,za^*,zie^*,zix^*,zo^*,zxe^*,zxu^*,zxii^*,zxa^*,zxie^*,zxix^*,zxo^*}

QS "LL-Liquid" {le^*,lu^*,lii^*,la^*,lie^*,lix^*,lo^*,re^*,ru^*,rii^*,ra^*,rie^*,rix^*,ro^*}

QS "LL-Front"
{ii^*,ie^*,pxe^*,pxu^*,pxii^*,pxa^*,pxie^*,pxix^*,pxo^*,fe^*,fu^*,fii^*,fa^*,fie^*,fix^*,fo^*,me^*,mu^*,mii^*,ma^*,mie^*,mix^*,mo^*,pe^*,pu^*,pii^*,pa^*,pie^*,pix^*,po^*,ve^*,vu^*,vii^*,va^*,vie^*,vix^*,vo^*,we^*,wu^*,wii^*,wa^*,wie^*,wix^*,wo^*}
LL-pau" {pau^*}
QS "LL-SIL" {SIL^*}

QS "LL-brth" {brth^*}

QS "L-Vowel" {*^a-*,*^e-*,*^ii-*,*^ix-*,*^ie-*,*^o-*,*^u-*}

QS "L-Consonant" {*^b-*,*^c-*,*^cx-*,*^d-*,*^f-*,*^g-*,*^h-*,*^j-*,*^k- *,*^l-*,*^m-*,*^n-*,*^nx-*,*^q-*,*^px-*,*^p-*,*^r-*,*^s-*,*^sx-*,*^t-*,*^tx-*,*^v-*,*^w-*,*^y-*,*^z-*,*^zx-*,*^xx-*}

QS "L-Stop" {*^b-*,*^d-*,*^px-*,*^g-*,*^k-*,*^p-*,*^t-*,*^tx-*,*^q-*}

QS "L-Nasal" {*^m-*,*^n-*,*^nx-*}

QS "L-Fricative" {*^f-*,*^xx-*,*^v-*,*^s-*,*^sx-*,*^z-*,*^zx-*}

QS "L-Liquid" {*^l-*,*^r-*}

QS "C-Voiced_lateral" {*-bw+*,*-dw+*,*-gw+*,*-b+*,*-d+*,*-g+*}

QS "C-Voiced_Consonant" {*-bw+*,*-dw+*,*-gw+*,*-b+*,*-d+*,*-g+*}

QS "C-Lateral" {*-bw+*,*-dw+*,*-gw+*,*-b+*,*-d+*,*-g+*}

QS "C-Voiced_Plosive" {*-bw+*,*-dw+*,*-gw+*,*-b+*,*-d+*,*-g+*}

QS "C-Voiced" {*-bw+*,*-mw+*,*-b+*,*-m+*}

QS "C-Unvoiced_Consonant" {*-p+*,*-t+*,*-k+*,*-f+*,*-s+*,*-sx+*,*-c+*}

QS "C-Front_Vowel" {*-a+*,*-e+*,*-i+*}

QS "C-Unvoiced_Fricative" {*-f+*,*-s+*,*-sx+*}

QS "C-Alveolar_Fricative" {*-s+*,*-xx+*,*-z+*}

QS "C-Nasal" {*^m-*,*^n-*,*^nx-*}

QS "C-Vowel" {a^*, e^*, ii^*, ix^*, ie^*, o^*, u^*}

QS "C-Unrounded" {*-e+*,*-ii+*,*-a+*,*-ie+*,*-ix+*}

QS "C-Voiced_Fricative" {*-Z+*,*-Zx+*,*-v+*}

QS "C-Velar" {*-k+*,*-g+*,*-q+*}

QS "C-Central_Vowel" {*-at+*,*-one+*}

QS "C-Back_Vowel" {*-o+*,*-u+*}

QS "C-silences" {*-pau+*,*-wb+*}

QS "C-alveolar_Fricative" {*-s+*,*-z+*,*-xx+*}
QS "C-Fricative" {*-f+*,*-v+*,*-s+*,*-s+*,*-Zx+*,*-sx+*,*-xx+*,*-h+*}
QS "C-Rounded" {*-o+*,*-u+*}
QS "C-Affricate" {*-c+*,*-j+*,*-cx+*}
QS "C-Voiced_Velar" {*-g+*}
QS "C-Glottal" {*-px+*,*-tx+*,*-q+*, *-qua+*,*-xx+*, *-cx+*}
QS "C-silences" {-pau+,-sil+,-brth+}

QS "RR-Vowel" {^*a,^*e,^*ii,^*ix,^*ie,^*o,^*u}
QS "RR-Consonant"
{^*axa,^*axu,^*axii,^*axie,^*axix,^*axo,^*be,^*bu,^*bii,^*ba,^*bie,^*bix,^*bo,^*ce,^*cu
,^*cii,^*ca,^*cie,^*cix,^*co,^*cxe,^*cxu,^*cxii,^*cxa,
^*cxie,^*cxix,^*cxo,^*de,^*du,^*dii,^*da,^*die,^*dix,^*do,^*fe,^*fu,^*fii,^*fa,^*fie,^*fix
,….}

QS "RR-Stop"
{^*be,^*bu,^*bii,^*ba,^*bie,^*bix,^*bo,^*de,^*du,^*dii,^*da,^*die,^*dix,^*do,^*pxe,^*px
u,^*pxii,^*pxa
,^*pxie,^*pxix,^*pxo,^*ge,^*gu,^*gii,^*ga,^*gie,^*gix,^*go,^*ke,^*ku,^*kii,^*ka,^*kie,^*
kix,^*ko,^*pe,^*pu,^*pii,^*pa,^*pie,^*pix,^*po,
^*te,^*tu,^*tii,^*ta,^*tie,^*tix,^*to,^*txe,^*txu,^*txii,^*txa,^*txie,^*txix,^*txo,^*qe,^*qu,^
*qii,^*qa,^*qie,^*qix,^*qo}

QS "RR-Nasal"
{^*me^*,^*mu,^*mii,^*ma,^*mie,^*mix,^*mo,^*ne,^*nu,^*nii,^*na,^*nie,^*nix,^*no,^*n
xe,^*nxu,^*nxii,^*nxa,^*nxie,nxix,nxo}
QS "RR-Fricative"
{^*fe,^*fu,^*fii,^*fa,^*fie,^*fix,^*fo,^*xxe,^*xxu,^*xxii,^*xxa,^*xxie,^*xxix,^*xxo,^*ve,
^*vu,^*vii,^*va,
^*vie,^*vix,^*vo,^*se,^*su,^*sii,^*sa,^*sie,^*six,^*so,^*sxe,^*sxu,^*sxii,^*sxa,^*sxie,^*s
xix,^*sxo,^*ze,^*zu,^*zii,^*za,^*zie,^*zix,^*zo,

^*zxe,^*zxu,^*zxii,^*zxa,^*zxie,^*zxix,^*zxo}

QS "RR-Liquid" {^*le,^*lu,^*lii,^*la,^*lie,^*lix,^*lo,^*re,^*ru,^*rii,^*ra,^*rie,^*rix,^*ro}

QS "RR-Front"

{^*ii,^*ie,^*pxe,^*pxu,^*pxii,^*pxa,^*pxie,^*pxix,^*pxo,^*fe,^*fu,^*fii,^*fa,^*fie,^*fix,^*fo,^*me,^*mu,^*mii,^*ma,^*mie,^*mix,^*mo

,^*pe,^*pu,^*pii,^*pa,^*pie,^*pix,^*po,^*ve,^*vu,^*vii,^*va,^*vie,^*vix,^*vo,^*we,^*wu,^*wii,^*wa,^*wie,^*wix,^*wo}

QS "RR-pau" {^*pau}

QS "RR-SIL" {SIL^*}

QS "RR-brth" {brth^*}

QS "R-Vowel" {*^a-*,*^e-*,*^ii-*,*^ix-*,*^ie-*,*^o-*,*^u-*}

QS "R-Consonant" {*^b-*,*^c-*,*^cx-*,*^d-*,*^f-*,*^g-*,*^h-*,*^j-*,*^k- *,*^l-*,*^m-*,*^n-*,*^nx-*,*^q-*,*^px-*,*^p-*,*^r-*,*^s-*,*^sx-*,*^t-*,*^tx-*,*^v-*,*^w-*,*^y-*,*^z-*,*^zx-*,*^xx-*}

QS "R-Stop" {*^b-*,*^d-*,*^px-*,*^g-*,*^k-*,*^p-*,*^t-*,*^tx-*,*^q-*}

QS "R-Nasal" {*^m-*,*^n-*,*^nx-*}

QS "R-Fricative" {*^f-*,*^xx-*,*^v-*,*^s-*,*^sx-*,*^z-*,*^zx-*}

QS "R-Liquid" {*^l-*,*^r-*}

QS "C--Syl_Front_Vowel" {*|a/C/*,*|e/C/*,*|i/C/*}

QS "C--Syl_High_Vowel" {*|a/C/*}

QS "C--Syl_Vowel" {*|at/C/*,*|a/C/*,*|e/C/*,*|i/C/*,*|o/C/*,*|u/C/*,*|one/C/*}

QS "C--Syl_Mid_Vowel" {*|a/C/*,*|e/C/*,*|o/C/*}

QS "C--Syl_Unrounded" {*|a/C/*,*|a/C/*,*|e/C/*,*|i/C/*,*|one/C/*}

QS "C--Syl_Central_Vowel" {*|a/C/*,*|one/C/*}

QS "C--Syl_Back_Vowel" {*|o/C/*,*|u/C/*}

QS "C--Syl_Rounded" {*|o/C/*,*|u/C/*}

Appendix D: Sample Speech Label

For the sentence

ሴቲቱ ግን በታማኝነት አልፀናችም ： ：

0.0000000 0.3752110 sil

0.3752110 0.5017615 sie

0.5017615 0.6593945 tii

0.6593945 0.7881651 tu

0.7881651 0.8791927 gix

0.8791927 0.9813211 nix

0.9813211 1.0834495 be

1.0834495 1.2011193 ta

1.2011193 1.3609725 ma

1.3609725 1.5119450 nxix

1.5119450 1.6495963 ne

1.6495963 1.7717064 tix

1.7717064 1.8560734 a

1.8560734 1.9471009 lix

1.9471009 2.1025138 xxe

2.1025138 2.3489541 na

2.3489541 2.5310092 cix

2.5310092 2.6864220 mix

Appendix E: Sample Amharic Lexicon

```
(defvar amhariclexdir (path-append lexdir "amharic"))
(define (amharic_lts_function word feats)
  "(amh_lts_function word feats)
Function called for amharic when word is not found in lexicon.  Uses
LTS rules trained from the original lexicon, and lexical stress
prediction rules."
  (require 'lts)
  (if (not (boundp 'amharic_lts_rules))
      (load (path-append amlexdir "amharic_lts_rules.scm")))
  (let ((dcword (downcase word))
          (syls) (phones))
(define (aau_amharic_addenda)
"(aau_amharic_addenda)
Add a whole host of various entries to the current lexicon with
amharic phones."
  (lex.add.entry
   '("ወር" nil (((we) 0)) ((rix) 0)))
  (lex.add.entry
   '("ጌታ" nil (((gie) 0) ((ta) 0)))
  (lex.add.entry
   '("ተመለሰች" nil nil (((te) 0) ((me) 0) ((le) 0) ((se) 0) ((cix) 0)))
  (lex.add.entry
   '("ወደ" nil (((ei) 1) ((s ii) 1))))
  (lex.add.entry
   '("በለዓምን" nil (((be) 0) ((le) 0) ((a) 0) ((mix) 0) ((nix) 0))))
  (lex.add.entry
   '("አባትዋ" nil nil (((a) 0) ((ba) 0) ((tix) 0) ((wa) 0)))
  (lex.add.entry
   '("ሆይ" nil (((ho) 0) ((yix) 0)))
```

(lex.add.entry
'("የኔ" nil (((ye) 0) ((nie) 0))))
(lex.add.entry
'("ጎሳ" nil (((go) 0) ((sa) 0)))
(lex.add.entry
'("ከምናሴ" nil (((ke) 0) ((mix) 0) ((na) 0) ((sie) 0)))
(lex.add.entry
'("ነው" nil (((ne) 0) ((wix) 0)))
(lex.add.entry
'("እጅግ" nil (((e jix) 0) ((gix) 0)))
(lex.add.entry
'("ልሰማው" nil (((lix) 0) ((se) 0) ((ma) 0) ((wix) 0)))
(lex.add.entry
'("ደካማ" nil (((de) 0) ((ka) 0) ((ma) 0))))
(lex.add.entry
'("ቀይ" nil (((qe) 0) ((yix) 0)))
(lex.add.entry
'("ባህር" nil (((ba) 0) ((hix) 0) ((rix) 0)))
(lex.add.entry
'("ቃዴስ" nil (((qa) 0) ((die) 0) ((six) 0)))
(lex.add.entry
'("ለራሷ" nil (((le) 0) ((ra) 0) ((sua) 0))))
(lex.add.entry
'("ሄደ" nil (((hie) 0) ((de)) 0)))
(lex.add.entry
'("ከዚያም" nil (((ke) 0) ((zii) 0) ((ya) 0) ((mix) 0)))
(lex.add.entry
'("እሷ" nil (((ix) 0) (sua) 0)))
(lex.add.entry

'("ግን" nil (((gix) 0) ((nix) 0)))

(lex.add.entry

'("እንዲህ" nil (((ix) 0) ((nix) 0) ((dii) 0) ((hix) 0)))

(lex.add.entry

'("አለች" nil (((a) 0) ((le) 0) ((cix) 0)))

(lex.add.entry

'("እኔ" nil (((ix nie) 0)))

.....

Appendix F: The Configuration of BLSTM Based Acoustic and Duration Models

BLSTM Based Acoustic Model Configuration

Merlins: /home/mah/merlin

TOPLEVELs: /home/mah/merlin/egs/build_your_own_voice/s1

work: %(TOPLEVELs)s/experiments/amharic_voice/acoustic_model

data: %(work)s/data

inter_data: %(work)s/inter_module

file_id_lists: %(data)s/file_id_list_full.scp

test_id_lists: %(data) s/test_id_list.scp

in_mgc_dirs: %(data)s/mgc

in_bap_dirs: %(data)s/bap

in_lf0_dirs: %(data)s/lf0

sptk: %(Merlin)s/tools/bin/SPTK-3.9

world: %(Merlin)s/tools/bin/WORLD

[Labels]

enforce_silence: False

silence_pattern: ['*-sil+*']

label_type: state_align

label_align:

(TOPLEVELs)s/experiments/amharic_voice/acoustic_model/data/label_state_align

question_file_name: % (Merlin)s/misc/questions/questions-aau.hed

add_frame_features: True

subphone_feats: full

[Outputs]

mgc: 60

dmgc: 180

bap: 1

dbap: 3

lf0: 1

dlf0: 3

[Waveform]
test_synth_dir: None
vocoder_type: WORLD
samplerate: 16000
framelength: 1024
fw_alpha: 0.58
minimum_phase_order: 511
use_cep_ap: True
[Architecture]
switch_to_tensorflow: False
switch_to_keras: False
hidden_layer_size : [512, 512, 512,512]
hidden_layer_type : ['BLSTM', 'TANH', 'TANH', 'TANH']
model_file_name: BLSTM
sequential_training : True
dropout_rate : 0.0
batch_size   : 256
lr_decay   : -1
learning_rate : 0.002
optimizer : sgd
warmup_epoch   : 10
training_epochs : 25
output_features   : ['mgc', 'lf0', 'vuv', 'bap']
gen_wav_features   : ['mgc', 'lf0', 'bap']
[Data]
train_file_number: 540
valid_file_number: 30
test_file_number: 30
buffer_size: 200000
[Processes]
AcousticModel : True

GenTestList : False

NORMLAB : True

MAKECMP : True

NORMCMP : True

TRAINBLSTM : True

BLSTMGEN : True

GENWAV : True

CALMCD : True

BLSTM Based Duration Model Configuration

Merlin: /home/mah/merlin

TOPLEVELs: /home/mah/merlin/egs/build_your_own_voice/s1

work: %(TOPLEVELs)s/experiments/amharic_voice/duration_model

data: %(work)s/data

inter_datas: %(work)s/inter_module

file_id_lists: %(data)s/file_id_list_demo.scp

test_id_lists: %(data)s/test_id_list.scp

in_dur_dirs: %(data)s/dur

[Labels]

silence_pattern: ['*-sil+*']

label_type: state_align

label_align:

%(TOPLEVEL)s/experiments/amharic_voice/duration_model/data/label_state_align

question_file_name: %(Merlin)s/misc/questions/questions-aau.hed

add_frame_features: False

subphone_feats: none

dur: 5

test_synth_dir: None

[Architecture]

switch_to_tensorflow: False

switch_to_keras: False

hidden_layer_size : [512, 512, 512,512]

hidden_layer_type : ['BLSTM', 'TANH', 'TANH', 'TANH']

model_file_name: BLSTM_neural_network

sequential_training : True

dropout_rate : 0.0

batch_size : 64

lr_decay : -1

learning_rate : 0.002

optimizer : sgd

warmup_epoch : 10

training_epochs : 25

output_features: ['dur']

train_file_number: 540

valid_file_number: 30

test_file_number: 30

buffer_size: 200000

DurationModel : True

GenTestList : False

sub-processes

NORMLAB : True

MAKEDUR : True

MAKECMP : True

NORMCMP : True

TRAINBLSTM : True

BLSTM-RNNGEN : True

CALMCD : True

Declaration

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

Declared by:

Name: Mahlet Awel Temam

Signature: _____

Date: _____

Confirmed by advisor:

Name: Yaregal Assabie (PhD)

Signature: _____

Date: _____