ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

SYLLABLE-BASED AMHARIC SPEECH SYNTHESIS (TTS) USING HMM

BAHIRU DEMESSIE DUBIE

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Information Science

June, 2017
Addis Ababa

Declaration

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

This thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A list of references is appended.

Signature:-_____

Bahiru Demessie

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

Advisor's Signature:-_____

Solomon Teferra (PhD)

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

SYLLABLE-BASED AMHARIC SPEECH SYNTHESIS (TTS) USING HMM

Bahiru Demessie Dubie

Approved By

Examining Board

1. **Solomon Teferra (PhD),** Advisor _____

2. **Million M. (PhD),** Examiner _____

3. **Wondwossen M. (PhD),** Examiner _____

Dedication

To my family

Acknowledgements

First of all, thanksgiving, praise and glory is all to almighty God, who gives me grace, love, patience, healthy, wisdom and ability to walk through all the problems and obstacles during the period of my life and all the courage to finalize this work.

Next, I would like to express the deepest appreciation to my advisor, Dr. Solomon for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. This thesis would never have been possible without his insightful observations and advice. His generousness for help others manifest via his ASR corpus avail publicly. I am one of the beneficiaries from his aptitude.

Thank you for all instructors who gives me knowledge during my course work specially Dr. Million, Dr. Solomon and Dr. Dereje, their deep scientific and technical competence push me to contest this challenging topic.

I would sincerely like to thank my friend Eshetu Biru for his encouragement. All the respondents who participated in filling out the evaluation questionnaire also deserve gratitude.

Finally I would like to express love, thanks, appreciation, and respect to my wife Amarech Hailu and our kids for their caring and praying throughout my study.

Table of Content

Contents **Page**

List of Tables..... viii

List of Figures..... ix

List of Appendices..... x

List of Acronyms..... xi

Abstract..... xii

CHAPTER ONE

1. INTRODUCTION---------- 1

1.1. **Background---------- 1**

1.2. **Motivation---------- 3**

1.3. **Statement of the Problem---------- 5**

1.4. **Research Questions---------- 6**

1.5. **Objectives of the Study---------- 7**

1.5.1. General Objective---------- ---------- 7

1.5.2. Specific Objectives---------- 7

1.6. **Significance or Benefits of the Study---------- 7**

1.7. **Scope and Limitation of the Study---------- 8**

1.8. **Research Methodology ---------- 8**

1.8.1. Critical Review of related Literature---------- 8

1.8.2. Data Selection and Preparation ---------- 8

1.8.3. Implementation and Tools ---------- 10

1.8.4. Testing Procedure---------- 10

1.9. **Organization of the thesis---------- 11**

CHAPTER TWO

2. LITERATURE REVIEW---------- 12

2.1. **Overview---------- -------- 12**

2.2. **Human Speech Production System**---------- **13**
2.3. **Speech Synthesis**---------- **18**
2.4. **Speech Synthesis Methods**---------- **19**
2.4.1. Rule-based---------- **20**
2.4.2. Corpus-based---------- **21**
2.5. **HMM-based Speech Synthesis**---------- **26**
2.5.1. The Hidden Markov Model---------- **26**
2.5.2. Speech Signal Parameter Modeling---------- **28**
2.5.3. Decision Tree Building for Context Clustering---------- **30**
2.5.4. Speech Parameter Generation---------- **31**
2.5.5. The HMM-based Speech Synthesis System---------- **31**
2.6. **Applications of Speech Synthesis Systems**---------- **33**
2.7. **Syllabification and Syllable structure**---------- **34**
2.7.1. Syllabification---------- **34**
2.7.2. Syllabification model for Amharic---------- **36**
2.8. **Related Works**---------- **40**

CHAPTER THREE

3. AMHARIC LANGUAGE, WRITING SYSTEM AND ITS PHONETICS------43
3.1. **Amharic language and its Orthography**---------- **43**
3.2. **Transcription System**---------- **46**
3.3. **Amharic Phonetics**---------- **48**
3.3.1. Vowels---------- **48**
3.3.2. Consonants---------- **49**
3.4. **Syllable structure of Amharic words**---------- **52**
3.5. **Stress and Syllables**---------- **54**

CHAPTER FOUR

4. DESIGNING SPEECH SYNTHESIS SYSTEM---------- **55**
4.1. **The overall system architecture**---------- **55**
4.2. **Description of the Architecture**---------- **57**

4.2.1. Data Collection and Preparation---------- 57
4.2.2. Normalization---------- 57
4.2.3. Transcription and Structuring---------- **58**
4.2.4. Labels and utterances---------- **58**
4.2.5. Feature Extraction---------- **63**
4.2.6. Training the Model---------- **64**
4.2.7. Synthesis Phase---------- -------- **67**
4.3. **Software Packages and implementation**---------- **69**
4.3.1. Software package listing---------- **69**
4.3.2. Software package description---------- **71**
4.4. **Experiment Setup and implementation**---------- **73**
4.4.1. Initial stages---------- **73**
4.4.2. Festvox---------- **73**
4.5. **HTS-demo**---------- **77**

CHAPTER FIVE

5. THE EXPERIMENTAL RESULTS AND EVALUATION---------- **79**
5.1. **Testing Procedure**---------- **79**
5.2. **Evaluation Criteria**---------- **80**
5.3. **Evaluation Results and Analysis**---------- **81**

CHAPTER SIX

6. CONCLUSION AND FUTURE WORK---------- **85**
6.1. **Conclusion**---------- **85**
6.2. **Recommendation**---------- **87**

REFERENCES---------- -------- **89**

List of Tables

Table 3.1: Distribution of Amharic character set---------- **44**

Table 3.2: Sample Amharic core characters---------- **45**

Table 3.3: The numeral symbols of Amharic----------**46**

Table 3.4 The Amharic consonants with the vowels (using ASCII translation) ---------- **47**

Table 3.5: Phonetic representation and characterization of Amharic consonants---------- **50**

Table 3.6: Different kinds of Amharic Syllable templates----------**53**

Table 4.1: question set format---------- **61**

Table 5.1: Performance Measure of Amharic prototype---------- **81**

Table 5.2: Scales used in MOS---------- **82**

Table 5.3: Average MOS Scores of phone based Amharic speech demo---------- **83**

Table 5.4: Average MOS Scores of syllable based Amharic speech demo---------- **83**

List of Figures

Figure 2.1: Human speech production system---------- **15**

Figure 2.2: Speech production process and the mode---------- **17**

Figure 2.3: Block diagram of a text-to-speech system---------- **19**

Figure 2.4: Basic principle of a concatenative unit selection speech synthesis system------ **22**

Figure 2.5: Example of a left to right HMM structure---------- **27**

Figure 2.6: MFCC coefficients computation---------- **29**

Figure 2.7: Decision tree context clustering in HMM-based speech synthesis---------- **30**

Figure 2.8: HMM speech Synthesis system ---------- **32**

Figure 2.9: Syllable structure σ –syllable---------- **35**

Figure 2.10: General automatic syllabification model for Amharic text---------- **36**

Figure 3.1: IPA maps of the Amharic vowels ---------- **49**

Figure 4.1: Overall system architecture---------- **56**

Figure 4.2: Overview of Synthesis Process---------- **68**

List of Appendixes

Appendix A: List of all Amharic core characters----------**94**

Appendix B: Amharic characters ASCII transliteration---------- **95**

Appendix C: ASCII Translation python code----------**96**

Appendix D: part of left of left of phoneme question set example----------**97**

Appendix E: part of phone set definition---------- **98**

Appendix F: sample prompts snap shoot---------- **100**

Appendix G: Configuration of HTS system----------**101**

Appendix H: Evaluation format---------- **104**

Appendix I: Evaluation Questionnaire----------**105**

Appendix k: frequency of syllables and phones within selected dataset----------**106**

List of Acronyms

ASR	Automatic Speech Recognition
CART	Classification and Regression Trees
CV	Consonant Vowel
DCT	Discrete Cosine Transform
DSP	Digital Signal Processing
F0	Fundamental frequency or Pitch
GV	Global Variance
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
HTS	(H Triple S) HMM-Based Text-to-Speech Synthesis
IPA	International Phonetic Alphabet
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MLSA	Mel Log Spectrum Approximation
MOP	Maximal Onset Principle
MOS	Mean Opinion Score
MSD-HMM	Multi Space Distribution Hidden Markov Model
NLP	Natural Language Processing
NSWs	Non-Standard Words
POS	Part Of Speech
SAMPA	Speech Assessment Methods Phonetic Alphabet
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum
TTS	Text-To-Speech
VODER	Voice Operating Demonstrator

Abstract

Speech Synthesis systems have been developed gradually over the last few decades and it has been integrated into several new applications. There is still much work and improvements to be done in prosodic, text preprocessing, and pronunciation fields to achieve more natural sounding speech.

In this thesis work, ASR corpus is used to develop a syllable based speech synthesis system for Amharic language using Hidden Markov Model. The datasets are randomly selected from ASR corpus with six female speakers' corpora as training data. Both text and speech with the size of each 600 were used. These corpus were split in to two, 90% for training and 10% testing data sets. Components of Hidden Markov Model and Amharic language features are studied. Though, every feature of the Amharic language was not considered since it needs a lot of time and deep linguistic knowledge.

The utterance structure generated by festival and festvox together with the parameters extracted from the raw wave data were used for training the model. Formerly, the speech parameter sequence, which is generated based on the predicted models, is used to synthesis the speech waveform by a vocoder. In this research work the text that is going to be synthesized was assumed to be transcribed. Lastly, the synthesized speech is generated from the trained model based on the labeled input text.

Evaluation is done in two ways. First, based on the researcher evaluation, the systems register on the overall performance 75.56% for syllable based and 77.78% for phone based system; Preference evaluation result shows that Syllable based synthesis performs better in naturalness than intelligibility while Phone based TTS performs better in intelligibility with 550 sentences' training data. Second, the average MOS evaluation of the system from eight listeners for the five Amharic sentences is found to be 2.94 and 3.02 for phone based and syllable based, respectively. It shows that, Syllable based TTS system outperforms the system that uses phone as basic unit.

According to the MOS results, the synthesis system is categorized as good in terms of both intelligibility and naturalness. The result looks encouraging and further improvement depends on proper works in different context such as phoneme coverage, lexicon, and question set.

Keywords: *ASR corpus, Syllable, Speech Synthesis, Hidden Markov Model.*

CHAPTER ONE

Introduction

1.1 Background

Speech synthesis is a process which artificially produces synthetic speech from the text input for various applications. Converting text to speech encompasses both natural language processing and digital signal processing [1].

Speech synthesis has emerged as an important technology in the context of human computer interaction. Although an intensively studied domain, its language dependency makes it less accessible for most of the languages. If for English, French, Spanish or German for example, the variety of choices starts from open-source user-configurable systems to high-quality proprietary commercial systems, this is not the case for Amharic. The lack of extended freely available resources makes it hard for the researchers to develop complete speech synthesis systems and design new language-dependent enhancements for the language.

Amharic is the official working language of government of Ethiopia, among 73 languages which are registered in the country [2]. Amharic is second largest spoken Semitic language in the world next to Arabic[3].The majority of the speakers of Amharic are found in Ethiopia, but there are also speakers in a number of other countries, particularly Israel, Eritrea, Canada, the USA and Sweden. It has five dialectical variations across different parts of the country [4]. These include Addis Ababa, Gojam, Gonder, Wollo and Menz.

Unlike other Semitic languages, such as Arabic and Hebrew, Amharic /amarnxa/ script uses a grapheme based writing system called fidel /fidel/ which is written and read from left to right [5]. Modern Amharic has inherited its writing system from Ge'ez /gixz/, which is still the classical and ecclesiastical language of Ethiopia [6]. An Amharic character represents a consonant vowel

(CV) sequence and the basic shape of each character is determined by the consonant, which is modified for the vowel. Amharic character symbols are categorized into four different categories consisting 276 distinct symbols; these are core character, labiovelar, labialized and labiodentals [8].

Amharic has 39 phones 31 of which are consonants while 7 are vowels. Most of these Amharic phones are used by other languages while there are Amharic phones that are not found in any other foreign language [7]. These are ጵ/p'/, ጥ/t'/, ፅ/s'/, ጭ/ʧ'/, and ቅ/k'/ which have a sharp click–like characters beside glottalized voice articulating at different places.

Modern Speech synthesis systems are based on two main techniques[9]: Concatenative TTS and Statistical TTS. The concatenative approach needs to employ large databases of speech waveforms in order to synthesize an intelligible high quality speech from the text, and has a number of drawbacks: (a) inefficient time/space usage due to the required large database. (b) Inherent inability to synthesize speech with desired features because the concatenated waveforms are predefined in the database. (c) Poor performance when the speech segment doesn't appear in the database to be synthesized. The statistical approach is a more promising one as it enables synthesizing speech with desired features; it doesn't need large databases as concatenative method, and it is not database dependent. On the other hand the statistical approach often suffers from several drawbacks: Oversmoothed synthesized speech (which is partially solved by introducing Global Variance features), lack of naturalness, (which is partially solved by including temporal speech features into the training phase and inter frame dynamic parameters)[9].

Various attempts are made on speech synthesis for Amharic with a promising performance evaluation result using concatenation techniques. However, the types of synthesis approaches

employed by those researchers have difficulties when trying to improve the naturalness or intelligibility of the synthetic speech. Besides, as far as the researchers' knowledge, it is well understood that there are few local attempts made to develop speech synthesis for Amharic using the statistical parametric approach. Therefore, it needs further research on the use of this approach for the language. In any attempt to apply this approach for the development of Amharic speech synthesis, there are issues to be considered, such as language features and sub-word sound units. Although the phones are very small in number and relatively easy to train, they are much more sensitive to contextual influence than the large units. Other extremes are the syllables which are the longer sub-word units and the least context sensitive ones. A syllable usually consists of vowel surrounded by one or more consonants. Moreover, the syllable has a close connection to articulation, integrates some co-articulation phenomena. The problem with syllable is their large number for a number of languages, such as English, which is not the case in Amharic [10].

All the above benefits of both statistical approach (HMM model) and syllables are the driving force to conduct this research to experimentally demonstrate syllable based acoustic models for designing Amharic text-to-speech system using HMM, in which speech waveform is generated from Hidden Markov Models themselves, and applies it to Amharic speech synthesis using the general speech synthesis architecture of HTS.

1.2 Motivation

Finding/searching prepared and annotated TTS corpus is difficult. Lack of corpus is one of the problems facing many researchers. Because of its nature, it requires huge amount of time and cost in order to produce and prepare speech dataset (but it is possible) due to different factors. In this research we attempt to answer the following questions:

- How would it be possible to develop TTS using available speech data, without preparing corpus explicitly for a language?
- Can we use ASR corpus for TTS for under resourced languages like Amharic?

The Amharic language is syllabic in nature and Amharic syllable is also small in number. In most literature the larger unit (like syllable) is preferred for speech productions because of phones are easily influenced by context. So the utilization of this linguistic feature of Amharic was the idea behind this work.

Synthesis of the speech from a consistent speaker, which is phonetically rich and without any noise, can be done by many techniques. It is always a tough job to synthesis that uses imperfect speech data. Poor phonetic coverage, speaker inconsistency and noise from background are common imperfections in speech data. Such imperfections are associated with a good performance ASR speech corpus except the phonetic coverage. Collecting the ASR speech data and building a synthesis system from such data is a challenge. Therefore,

- What technique is suitable for this purpose?

  Statistical Parametric Synthesis system is built from ASR speech using HMM. This type of speech synthesis system has been successfully used in such imperfect speech with a limited amount of data for other language (like English) previously.

Hence, the motivation arose to do a research that explores intelligible and natural sounding synthetic voices in Low Resource Languages, without the expense of collecting and annotating specialized data specifically for text-to-speech. The researcher focuses on Amharic language, in order to demonstrate speech synthesis that syllable as a basic unit of the acoustic model using HMM.

1.3 Statement of the problem

To obtain various voice characteristics in text-to-speech systems based on the selection and concatenation of acoustic units, a large amount of speech data is necessary. However, it is difficult to collect, segment, and store it. From these points of view, in order to construct a speech synthesis system which can generate various voice characteristics, statistical (HMM-based text-to-speech synthesis) system would be a solution.

Concatenative speech synthesis (USS)[11] uses phonemes, diphones, syllables, words or sentences as a basic speech units. In statistical based techniques, phones (along with their context) are used as basic units. A major issue is to model the trajectories and reduce the over smoothing effect and also longer units such as syllables are not straight-forward to model when compared to phones.

Different attempts have been made to use syllables as a unit of recognition for the development of ASR [10]. One of them showed that the use of syllables in the development of ASR for Amharic seems promising, because Amharic has only 233 distinct CV syllables. ASR for Amharic using HMM by Solomon Teferra et al. [10], have analyzed the results of their experiments from the point of view of word recognition accuracy, recognition speed and memory requirements, and conclude that for Amharic modeling CV syllables, as represented by the orthographic symbols, is better alternative to the prevailing modeling unit of elementary sounds, like phones. Most important is that their research reflects the capabilities of HTK speech recognition toolkit with regards to use Amharic consonant-vowel (CV) syllables as units of speech recognition.

One of few attempts made on TTS for Amharic is that of Sebsibe H/Mariam et al. [12]. Their focus was describing the issues to be considered in developing a concatenative speech

synthesizer for Amharic language. The syllable structure of the language, the phonetic nature of the language, and the result of the perceptual test of the synthesizer has been discussed. The researchers recommended the need to work on improvement of the quality of speech synthesizer for Amharic. They suggested that, first, proper selection of unit, since the language is phonetic, syllable as a basic unit may outperform the phone as a basic unit. And second, optimal selection of corpus, which proportionally covers all basic units and variations, will give better quality.

Based on the reviews made so far and knowledge of the researcher, the sub-word units (syllables) are represented using HMMs as they are generative models, in this framework. Unlike ASR, where HMMs are used for classification, here HMMs are used to reconstruct speech by generating parameters, unfortunately this research uses ASR corpus. Bereket [13] has tried to show in his research on HMM based speech synthesis for Amharic that, phones are used as the basic units of synthesis in HTS. Our hypothesis in this research is that as that of the ASR, syllables may outperform the phone as a basic unit of speech synthesis in HTS.

1.4 Research questions

This research tries to answer and address the following research questions:-

- How effective are syllable based HMM models in generating TTS for Amharic language?
- How useable is an ASR corpus for the development of Speech synthesis in a language?
- What are the limitations and challenges for designing and modeling HMM based TTS using syllables as sub-word units and ASR corpus?

1.5 Objectives of the Study

1.5.1 General Objective

The main objective of this research is to experimentally demonstrating syllable based acoustic models for developing Amharic speech synthesis system using HMM and ASR corpus.

1.5.2 Specific Objectives

- To study orthographic and phonetic characteristics of Amharic language.
- To critically review literature on Synthesis Systems (TTS), primarily for the Amharic language.
- To select a text and speech corpus from existing Amharic ASR corpus that can support training of the HMM model.
- To design and model phone-based and syllable-based Amharic speech synthesis, using HMM.
- To evaluate the performance of TTS Systems in terms of their quality of speech.

1.6 Significance or Benefits of the Study

On one hand, the research shows that the possibility of using existing ASR corpus for developing speech synthesis system, for under resourced languages like Amharic. And the outcome of this research can help for the developer to know the best acoustic unit for modeling HMMs in HTS, for developing Amharic TTS system. It can also be the groundwork for the next researcher for designing and enhancing the selected unit (syllable) with full features and characteristics.

On the other hand, it also reflects the capabilities of HMM toolkit with regards to use Amharic consonant-vowel (CV) syllables as units of speech synthesis and ASR corpus.

1.7 Scope and limitation of the study

Syllable based approach to speech synthesis is an interesting alternative to the phone based approach, especially for the syllable-timed languages like Amharic. The main focus is to model and develop a speech synthesizer prototype (model) for Amharic texts which produce synthetic speech based on both appropriate speech units like syllables (CV) and statistical parametric techniques using ASR corpus. The stress, duration and intonation modeling and Amharic gemination either lexical or morphological are challenging problems of speech synthesis [14], these are beyond the scope of this work. In addition text processing for Amharic NSWs is beyond the scope of this thesis. But these are considered as future work, since those issues are related with syllable with variable length as speech unit.

1.8 Research Methodology

A research methodology defines what the activity of research is, how to proceed, how to measure progress, and what constitutes success.

1.8.1 Critical Review of Related Literature

Systematic Literatures review related with both Amharic language and statistical parametric speech synthesis had been conducted so as to understand both the nature of the language and how to design a new system respectively. Research publications and the Internet had been assessed. Various speech synthesis techniques are also briefly studied in order to identify their differences.

1.8.2 Data Selection and Preparation

The main objective of the data collection methodology is to prepare appropriate training and testing dataset for the desired system. Unless, the training dataset have to have proper

representation of phonemes and enough training data, observation sequence problem might be occurred, so needs high curiosity as possible.

Amharic has 276 distinct CV syllabic symbols. However, some of the symbols are duplicate in a sense that they represent the same syllabic sounds [15]. Redundant symbols that represent the same syllabic sounds have been taken as one and a total of 233 CV syllables and 39 Amharic phones have been considered as acoustic units in this study.

The researcher had planned the preparation of speech corpus in two ways. In the first plan, the researcher decided that previously developed statistical speech synthesis system and corpus would be surfed for new change of Amharic speech synthesis system using HMM approach, if there exists. This was because; the existing system and corpus were already tested and verified by researchers and avoids reinventing the wheel. Unfortunately, the researcher had got only ASR corpus for Amharic and tried to use it. And then, using ASR speech corpus and try whether this corpus works for TTS or not. If it works, it might also be useful for future work (such as, speaker adaptive TTS, unified ASR and TTS model and so on). So as, the training speech corpus were taken from ASR speech corpuses which were corresponding to text corpus. All are female speakers with different noisy environment and tones. And the second plan was the recording process took place in an office environment with minimal noise with one speaker. This approach is already tested and workable but also it needs more time for practicing, recording and breach speaker conditions. Praat is the software that was used to record the speech corpus. A regular microphone and a normal laptop made up the hardware equipment used to record the speech corpus. Finally the first plan was successfully done.

The data selection methods from a corpus of size 10,000+ ASR men and women speakers' speeches and corresponding sentences are used for selecting the training and testing dataset using

random sampling technique. The syllabic and phonetic coverage of selected dataset are described in chapter four.

1.8.3 Implementation and Tools

The toolkit that is used for HMM based speech synthesis system is called HTS. The HMM-based Speech Synthesis System (H Triple S - HTS) is a collection of open source tools dedicated to the development of text-to-speech systems using Markov models. The content of these tools refer strictly to the modeling, training and speech generation without text processing. HTS requires platform such as, operating system and a PC or laptop with high processing speed during training the model. Moreover, HTS require HTK to be in place. Hence, it is also installed on the machine. In addition, many tools for different purpose such as Festival, wavesurfer and python programing language are used to prepare, analyze, and develop the corpus and required system modules. To end with, the designing, modeling, integrating and testing tasks will be done and using EndNote to generate references as IEEE format for the report document.

1.8.4 Testing Procedure

The performance of the Amharic synthesizer prototype is evaluated using representative test datasets. The workability of the system is examined for the given labeled text. In addition, the overall quality (that is, intelligibility and naturalness) of synthetic speech created by the system is also measured by the user.

On the other hand, to test the performance of TTS system a number of testing techniques are suggested in literatures [16][17], Modified Rhyme Test (MRT), and Diagnostic Rhyme Test (DRT), Mean Opinion Score (MOS), Pair Comparison (PC) and Semantically Unpredictable Sentences (SUS) are some of the performance testing methods. In this thesis work, Mean Opinion Score (MOS) is used, which is the most widely and simplest method to evaluate speech

quality. It is also suitable for overall evaluation (intelligibility and naturalness) of synthetic speech. MOS is a five level scale from bad (1) to excellent (5) and it is also known as ACR (Absolute Category Rating) [17]. The evaluators give their option based on MOS scale for both intelligibility and naturalness criteria of synthesized speech.

1.9 Organization of the Thesis

This thesis is organized as follows: Chapter 1 presents an introduction to the subject matter under study, motivation, description of the statement of the problem along with justifications, objectives of the research and methodology. Chapter 2 presents literature review on human speech production system, speech synthesis and its techniques, syllable structure and syllabification especially for Amharic language briefly discussed. In this chapter related works on speech synthesis system on local language using different techniques done by different researchers had been reviewed. In Chapter 3, present about Amharic language, its orthography, phonetic feature are discussed. Design of speech synthesis model for Amharic is presented in Chapter 4. The experimental results and evaluation are discussed in Chapter 5. In Chapter 6, conclusion and future work are pointed out.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

This chapter gives us background about this thesis work based on and it also includes review of related works. The use of machines ranging from manual to automatic, mechanical to electronic, handheld to room-filling have assisted people to exercise problem solving effectively and efficiently. The next generation of machines that human beings are so ambitious about is intelligent machines. An important function of these machines is that they master speech input and output to communicate with human beings naturally using natural languages. They understand human language and generate human like language [48]. Speech technology as one part of natural language processing (NLP) is the field that aims at the development of technologies that allow human beings to use natural languages in order to communicate with computers and other devices. Some of the most important applications of NLP technology are speech recognition (converting speech into text), text-to-speech synthesis (converting text to speech), and speech/text understanding (maps words into action and plans system-initiated actions) [48].

As it is already mentioned in the first chapter, speech synthesis, also called text-to-speech synthesis, is the artificial production of human speech from text. The process of converting written text into speech passes through a number of steps, which are broadly classified into two basic modules: the Natural Language Processing (NLP) module (text processing) and the Digital Signal Processing (DSP) module (speech synthesis). The NLP module is responsible for producing a phonetic transcription of the text together with the desired intonation and rhythm.

The DSP module transforms the symbolic information it receives from the NLP module into speech [10].

All languages have syllables with onsets; many languages require all syllables to have onsets in surface representation; no language requires all syllables to have codas. Each syllable has a nucleus, and language-particular conditions govern the class of possible onsets and codas[18].

Languages differ considerably in the syllable structures that they permit. For most languages, syllabification can be achieved by writing a set of declarative grammatical rules which explain the location of syllable boundaries of words step-by-step. It needs to have knowhow about the structure and the properties of Amharic syllables as of related studies.

Prior to taking a brief look at the text-to-speech synthesis technique, it is necessary to have an insight about how the human speech production system works. This helps us understand the whole text-to-speech synthesis procedure well. Section 2.2 describes the human speech production system briefly. Sections 2.3 and 2.4 describe speech synthesis and its techniques, respectively. 2.5 Application. 2.6 syllable structure and syllabification and finally review of related works are presented.

2.2 Human Speech Production System

Speech is produced by air-pressure waves emanating from the mouth and the nostrils of a speaker. The gross components of the speech production apparatus are the lungs, trachea, larynx (organ of voice production), pharyngeal cavity (throat), oral and nasal cavity. The pharyngeal and oral cavities are typically referred to as the vocal tract, and the nasal cavity as the nasal tract[19]. **Figure 2.1** shows the important articulatory motions and models of the human speech production system.

Rong-Wei[20] , quoting Huang [19], and mentioned that the lung is the source of the air during speech production. If the speech sound made the vocal folds (vocal cords) close together and oscillates against one another, the sound is said to be voiced. When the folds are too slack or tense to vibrate periodically, the sound is said to be unvoiced. The larynx is the structure that holds and manipulates the vocal cords. The "Adam' s apple" in males is the bump formed by the front part of the larynx. The place where the vocal folds come together called the glottis. The epiglottis is the fold of tissue below the root of the tongue. The epiglottis helps to cover the larynx during swallowing; food goes into the stomach and not the lungs. A few languages use the epiglottis in making sounds. For instance, a short distance behind the upper teeth is a change in the angle of the roof of the mouth. This is the alveolar ridge. Sounds, which involve the area between the upper teeth and this ridge called alveolar (i.e. a consonant articulated with the tip of the tongue near the gum ridge). The hard portion of the roof of the mouth called hard palate that the term "palate" by itself usually refers to the hard palate and the Velum (Soft Palate) operates as a valve and it is the soft portion of the roof of the mouth, lying behind the hard palate called soft palate that separates the oral and nasal cavities. The velum can also move: if it lowers, it creates an opening that allows air to flow out through the nose; if it stays raised, the opening is blocked, and no air can flow through the nose.

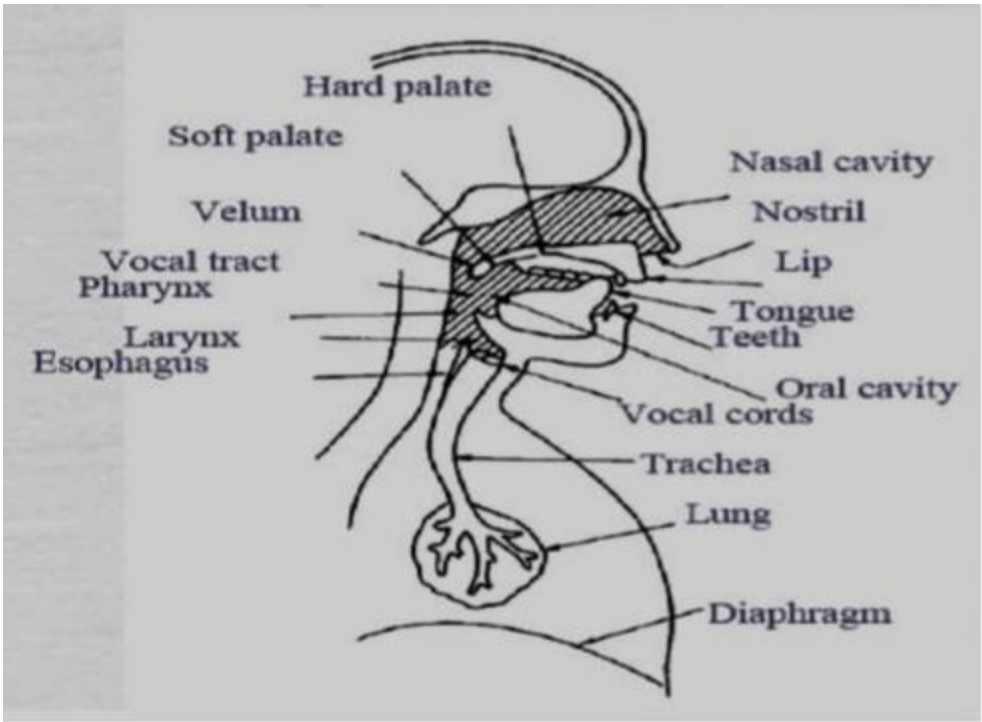

Figure 2.1: Human speech production system [19].

The tongue is among the finer anatomical features critical to speech production. The dorsum is the main part of the tongue, lying below the hard and soft palate. It is found at the back part of the tongue (hence "dorsum", or "back" for Latin) and shaped away from the palate for vowels, placed close to or on the palate or other hard surfaces for consonant articulation. The teeth are also another place of articulation used to brace the tongue consonants[21]. For example, plosives are considered as the most basic type of consonant, which are produced by stopping the flow of air at some point and suddenly releasing it. They form a complete obstruction to the flow of air out of the mouth and nose, and normally these results in a build-up of compressed air inside the chamber formed by the closure. When the closure is released, there is a small explosion that causes a sharp noise. The basic plosive consonant type can be exploited in many different ways: plosives may have any place of articulation, may be voiced or voiceless and may

have an egressive or ingressive airflow. The airflow may be from the lungs (pulmonic), from the larynx (glottalic) or generated in the mouth (velaric).

As mentioned above, the vocal cords play a great role in generating the kind of sound to be produced. The vowel are grouped in voiced sounds and the consonants are unvoiced sounds. Consonants involve constrictions, or gestures that narrow the vocal tract at a particular point. When we classify consonants, one of the most important things to consider is the place where this obstruction is made; this is known as the place of articulation, and in conventional phonetic classification, each place of articulation has an adjective that can be applied to a consonant. For example in Amharic language, there are five place of articulation as: labial, dental, palatal, velar, and glottal [14].

The other important thing that we need to know about the human speech production system is what sort of obstruction it makes to the flow of air. A vowel makes very little obstruction, while a plosive consonant makes a total obstruction. The type of obstruction the phonemes make is known as the manner of articulation. In Amharic, the common manners of articulations are stops, fricative, nasals, liquids and affricatives[22]. In general the human speech production, the model and parameter control (see **Figure 2.2**), which are employed to generate speech waveforms.

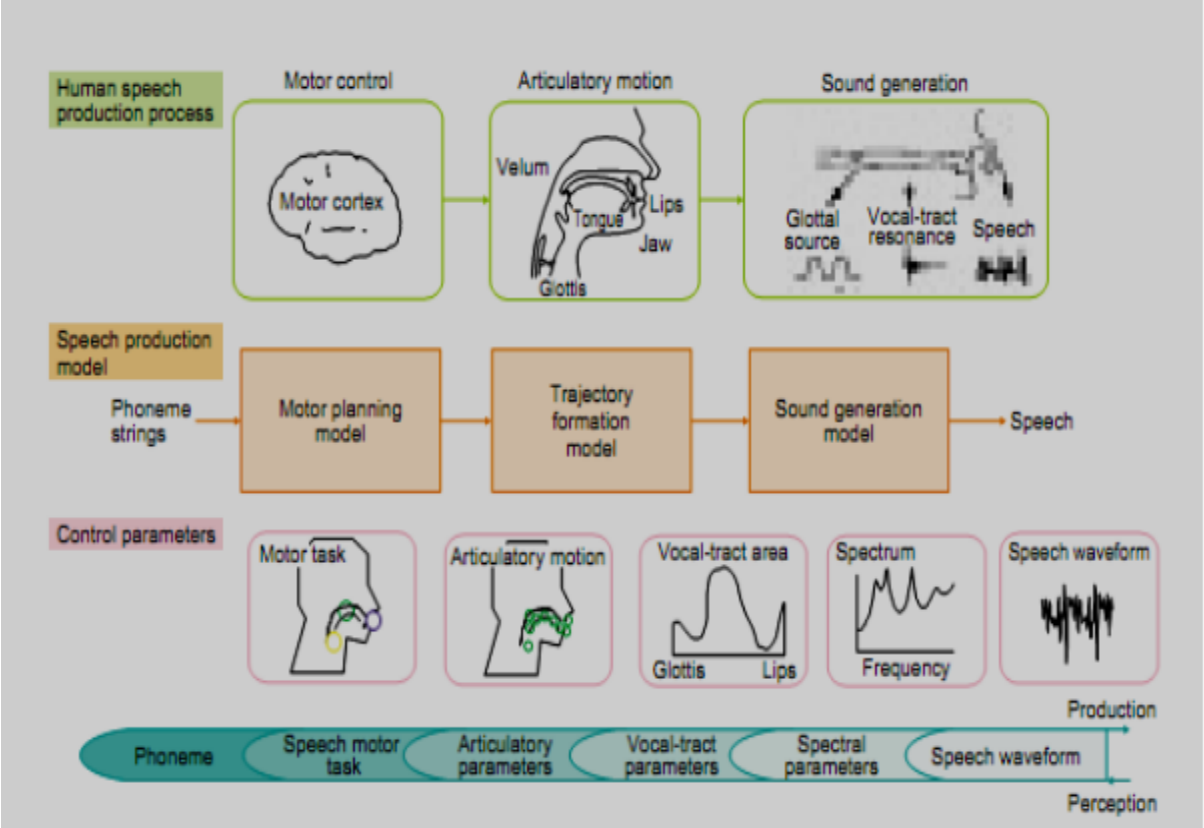

Figure 2.2: Speech production process and the mode [23]

Basically the human speech production process targeted to produce speech waveforms from different articulatory and control parameters, and acoustic units as shown in **Figure 2.2** above.

Tesfaye [24], quoting Elker discuses that the vocal tract is bound by hard and soft tissue structure. The structures are either essentially immobile or mobile. The mobile structures associated with high speech production are also referred to as articulators (i.e. jaw, tounge, lips and mouth). Movement of these articulators appeared to account for most of variations in vocal tract shape associated with speaking style. Modeling and controlling the segmental co-articulation and other phonetic factors is an important part of a TTS system.

2.3 Speech Synthesis

Speech synthesis is a method for deriving human-like speech from a text input. **Figure 2.3** shows the basic blocks of a TTS system. The process can be more easily understood if a parallel to learning a new language is made. Given a text in a new language, the first step is to determine the text segments which have to be preprocessed for a correct reading, such as numbers, abbreviations, buzzwords, and so on. Then, each letter has to be transposed in an acoustic correspondent or phoneme. Individual correspondents are not enough, as context influences the sound of a given letter. Having the correct succession of phonemes can then be concatenated into syllables, words, phrases and so on. Simple phrasing, duration and basic intonation are then assigned. And, at last the physical process of speech production, through mechanical articulation of the sounds takes place. If the person is a more advanced speaker of that language, emphasis and prosody are more likely to be reproduced correctly, similar to a native speaker.

In the same way, text-to-speech systems evolved from the simple reproduction of individual sounds through wooden tubes, to state-of-the-art speech synthesizers which use advanced semantic analysis and can output high-quality expressive speech. The entire system is usually broken down into two major components: text processing and speech synthesis. Each of them implies extensive analysis and synthesis methods with their correspondent arising problems.

The goals of a TTS system according to [25] are to clearly get the message across to the listener in terms of intelligibility and naturalness, and to be able to synthesize any given input text. This means that the text processor has to be able to transform any input text into a sequence of labels, and that the speech synthesizer has the means of outputting qualitative speech from any sequence of input labels.

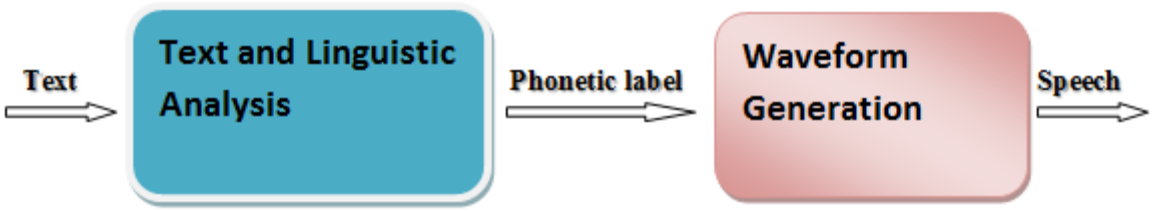

Figure 2.3: Block diagram of a text-to-speech system

The need for a text-to-speech system can be emphasized through its applications. The initial purpose of TTS was for visually impaired people to have access to written texts without the help of the Braille alphabet. With the appearance of analogue and digital storing devices, the speech synthesizers were used in even more applications. By concatenating pre-recorded speech segments, the system could output a limited number of combinations between the samples. This type of synthesizer is still used in client information applications, such as automated answering and GPS machines or ATMs. More advanced TTS systems are used in intelligent dialogue applications or in combination with automatic speech recognition, even translating applications from one language to another.

The following sections give an overview of the standard synthesis methods and the use of prosody within the text-to-speech systems.

2.4 Speech Synthesis Methods

Speech production is a complex process which involves a large number of computational resources and memory. Aside from the even more complex task of caring out a dialogue, even the reading aloud of a text implies training and processing on behalf of a person. Over the years multiple methods of speech synthesis have been proposed. Fortunately, nowadays, speech

synthesizers have evolved to a point where their intelligibility and naturalness is comparable to human speakers. Based on the main method of generating the speech signal, speech synthesizers can be classified into rule-based and corpus-based.

2.4.1 Rule-based

In rule-based methods no pre- recorded speech samples are used, each sound is defined by a fixed set of parameters.

- **Formant Synthesis**

Formant synthesis determines a set of rules on how to modify pitch, formants frequencies and other parameters from one sound to the other. It is based on the source-filter model of speech production. In formant synthesis, the formant resonances are represented by a number of filters having as input a train of impulses for the voiced segments and white noise for the unvoiced segments.

The most representative model of formant synthesis is the one described by[27], which later evolved into the commercial system of MITalk. There are around 40 parameters which describe the formants and their respective bandwidths, and also a series of frequencies for nasals or glottal resonators. A parallel structure of second order FIR filters is implemented for the fricatives and stops, and a cascade structure for the voiced sounds[28].

The problem with the formant synthesis is that the source-filter model itself has the drawback of not including the reaction of the filter unto the source. Another drawback is the fact that the acoustic realization of a sound varies over time, and cannot be represented by a fixed set of parameters. The same speaker asked to repeat the same word multiple times, will use different duration and intonations. Therefore, the formant synthesis lacks the modeling of the minute variations that make a long duration speech sample natural.

- **Articulatory Synthesis**

Articulatory synthesis has the potential of becoming one of the best synthesis methods. It uses mechanical and acoustic models of speech production to synthesize speech[30]. The physiological effects are modeled, such as the movement of the tongue, lips, jaw, and the dynamics of the vocal tract and glottis. Biomechanical, aerodynamic and acoustic studies are also involved. For example[31] uses lip opening, glottal area, opening of nasal cavities, constriction of tongue, and rate between expansion and contraction of the vocal tract along with the first 4 formant frequencies. This is a method which articulatory synthesis with the formant-based one, thus trying to alleviate the drawbacks of the later.

Unfortunately, the model is very complex and there is still a lack of analysis methods of the processes involved. A complete articulatory model would also include the electric impulses of the nerves and muscle movement of the entire phonatory apparatus. The use of magnetic resonance imaging has offered some more elaborate models of muscle movement and thus the results of the articulatory synthesis have improved.

However, the results of this type of synthesis are still far from being considered natural because of the use of partially heuristic determined rules and the fact that the acoustic processes vary from speaker to speaker. The physical characteristics of a person, such as length of vocal tract and tongue size influence the mechanical movement in speech production. Accurate physiological understanding of speech production is also lacking and thus the parameters of the model cannot be fully determined.

2.4.2 Corpus-based

Corpus-based methods use either the speech samples or segments of them, or derive their parameters from the direct analysis of the speech corpus.

- **Concatenative Synthesis**

Concatenative synthesis is the most commonly used method in commercial systems. It became a popular choice once the storage and computational characteristics of the digital devices became more and more advanced. The basic idea is the use of pre-recorded speech samples of fixed or variable lengths, which can fully capture the fine details of speech. This aspect was not possible in the rule-based methods.

In this type of method, an utterance is synthesized by concatenating several natural segments of speech (as shown in **Figure 2.4**). The samples are stored in a database, indexed by the phonetic content along with prosodic markers, context or other additional information. Samples of speech can include utterances, words, syllables, diphones or phonemes. Based on the type of segment stored in the database, the concatenative synthesis is either **fixed inventory** - segments in the database have the same length, or **unit selection** - segments have variable length and the system makes a decision of the best match.

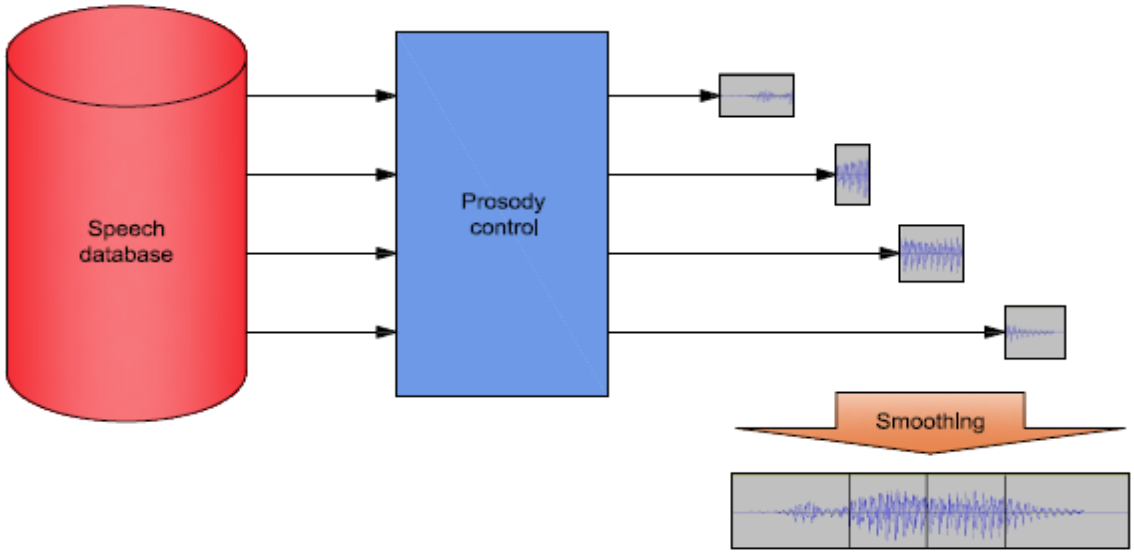

Figure 2.4: Basic principle of a concatenative unit selection speech synthesis system ([30])

The most common fixed inventory concatenative synthesis is the diphone concatenation[32]. A diphone in this case is defined from the middle of the first phoneme to the middle of the second one. Using this type of segmentation avoids the concatenation discontinuities at phoneme boundaries. For a simple diphone concatenation system, the database or speech corpus would include a single repetition of all the diphones in a language. Some more elaborate systems use diphones in different context (e.g. beginning, middle or end of word) and with different prosodic events (e.g. accent, variable durations etc.). Two major problems with this approach are [33]: the coarticulation effects over longer units - which cannot be captured by the diphones; and the concatenation errors - diphones taken from different contexts have different amplitudes and pitch values.

The best concatenative synthesis solution is unit selection, which uses variable length speech samples. The samples are selected using scores to determine the best match, and can be phonemes, diphones, syllables, words, or even entire phrases. The speech corpus design is minimum, although extended databases provide better results. Coarticulation problems are solved in unit selection by introducing the target cost (**Eq. 2.1**) and concatenation costs (**Eq. 2.2**)[34]. Target cost represents the cost of selecting a particular unit from the database, while concatenation cost is the cost of using that unit in the utterance. The best unit is selected using a Viterbi-like search algorithm over the two cost functions.

$$C_{target}(u_i, u_s) = \sum_{j=1}^{N} w_t^{(t)} c_j^{(t)}(u_i, u_s), \qquad \textbf{(2.1)}$$

u_i *represents the candidate unit, and* u_s *the current unit.*

$$C_{concatenation}(u_{i-1}, u_i) = \sum_{k=1}^{M} w_k^{(c)} c_k^{(c)}(u_{i-1}, u_i), \qquad \textbf{(2.2)}$$

u_i *represents the newly selected unit, and* ***u_{i-1}*** *the previously selected unit.*

- **Statistical Parametric Synthesis**

Unit selection synthesis, although providing one of the best quality synthetic speeches, lacks the flexibility of the output speech. Quality is directly determined by the speech corpus and the unit selection algorithms. Parameterizing the speech waveform is a solution to the generalization problem of the synthetic speech in concatenative systems. Parametric corpus-based synthesis implies the use of a pre-recorded speech corpus from which it extracts a selection of parameters. Thus speech synthesis becomes a statistical analysis of a speech corpus. Parameters are clustered according to context and prosodic features.

The most important parametric technique is the one based on hidden Markov models (HMMs), a concept borrowed from automatic speech recognition and with very good applicability and flexibility within speech synthesis as well. A first attempt to model speech using HMMs is that of [36] , but the results were unnatural and did not come to the attention of the specialists. With the introduction of the HMM-based Speech Synthesis System (HTS) [37], some of the initial problems were solved, and this method became the choice for research in speech synthesis. In HTS, speech is modeled through a 3 state HMM for each phoneme. Each state includes mel-frequency-cepstral coefficients and F0 with their delta and delta-delta features, and state duration. Decision trees are employed for the context clustering of the feature vectors to ensure no low or zero occupancy states. Contextual factors include phonetic, accentual and syntactic features.

From the target phoneme sequence a sentence of HMMs is derived using the Maximum Likelihood (ML) algorithm. Over smoothing of the spectral sequence is partially solved by the global variance (GV) principle [38], which maximizes the dynamic variation of the speech parameters.

The approach uses formants as acoustic observation, thus trying to overcome some of the formant synthesis problems [39]. A very good method of parameterization is that of [40], called STRAIGHT and which uses source and spectral parameters in the form of: a mixed-excitation model based on a weighted combination of fundamental frequency and noise, and a set of aperiodicity bands.

The advantages of the parametric synthesis refer to [40]:

- the small footprint necessary to store speech information;
- automatic clustering of speech information - removes the problems of hand-written rules;
- even small training corpora can result in good quality of the synthetic speech, if the corpus is well designed;
- generalizable - even if for a certain phoneme context there is not enough training data, the model might be clustered along with similar parameter characteristics;
- Flexibility - the trained models can be easily adapted to other speakers or voice characteristics with minimum amount of adaptation data.

And of course, there are also disadvantages such as low speaker similarity due to the use of a parameterization method which cannot capture the fine details of speech [40]. Training on a large database leads to high computational requirements during the training stage, but the synthesis part is still minimum computational consuming. Another disadvantage, but can be also considered as an advantage is the fact that the output is highly dependent on the parameterization method, which can be modified and adapted according to new researches.

In conclusion, TTS is used in many applications. However, although most text-to-speech systems still cannot synthesize speech with various voice characteristics such as speaker individualities and emotions. To obtain various voice characteristics in text-to-speech systems based on the

selection and concatenation of acoustical units, a large amount of speech data is necessary. However, it is difficult to collect, segment, and store it. From these points of view, in order to construct a speech synthesis system which can generate various voice characteristics, HMM-based text-to-speech system is a suggested solution.

2.5 HMM-based Speech Synthesis

HMM-based Speech Synthesis is a text to speech system using Markov models.

2.5.1 The Hidden Markov Model

A hidden Markov model is a finite state machine which generates time discrete observations [41]. In a Markov chain, each state corresponds to a deterministic observable event. Non-deterministic processes are the input of the hidden state models and the output is any of model's state. So that, an observation is a state dependent probabilistic function. It therefore exists a hidden stochastic process which cannot be observed. The hidden process can only be associated with another observable process, producing a series of observable characteristics. At each time sample, HMM modifies its states according to a transition probability and generates the observation **o** according to the probability distribution of the current state.

A continuous HMM is described according to [41] as follows:

- **o** - an output observation data. The observation data corresponds to the physical output of the system being modeled
- **ω** = 1, 2,…N - a set of states representing the state space. Here s_t is denoted as the state at time t
- **A = a_{ij}** - a transition probability matrix, where a_{ij} is the probability of taking a transition from state i to state j, i.e. $a_{ij} = P(s_t = j|s_{t-1} = i)$

- **B = $b_i(o)$** - an output probability distribution. The output probability distribution $b_i(o)$ of the observational data **o** of state i is modeled by a mixture of multivariate Gaussian distributions according to ,

  $b_i(o) = \sum_{m=1}^{M} w_{im}\mathcal{N}(o; \mu_{im}, \Sigma_{im})$

  where M is the number of mixture components of the distribution, and w_{im}, μ_{im} and Σ_{im} are a weight, a L-dimensional mean vector, and a L x L covariance matrix of mixture component m of state i, respectively.

  A Gaussian distribution $N(o; \mu_{im};\Sigma_{im})$ of each component is defined by, where L is the dimensionality of the observation data **o**.

$$\mathcal{N}(o; \mu_{im}, \Sigma_{im}) = \frac{1}{(2\pi)^L|\Sigma_{im}|} exp\left(-\frac{1}{2}(o-\mu_{im})^\top \Sigma_{im}^{-1}(o-\mu_{im})\right)$$

- $\pi = \pi_i$ an initial state distribution where $\pi_i = P(s_o = i), 1 \leqslant i \leqslant N$

The following properties must be satisfied:

$$a_{ij} \geqslant 0, w_{im} \geqslant 0, \pi_i \geqslant 0, \forall i, j, m$$
$$\sum_{j=1}^{N} a_{ij} = 1, \ i = 1...N$$
$$\sum_{m=1}^{M} w_{im} = 1, \ i = 1...N$$
$$\sum_{i=1}^{N} \pi_i = 1, \ i = 1...N$$
$$\int_o b_i(\mathbf{o})d\mathbf{o} = 1$$

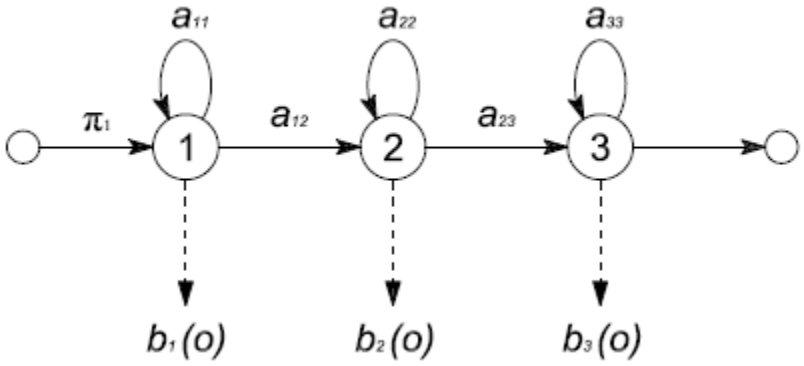

Figure 2.5: Example of a left to right HMM structure

To sum up, a complete specification of an HMM includes two constant-size parameters, N and M the total number of states and the number of mixture components, w_{im} the Gaussians weights, the observational data **o**, and three sets (matrices) of probability measures A; B; π in the following notation [41]:

$\phi = (A; B; \pi)$ to indicate the whole parameter set of an HMM.

In speech processing, the most common HMM structure used is the left-right one (as shown in **Figure 2.5**). In this structure, the state index is incremented or remains constant. This type of model approximated correctly the speech signal, whose characteristics modify over time.

2.5.2 Speech Signal Parameter Modeling

As a parametric synthesis method, HMM-based speech synthesis needs a set of features extracted from the speech in order to estimate its inner models. In the HMM-based speech synthesis, the speech parameters of a speech unit such as the spectrum, fundamental frequency (F0), and phoneme duration are statistically modeled and generated by using HMMs based on maximum likelihood criterion [41]. The speech is analyzed at frame level and develops context-based models for each phoneme. The spectrum features are represented by the mel-Frequency Cepstral Coefficients, similar to the method used in automatic speech recognition. The following subsections describe the extraction of the feature vector and the building of the HMMs.

- **Mel-Cepstrum Analysis**

In speech analysis, the most common model for speech production is the source-filter model. Within the mel-cepstrum analysis, the transfer function of the vocal tract, H(z) is modeled by the mel-cepstrum coefficients (MFCC). This representation is obtained by applying the discrete Fourier transform (DFT) over a speech frame. The Fourier spectrum is then filtered through a Mel-scale frequency filter bank. From each sub band, the log of the power is computed and the

discrete cosine transform (DCT) is applied to the result. The MFCCs are the amplitudes of the resulting spectrum (as shown in **Figure 2.6**).

Figure 2.6: MFCC coefficients computation

The advantage behind this type of representation is the fact that these coefficients remain independent and allow for a probability distribution modeling by a diagonal covariance matrix. Along with the MFCC coefficients, the feature vector also includes the delta and delta-delta coefficients of the MFCC.

- **Fundamental Frequency Modeling**

MFCC models the spectrum of the speech, while an important characteristic of speech is the pitch or fundamental frequency. Because of the lack of pitch values in the unvoiced segments, F0 cannot be modeled using conventional discrete or continuous HMMs. Thus, a new type of HMMs are defined, the Multi-space Probability Distribution HMM (MSDHMM) [42]. In order to model the pitch using MSD-HMMs two spaces are defined: a one dimensional space with a probability density function for the voiced segments and a zero dimensional space containing a single point for the unvoiced segments. In this way, F0 can be modeled without making any heuristic assumptions of its values.

- **HMM state duration Modeling**

In standard HMM models, the transition probabilities determine the duration characteristics of the model. In phoneme synthesis, the duration must be explicitly specified, because of its major influence in the speaker characteristics and in the rhythm and prosody of speech. Another type of

HMM models is so defined. They are called Hidden Semi-Markov Models (HSMM) and the transition probabilities are replaced by explicit Gaussian models for the duration [42].

2.5.3 Decision Tree Building for Context Clustering

In continuous speech, the parameter sequences of an acoustic unit vary according to the phonetic context. The correct modeling of these variations implies context dependent models, such as triphones or quinphones. In HMM-based speech synthesis systems, the context is defined by both the phonetic and the linguistic and prosodic context. A contextual clustering is achieved using binary decision trees (see **Figure 2.7**). Each tree node defines a cluster based on a contextual factor. Each tree leaf contains an output probability distribution of the state. The trees are built using the Minimum Description Length (MDL) principles are used to cluster pitch, duration and spectrum.

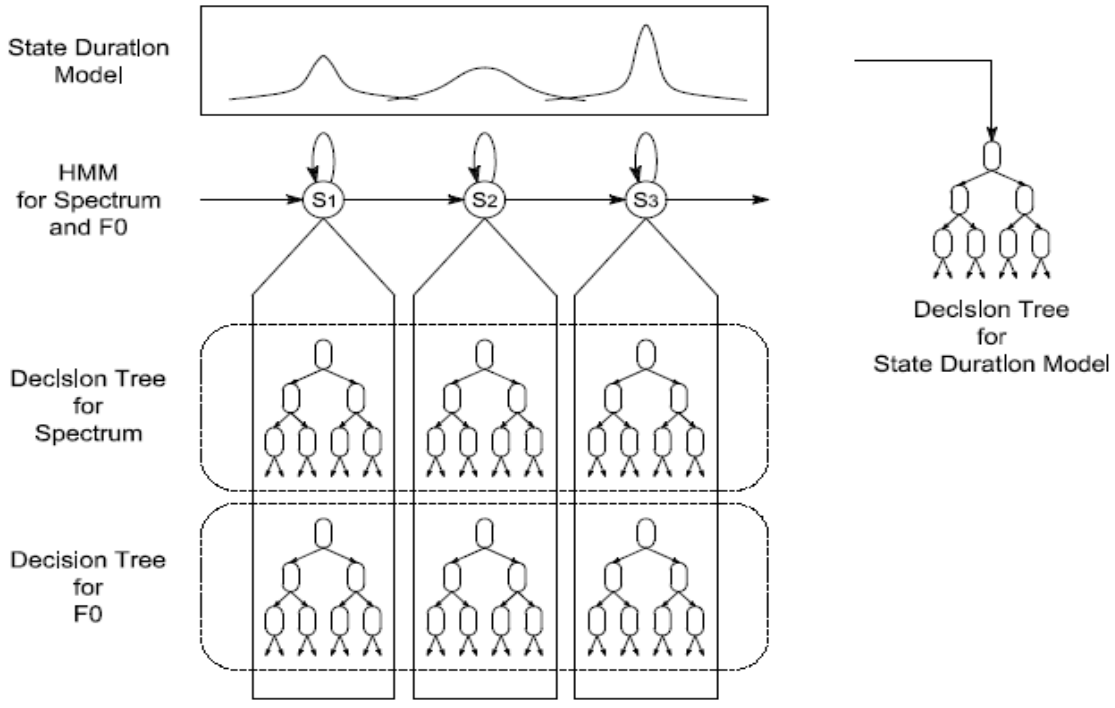

Figure 2.7: Decision tree context clustering in HMM-based speech synthesis [42].

2.5.4 Speech Parameter Generation

The input labels of the HMM-based speech synthesizer offer information about the phoneme sequence, but not about the HMM states that should be used in synthesis. To determine the state sequence, the Maximum Likelihood (ML) algorithm is applied. The MFCC coefficients are synthesized using a Mel Log Spectrum Approximation (MLSA) filter[43].

2.5.5 The HMM-based Speech Synthesis System

The HMM-based Speech Synthesis System (H Triple S - HTS) is a collection of open source tools dedicated to the development of text-to-speech systems using Markov models[44]. The content of these tools refer strictly to the modeling, training and speech generation without text processing. The system input is represented by the HTS labels presented in next section.

HTS is built on the Hidden Markov Model Toolkit (HTK)[45]. HTK was initially developed for automatic speech recognition. The training part of the HTS is a modified version of HTK. In **Figure 2.8** the block diagram of the HTS system is presented. Two main sections can be observed: training and synthesis. In the training section, the spectrum, pitch and duration HMM models are extracted. Decision tree clustering is applied to the MFCC coefficients, pitch values and duration. The HMMs are re-estimated using a Baum-Welch algorithm. The result of the training section is the decision tree clusters with their respective parameters in the leaf nodes.

The synthesis section uses HTS labels to generate phoneme level HMM state sequences. The MLSA filter is then applied to generate the synthetic speech from the state parameters. HTS is very flexible and allows for the following parameter modification, both in the training and in the synthesis sections [45]:

- training data set
- sampling frequency

- non-linear transformation of the frequency scale
- analysis/synthesis frame length
- cepstral order
- analysis method: STRAIGHT Mel-cepstrum, STRAIGHT Mel-generalised cepstrum, STRAIGHT Mel-LSP, STRAIGHT Mel-generalized LSP
- pitch estimation method: IFAS, Fixed-Point analysis, ESPS, voting between previous methods
- number of HMM states
- the root node decision tree question

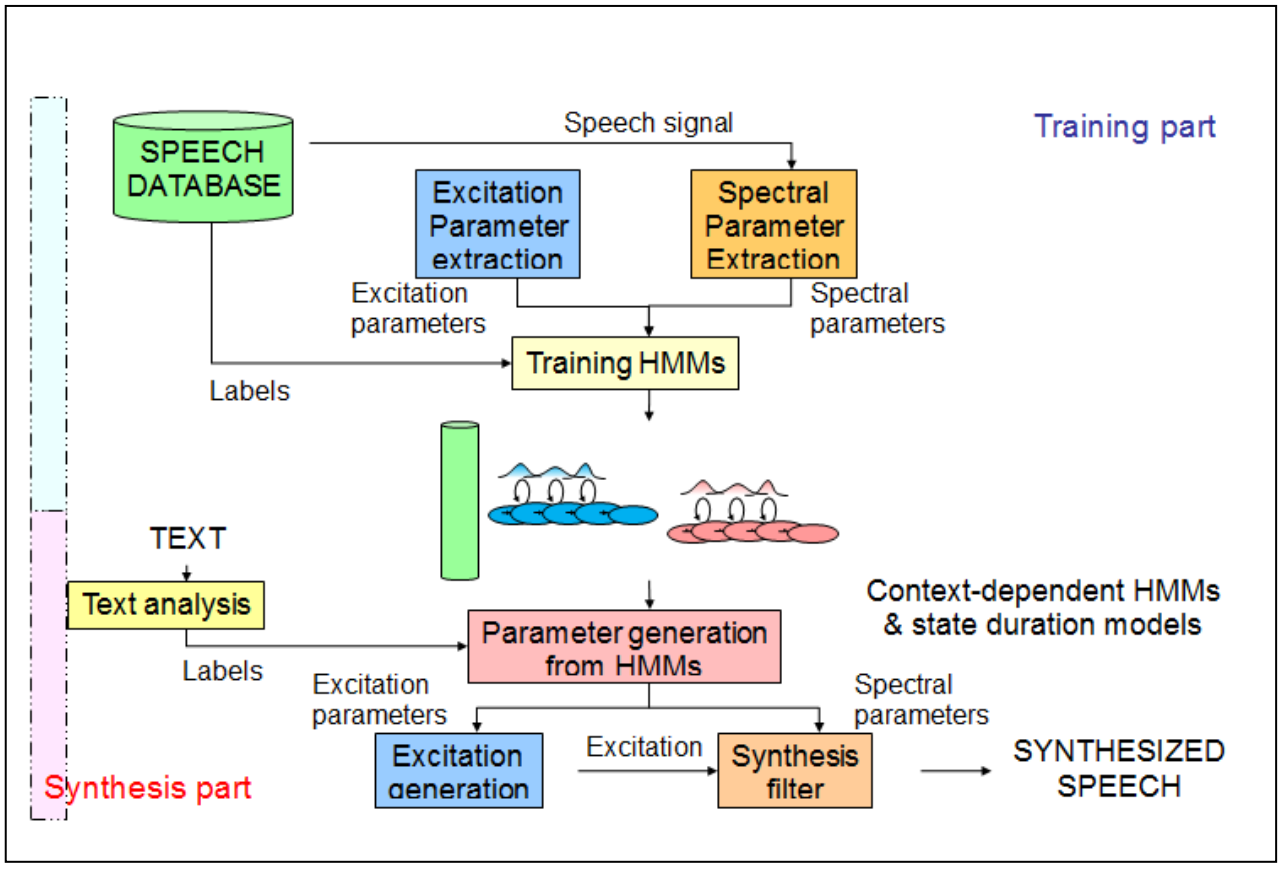

Figure 2.8: HMM speech Synthesis system [50]

An important development of the HTS system is the speaker conversion described by[46]. Starting from the speaker dependent or independent trained decision trees, using Maximum Likelihood Linear Regression (MLLR), the models are adapted to a new training data. The results of the adaptation are not considered high-quality, but the amount of necessary training data is largely reduced[47]. Speaker adaptation represents one of the major advantages of parametric synthesis over the concatenative system. Studies show that an average of 5 minutes recordings can capture the gross features of a new speaker if the starting models are speaker independent.

2.6 Applications of Speech Synthesis Systems

Speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread in our life. It allows environmental barriers to be removed for people with a wide range of disabilities. The longest application has been in the use of screen readers for people with visual impairment, but text-to-speech systems are now commonly used by people with dyslexia and other reading difficulties as well as by pre-literate children [17].

Text To Speech systems applications can be group as follows: Aid-to-the handicapped, Education, interactive voice response (IVR) systems, entertainment applications, and speech-to-speech translation [1].

- **Aid-to-the handicapped:** The most important application of TTS is the aid to blind people in various areas such as: reading books, easy use of different devices like computers and talking calculators.
- **Education:** Text to speech systems can be used in learning different languages.
- **Entertainment applications:** like reading e-mail while driving, or other applications which belong to car navigation systems.

- **Interactive voice response (IVR) systems:** IVR is a system that allows interaction between a computer and humans. The input to this system is entered by a keypad or by a speech recognition module and the output speech is produced either from pre-recorded messages or from a text to speech system. An important use of IVR systems is in the field of customer service in companies, hospitals, and etc, which reduces the cost and increase the service efficiency.
- **Speech-to-speech translation (S2S):** speech-to-speech translation between different languages enables cross-lingual communication. S2S requires speech recognition, machine translation and text-to-speech synthesis.

After modeling and implementing the TTS system for Amharic it is possible to develop and apply any of the above applications especially for those who are visually impaired and reading disabilities people as an assistive technology.

2.7 Amharic Syllabification

2.7.1 Syllabification

Syllabification is the task of segmenting a sequence of phonemes into syllables [51]. A syllable is a unit of sound composed of a central peak of sonority (usually a vowel), and the consonants that cluster around this central peak. Syllabification has importance in a variety of speech applications. For instance, in speech synthesis, syllables are important in predicting prosodic factors like accent. The realization of a phone is also dependent on its position in the syllable (onset is pronounced differently than coda). In speech recognition syllabification has been used to build recognizers which represent pronunciations in terms of syllables rather than phonemes[51], it also helps to detect out of vocabulary words.

Syllabification includes the separation of a word into syllables, whether spoken or written. Speech is organized into syllables. Although nearly everybody can identify syllables, almost nobody can define them. It is difficult to state an objective procedure for locating the number of syllables in a word or a phrase in any language. There are words difficult to be agreed upon in determining the number of syllables contained, but it is important to remember that there is no doubt about the number of syllables in the majority of words. Syllable is a unit larger than a single segment and smaller than a word, and this characteristics can be described from both a phonetic and a phonological point of view, one of which is distinguished from the other, although the differentiation is not yet agreed upon by all scholars[52].

From phonological standpoint syllable is a conventional unit which is a group of sounds that constitute the smallest unit of the rhythm of a language. These phonological syllables differ from language to language. In English, for example, it is theoretically possible to make a single syllable as CCCVCCCC [52], where previous studies related to the syllable structure of the standard Amharic have shown that the following syllable types V, VC, VCC, CV, CVC and CVCC occur as part of the phonological system of Amharic [53] [54]. The syllable, in this view, is considered as an important abstract unit explaining the way vowels and consonants are organized within a sound system. Technically, the basic elements of the syllable are the onset (zero or more consonants) and the rhyme. The rhyme (sometimes written as 'rime') consists of a vowel, which is treated as the nucleus, plus any following consonant(s), described as the coda[55]. The internal organization of syllables characterized as in **Figure 2.9**.

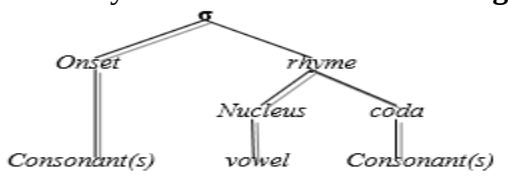

Figure 2.9: Syllable structure σ –syllable

Languages differ considerably in the syllable structures that they permit. For most languages, syllabification can be achieved by writing a set of declarative grammatical rules which explain the location of syllable boundaries of words step-by-step. It has been adhered to the well-known principles “the Maximum Onset Principle” and “the sonority hierarchy principle”.

2.7.2 Syllabification model for Amharic

According to Nirayo [56], having gemination handling rules, syllabification rules, epenthesis (epenthetic vowels insertion) rules and syllable templates of the language it is possible to syllabify (mark syllable boundaries) given the Amharic text. Moreover, we have the syllable templates of the language from different linguistic literatures in accordance with empirical experiment. We can have general syllabification model at this level. **Figure 2.10** shows the general syllabification model for Amharic language.

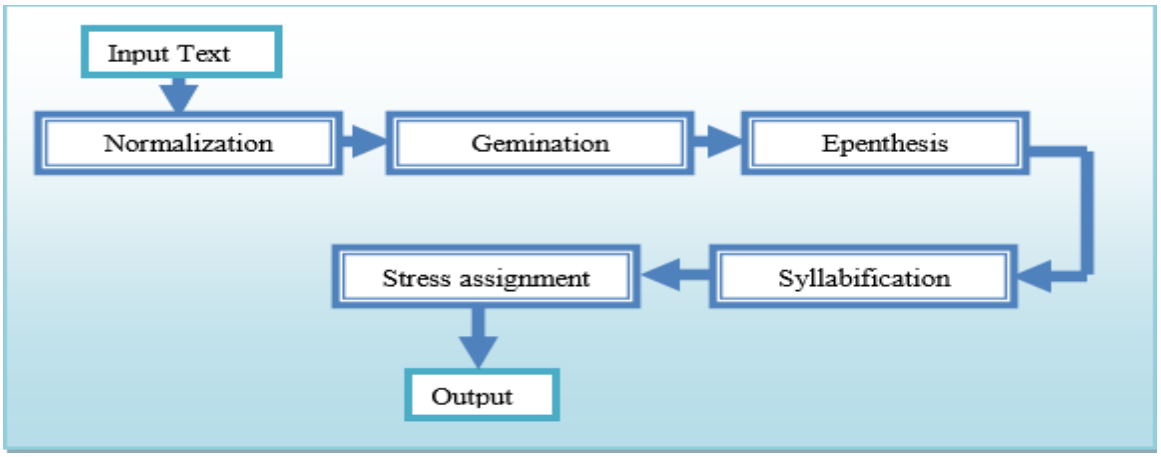

Figure 2.10: General automatic syllabification model for Amharic text

As it is shown in the model (**Figure 2.10**), as an input to activate the module, Amharic text is used. Then, in the normalization module normalization is carried out. The normalized text is again passed to the gemination module as an input and gemination is takes place applying the gemination rule of Amharic language. At this point, epenthesis can be carried out; the

gemination module final result is used as an input for this module. After applying insertion of epenthetic vowel based on the rule of epenthesis in the language, syllabification is done by the syllabifier module. The syllabifier applies syllabification using an algorithm to syllabify texts in their legal sequence. Finally, by examining the syllable sequences in accordance with their syllable weight and the stress assignment rules, stress assignment is takes place in the final module. The final output will be stress and syllable boundary marked transcribed Amharic text.

In the following sections we have seen the rules of the language in relation with epenthetic vowel insertion and gemination handling.

- **The issue of Gemination in Amharic words**

One of the important features of Amharic phonology that should be handled in automatic syllabification is gemination of consonants. In phonetics, gemination happens when a spoken consonant is pronounced for an audibly longer period of time than a short consonant. Although double consonants (sequence of the same consonants or consonant clusters) and geminates are phonetically (relating to speech sounds) the same, they are phonologically different. Gemination is distinct from stress and may appear independently of it, it is a doubling of consonants.

In Amharic, all consonants except (ሀ)/h/ and (ዕ) /ax/ may occur in either a geminated or a non-geminated form. Amharic gemination is either lexical or morphological. As a lexical feature it usually cannot be predicted. The failure of the orthography of Amharic to show geminates is the main challenge in Grapheme-To-Phoneme (GTP) conversion [58].

Gemination occurs in many of Amharic words for example; we can observe the difference between /kixft/ and /kixffixtt/ because of the geminate consonant /f/ (ፍ) and /t/ (ት).When /f/ and /t/ are geminated there is epenthesis vowel /ix/ inserted between them therefore syllabification for the two words becomes completely different. Without gemination the word contains only

single syllable type CVCC. Therefore, the whole phoneme sequence, /kixft/, is taken as a single syllable. But when it is geminated, the word will have two CVC syllables, /kixf-fixtt/.

- **The issue of epenthesis in Amharic words**

The process of epenthesis is common in Amharic. It can occur word-initially or medially. As Hudson [58] stated epenthesis is extensive in word-formation in the Ethiopian Semitic languages, since many morphemes, both roots and affixes, consists only consonants. Amharic epenthesis vowel may be said to provide almost all occurrences of the high central vowel /ix/ (ɨ).

There are two general rules concerning automatic insertion of an epenthetic vowel in Amharic.

1. Word-initially no consonant clusters are allowed.
2. Elsewhere clusters of no more than two consonants are tolerated.

Hudson [58] also proposes three environments for epenthesis corresponding to the possibilities of consonants sequence word initial, medial and final position in Amharic. #CC as in words like /tsebr/ ->/tixsber/, /sber/ ->/sixber/ (word initial consonant clusters are impermissible) (the (#) indicates the position of the cluster, word initial or word final). CC# as in word like/mkr/-> /mixkixr/. In this case, the sonority of the final consonant, /r/, is greater than that of the preceding consonant, /k/. Thus, to split up the final cluster epenthetic vowel /ix/ is inserted. On the other hand, if the sonority of the first is equal or greater than that of the second consonant, epenthesis will not be applied. Hudson[58] proposes three types of CCC violation where epenthesis /ix/ is required.

a. CCC-> CCixC, in a word like /fendto/-> /fendixto/ “exploit”

b. C:C-> C:ixC, in a word like /fellgo/->/fellixgo/ “want”

c. CC:->CixC:, in a word like /sebrre/->/sebixrre/ “break”

Mulugeta [54] also discussed further about the process of epenthetic vowel in Amharic by dividing into six sections, we present all of them with examples as follows.

1. Word initially no consonant cluster—the sonority of the initial consonant is greater than that of the following consonants for word initial cluster, the epenthesis is inserted before the cluster.
2. If a word medial cluster of consonants contains the geminate and singleton in sequence, the epenthesis vowel is inserted after the geminate consonants.

   Examples: /fel:go/->/fel:ixgo/,/mel:so/-> /mel:ixso/, /lem:no/->/lem:ixno/ etc.
3. If word medial cluster of consonants contains a singleton and geminate in sequence, the epenthesis is inserted before the geminate consonant.

   Examples: /sebr:e/->/sebixr:e/, /gedy:ie/->/gedixy:ie/, /txrg:ie/-> /txrixg:ie/.
4. If word medial or final cluster of consonants contain two geminate consonants in sequence, the epenthesis inserted between the two different geminates.

   Examples: /sbbrr/->/sixbbixrr/, /kfftt/->/kixffixtt/.
5. If three consonants are appeared in sequence word medially, the epenthesis vowel is inserted before the third consonant.

   Examples: /sentxqo/->/sentixqo/, /fendto/-> /fendixto/, /bergdo/-> /bergixdo/.
6. If the sonority of the final consonant is greater than the preceding consonant, the epenthesis can be inserted between the final clusters.

   Example: /dngl/->/dixngixl/ “virgin”, /tsebr/-> tixsbixr “may you break”, /mkr/-> /mixkixr/ “advice”.

2.8 Related Works

Currently, researches in speech synthesis are getting more attention by different scholars. Many techniques are available to handle speech synthesis, but hard it is to find a best method that satisfies the naturalness and intelligibility of the synthesizing synthetic speech. Since Ethiopia is a multi-linguistic society, which needs multi-lingual synthesizer and tried by different scholars. Taking the advantage of speech synthesis systems into consideration, some interesting works have been done on speech synthesis for the local languages. In the following sections, different works related to speech synthesizers for local languages will be discussed. Issues are related to this research either by the language features or the synthesis technique used.

First Text to speech synthesis for the Amharic Language, regarding speech synthesizer for local language was done by Laine [61]. The technique that he used in his research was concatenative speech synthesis technique and diphones were used as the basic concatenation units. There are different methods to produce the synthesized speech using concatenation, like LPC (Linear Predictive Coding), TD-PSOLA (Time-domain Pitch Synchronous Overlap ADD), and FD-PSOLA (Frequency-Domain Pitch Synchronous Overlap ADD). Though he did not explain why he chooses it, out of these techniques, he has used the linear predictive coding method (LPC).

A synthesizer which follows a formant synthesis approach to generate a speech for a given Amharic input text was developed by Yibeltal Tefera[62]. The developer collected speech for voiced sounds and extracted parameters such as formants, bandwidth, pitch, etc from the collected speech. The unvoiced sounds were also stored by segmenting them from all Amharic syllables. He 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. The system works on word level.

In the research work presented in [14], Tadesse *et al.* developed a syllabic rule-based (formant) TTS system with prosodic control method for Amharic. The designed Amharic TTS (AmhTTS) is parametric and rule-based system that employs a Cepstral method and uses a Log Magnitude Approximation (LMA) filter. They claim that their study provides a total solution on prosodic information generation mainly by modeling the durations.

A research work done by Henock [64], applied concatenative as synthesis techniques, speech units of diphone and syllable to synthesize sample and time domain pitch synchronous overlap-add (TD-PSOLA) algorithm to analyze and generate synthetic speech. In addition to this, the researcher also considered prosodic factors. For the system performance evaluation, the researcher adopted Open Rhyme Test method to show the system performance and achieved result is 88% and 75% for diphone and syllable respectively. Based on result analysis, the researcher concluded that Diphone based synthesis gives better result than syllable based as speech unit selection. The researcher suggested that syllable as speech units and concatenative technique with large speech database can be applied on different local Ethiopian language for future work.

Amharic language TTS system done by Sebsibe [12], was called “Unit selection voice for Amharic using Festvox”, which developed a unit selection concatenative speech synthesizer by using transliteration scheme to work with Amharic scripts and incorporated Amharic phone set, syllabification rules, letter to sound rules into Festvox. The system achieved cumulative result of 2.9. In this work, the researcher suggested as future work that for Amharic language the proper selection of unit and optimal selection of corpus will give a better quality of speech waveform from the synthesizer [12].

Bereket Kasay [13] had applied the Hidden Markov Model (HMM) to develop Amharic Speech Synthesizer. Parameters such as mel-cepstrum coefficients and fundamental frequencies along with festival's and festvox's utterance structure were used in training the model. In his work, the Mean Opinion Score (MOS) evaluation technique is used. The results from the MOS were found to be 4.12 and 3.6 for intelligibility and naturalness respectively for speeches synthesized by his system and Using concatinative method the result obtained for intelligibility and naturalness are 3.54 and 3.25 respectively.

This thesis relate with all those except Bereket [13] work which are based on statistical speech synthesis method (i.e. HMM). According to reviewed literatures so far, this approach over other speech synthesis methods was inspired the researcher by its ability to synthesize intelligible and natural sounding speech without requiring a huge training corpus. Low memory requirements, flexibility, and ease of adaptability to speaker's voice characteristics and speaking styles using the HTS toolkit are some of the factors that favored the choice of this method of speech synthesis over other methods. This method achieves its task by statistically modeling speech parameters using HMM. Furthermore, the runtime synthesis engine of HTS – the toolkit used for HMM-based speech synthesis – is considerably small, spanning only a few megabytes (MBs), when excluding the text analysis component which is done by festival and festvox toolkit. It is, therefore, easy to implement a system built using HTS on multiple platforms, including those associated with handheld devices. And finally, the only difference with Bereket work is the basic unit selection and the corpus used. His research work taken as a pioneer using HMM based speech synthesis method for Amharic language, phone as a basic unit. But in our work, syllable is taken as a basic unit that are outperformed phone and used ASR corpus.

CHAPTER THREE

AMHARIC LANGUAGE, WRITING SYSTEM AND ITS PHONETICS

This chapter presents about Amharic language and its writing system and how to transcribe to its phoneme sound representation. The next is about phonetics which refers to the study of speech sounds used in the language. Phonetics is concerned with the sounds of the language, how these sounds are articulated and how the listener perceives them. Finally, Amharic words syllable structure, stress and syllable revised.

3.1 Amharic language and its Orthography

Amharic is the official working language of government of Ethiopia, among 73 languages which are registered in the country [2]. Amharic is second largest spoken Semitic language in the world next to Arabic [3]. Unlike other Semitic languages, such as Arabic and Hebrew, Amharic /amarixnxa/ script uses a grapheme based writing system called fidel /fidel/ which is written and read from left to right [5]. Modern Amharic has inherited its writing system from Ge'ez /gixz/, which is still the classical and ecclesiastical language of Ethiopia [6].

The Ethiopic alphabet has 33 basic characters. Each of such character is modified in some regular fashion to reflect the seven vowels of the language. Therefore, there are in total 33 x 7 = 231 characters. Even though Amharic alphabet is Unicode standard, it is sometimes convenient to represent it in ASCII. Written in SERA (System for Ethiopic Representation in ASCII) [65], the basic characters which can also be called the consonants of the language are in alphabetic order: C = {h/ሀ; l/ለ; H/ሐ; m/መ; s/ሰ; r/ረ; ’s/ሠ; x/ሸ; q/ቀ; b/በ; t/ተ; c/ቸ;’ h/ኀ; n/ነ; N/ኘ; a/አ; k/ከ; K/ኸ; w/ወ;’ a/ዐ; z/ዘ; Z/ዠ; y/ደ; d/ድ; j/ጀ; g/ገ; T/ጠ;C/ጨ; P/ጰ; S/ፀ; ’ S/ጸ f/ፈ; p/ፐ}

The vowels are:

V = {e/ኧ; u/ኡ; i/ኢ; a/ኣ; E/ኤ; I/እ; o/ኦ}

Out of the 33 basic consonants, Amharic identifies 28 unique sounds. This implies that, in some cases, more than one consonant is used to represent the same sound.

These are: {h, H, h'}, {s, s'}, {S,'S} and {a, a'}

The above sets represent different sounds. The letters in each set represent the same sound. It is important to recognize these letters in natural language processing tasks. For example, an Amharic word that has the letter h can be written in three equivalent ways and must be treated as one for NLP tasks. There are speech sounds of Amharic that are specific and not found in any other foreign language [66]. These include sounds such as ጵ/p'/, ጥ/t'/, ፅ/s'/, ጭ/ʧ'/, and ቅ/k'/ which have a sharp click–like characters beside glottalized voice articulating at different places. Amharic symbols are categorized into four different categories consisting 276 distinct symbols; these are core character, labiovelar, labialized and labiodental. The detail category is presented in **Table 3.1**.

Category	**Character set**	**Order**	**Total**
Core characters	33	7	231
labiovelar	4	5	20
labialized	18	1	18
labiodental	1	7	7
total			276

Table 3.1: Distribution of Amharic character set

Amharic has a total of 231 (33*7) distinct core characters, 20 (4*5) labiovelar symbols, 18 labialized consonants and 7 labiodental. The first category possesses 33 primary characters each representing a consonant having 7 orders in form to indicate the vowel which follows the

consonant to represent CV syllables. **Table 3.2** shows sample core characters used in Amharic writing system with their seven orders. List of all Amharic core characters found in **Appendix A**. In the same way, labiodental category contains a character ቨ/v/ with 7 order (ə, u, i, a, e, ɨ, o) borrowed from foreign languages and appears only in modern loan words like ቫይታሚን /vayixtamin/. Similarly, the labiovelar category contains 4 (ቀ/k'/, ኀ/h/, ከ/k/ and ገ/g/) characters with 5 orders (^{w}ə, ^{w}i, ^{w}a, ^{w}e, and ^{w}ɨ) that generates 20 distinct symbols. Furthermore, there are labialized 18 characters for instance ሏ/ l ^{w}a /, ሟ/m ^{w}a /, ሯ/r ^{w}a / and ሷ/s ^{w}a /.

First	Second	Third	Fourth	Fifth	Sixth	Seventh
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ

Table 3.2: Sample Amharic core characters

There are also characters in Amharic other than mentioned above. Punctuation marks such as ፡ (Ethiopic word space), ። (Ethiopic full stop), ፣ (Ethiopic comma), ፤ (Ethiopic semicolon), and borrowed symbols like ?, !, (,), and Numerals consisting of a single character for 1 to 10, for

multiples of 10 (20 to 90), for 100 and 1000. The numeral symbols are shown in **Table 3.3**.

1	፩	6	፮	20	፳	70	፸
2	፪	7	፯	30	፴	80	፹
3	፫	8	፰	40	፵	90	፺
4	፬	9	፱	50	፶	100	፻
5	፭	10	፲	60	፷	1000	፼

Table 3.3: The numeral symbols of Amharic

The style of the writing was also modified from left to right. By the time Ge"ez ceased to be a living spoken and written language and replaced by Amharic and other languages, further changes took place. Amharic did not discriminate in adopting the Ge"ez fidel; it took all of the symbols [67] and added some new ones that represent sounds not found in Ge"ez. These added alphabetic characters are ቸ, ጨ, ጀ, ኘ, ቨ, ሸ, ኸ, and ኘ. In Amharic there is no Upper-Lower case distinction.

3.2 Transcription System

As stated in [17], transcription is needed because written text in most 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) [68] and SAMPA [69] 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, suprasegementals, tones/word contours, and diacritics. On the other hand, SAMPA is designed to map IPA symbols to 7-bit printable ASCII characters to alleviate the complexity and the use of

Greek symbols that make IPA alphabets unsuitable 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 [17]. Transliteration can be used as an alternative to using the IPA alphabets [64].

It is designed based on the orthographic ordering of the script and acoustic similarity of the letters. The transliteration scheme used in [64] is adopted in this study.

The script of Amharic language is phonetic in nature. In order to represent Amharic characters to their corresponding sounds, ASCII transliteration system is used throughout this thesis work that attached in **Appendix B** [70] and this transliteration is accomplished automatically with Python code developed for this purpose which takes ASCII translation table expanded for all CV syllable as input that attached in **Appendix C**.

The translator also normalized to some common sounds since in Amharic language there are some characters which have different orthography but the sound is the same like ሀ, ሐ, ኀ, ፅ, ሃ and ሰ, ሠ. **Table 3.4** shows, the ASCII transcription of three consonants with their 7 orders.

In case of Amharic character ሀ, it is represented by “ha” instead of using “he” unlike other Amharic character sets/orthographies since the sound is the same as its 4th order.

order	1st	2nd	3rd	4th	5th	6th	7th
consonants	vowels						
	e	u	ii	a	ie	ix	o
/m/	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
/b/	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
/l/	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ

Table 3.4 The Amharic consonants with the vowels (using ASCII translation)

3.3 Amharic Phonetics

Phonetics is the scientific study of speech whose central concerns are the discovery of how speech sounds are produced, how speech sounds are recorded with written symbols, and how we hear and recognize different sounds [71]. Amharic language is primarily comprised of 39 phonemes – 7 vowels and 31 consonants [72]. One additional consonant /ቨ/ [v] is inherited and included summing up to a total of 39 phonemes. These phonemes are categorized into vowels and consonants. The following subsections brief about how consonants and vowels are produced in Amharic language.

3.3.1 Vowels

Vowels have different categories based on the position and height of the tongue and their shapes during speech production. The seven vowels in Amharic language include: /ኧ/, /ኡ/, /ኢ/, /ኣ/, /ኤ/, /እ/, and /ኦ/. All are oral and voiced. When we say the vowels are oral, we mean air flow passes through oral cavity during their articulation. Based on the tongue position in the oral cavity, vowels are classified into three sub categories that are front, central and back. Based on the height of the tongue, these vowels are also classified into high, middle and low. Based on their shapes during speech production, vowels are classified into two sub classes that are rounded and unrounded. The vowel classifications are indicated in **Figure 3.1** i.e. rows represent vowels classification based on the position of the tongue and columns represent vowels classification based on the height of the tongue. In addition to these classifications, the right represents a rounded vowel in which the lips are rounded (Among the Amharic vowels, /ኡ/ and /ኦ/) while the

left is its unrounded counterpart. The central vowels are also considered to be unrounded (/ኧ/, /ኢ/, /ኣ/, /ኤ/, and /ኧ/).

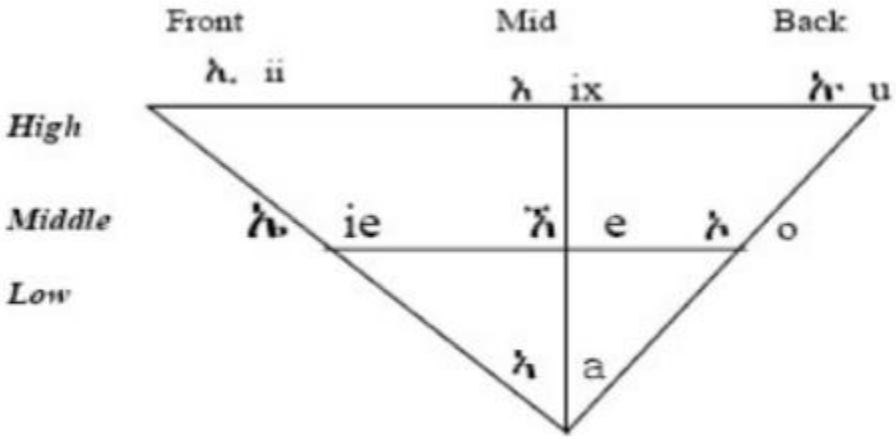

Figure 3.1: IPA maps of the Amharic vowels [12]

A description of Vowel Classification

By Height: the vertical position of the tongue relative to either the roof of the mouth or the aperture of the jaw (Low, Middle or High).

By Position: the position of the tongue during the articulation of a vowel relative to the back of the mouth (Front, Mid or Back).

Lip Rounding: The lips are (+) or not (-) in a rounded position.

3.3.2 Consonants

Consonant phonemes are described in the following categories: voiced vs. unvoiced; manner of articulation; place of articulation as shown in **Table 3.5**. The place of articulation means where the constriction is located in the vocal tract. Based on manner of articulation, consonants are categorized into stops, fricatives, affricatives, nasals, liquids and semivowels [73].

Described consonants **"By Voicing:" Voiced (+)**: Vocal cords vibrate; **Unvoiced (-)**: No vibration

Manner of Articulation	Voicing	Place of Articulation											
		Labials		Alveolar		Palatals		Velars		Labio-Velar		Glottals	
Stops	Voiceless	p	ፐ	t	ተ			k	ከ	kx	ኩ	ax	ዕ
	Voiced	b	በ	d	ደ			g	ገ	gx	ጉ		
	Glottalized	px	ጰ	tx	ጧ			q	ቀ	qx	ቁ		
Fricatives	Voiceless	f	ፍ	s	ሰ	sx	ሸ					h	ሀ
	Voiced	v	ቭ	z	ዘ	zx	ዠ						
	Glottalized			xx	ጸ							hx	ኀ
Africatives	Voiceless					c	ቸ						
	Voiced					j	ጀ						
	Glottalized					cx	ጭ						
Nasals	Voiced	m	ም	n	ነ	nx	ኘ						
Liquids	Voiced			l	ለ								
	Voiced			r	ር								
Glides		w	ው			y	ይ						

Table 3.5: Phonetic representation and characterization of Amharic consonants [12].

The detailed description about these characteristics and also the characters under different categories is given in the following paragraphs.

Stops

Stop consonants, also called plosives, are produced when the vocal tract is closed causing stop or attenuated sound. When the tract reopens, it causes noise-like, impulse-like or burst sound. The stop can occur at the lips, at the alveolar when the tip of the tongue touches it, at the palate when the middle of the tongue touches it, at the velum when the back of the tongue touches it, or at the glottis. As it can be seen from **Table 3.5**, Amharic consonants that are found in stop category include: /ፐ/, /በ/, /ጰ/, /ተ/, /ደ/, /ጠ/, /ከ/, /ገ/, and /ቀ/. These sounds, based on the place of articulation, are classified into: labial – (/ፐ/, /በ/, and /ጰ/), alveolar – (/ተ/, /ደ/, and /ጠ/), and

palatal – (/ክ/, /ግ/, and /ቅ/). In terms of voicing, /ፐ/, /ጵ/, /ት/, /ጥ/, /ክ/, and /ቅ/ are unvoiced or voiceless, while /ብ/, /ድ/, and /ግ/ are voiced.

Fricatives

Fricatives are produced when the vocal tract is constricted in some place so that the turbulent flow causes noise which is modified by the vocal tract resonances. Amharic consonants classified under this category include: /ፍ/, /ቭ/, /ስ/, /ዝ/, /ጽ/, /ሽ/, /ሾ/, and /ህ/. The constriction can occur between upper teeth and lower lip, the tip of the tongue and the alveolar, the middle of the tongue and the palate, or the back of the tongue and the velum. Based on this, /ፍ/ and /ቭ/ are labial, /ስ/, /ዝ/, and /ጽ/ are alveolar, /ሽ/ and /ሾ/ are palatal, and /ህ/ is glottal. Among the fricatives, /ዝ/, /ቭ/, and /ሾ/ are voiced, while /ፍ/, /ስ/, /ጽ/, /ሽ/, and /ህ/ are voiceless.

Affricatives

Affricatives have the characteristics of both stops and fricatives. An affricative is a type of consonant consisting of a plosive followed by a fricative with the same place of articulation. The three Amharic consonants categorized under this category are: /ች/, /ጅ/, and /ጭ/. All of them are palatal and among them /ጅ/ is voiced, while /ች/ and /ጭ/ are voiceless.

Nasals

A nasal consonant is the one in which the air escapes only through the nasal cavity during its articulation. For this to happen, two articulatory actions are necessary: firstly, the velum must be lowered to allow air to escape past it, and secondly, a closure must be made in the oral cavity to prevent air from escaping through it. The closure may be at any place of articulation from bilabial at the front of the oral cavity to uvular at the back. Amharic consonants classified under

this category are /ም/, /ን/, and /ኝ/. All consonants under this category are voiced and based on place of articulation, /ም/ is labial, /ን/ is alveolar, and /ኝ/ is palatal.

Liquids

Liquids are consonant sounds that are produced when the tip of the tongue closes the oral cavity leaving a side route for the air flow. Amharic consonants under this category include /ል/ and /ር/. Both sounds are alveolar and voiced.

Glides

Glides, also known as semivowels, are sounds with vowel and consonant features. Like vowels there is no major obstruction of air pressure emanating from the lung. Functioning as consonants they precede vowels that form the nucleus of syllables. In Amharic, the two semivowels are /ው/ and /ይ/. In view of place of articulation, /ው/ is labial while /ይ/ is palatal. But both /ው/ and /ይ/ are voiced in manner of articulation.

3.4 Syllable structure of Amharic words

Syllable structure, which is the combination of allowable segments and typical sound sequences, is language specific. As we have perceived from different literatures, the syllable type of a language can be defined in terms of underlying syllable templates. All the syllable type of Amharic can be also defined at the phonological representation. In Amharic language, there are six main syllable templates [54]. These templates accounts for all other possible syllable patterns. Moreover, the longest possible syllable is CVCC [54].The main syllable templates of Amharic language are: 1. V

2. VC

3. VCC

4. CV

5. CVC

6. CVCC

We can classify these templates into different classes based on the grammatical structure of the syllable templates. In a simple weight distinction, there are heavy and light syllables in Amharic. Heavy syllable is a syllable that either ends in a consonant or has a long vowel or diphthong. Light syllable is a syllable that ends in a short vowel. A closed syllable is one that ends in a consonant and an open syllable is one that ends in a short vowel or diphthong which was considered in this work. Thus we can restate the definition above: short vowel open syllables are light, all others are heavy. **Table 3.6** shows the summary of different kinds of Amharic syllable structure.

Table 3.6: Different kinds of Amharic Syllable templates

Kind	Description	Example
Heavy	Has a branching rhyme. All syllables with a branching nucleus (long vowels) are considered heavy. Some languages treat syllables with a short vowel (nucleus followed by a consonant (coda) as heavy.	CVCC, CVC
Light	Has a non-branching rhyme (short vowel). Some languages treat syllables with a short vowel (nucleus) followed by a consonant (coda) as light.	CV,VC, VCC
Closed	Ends with a consonant coda.	CVC, VC, CVCC, VCC
Open	Has no final consonant	CV, V

3.5 Stress and Syllables

There has never been agreement among linguistics on the topic of stress assignment in Amharic. Stress in Amharic words is complex [74]. However, there are some systems proposed in relation with stress and syllable structure. In many stress languages, stress is sensitive to a distinction called syllable weight. In a simple weight distinction, there are heavy and light syllables, defined as above.

Regarding the stress assignment rules of Amharic, we get the following rules form different literature. There are also other methods proposed by different scholars but the following rules have direct relation with syllables and syllable weight [74].

1. Stress falls on a heavy final syllable only in bisyllabic words when the first syllable is light.
2. Otherwise, the final syllable is skipped and the right most heavy syllable is stressed.
3. In the absence of any heavy syllables, the left most of a string syllables is stressed.

Although stress assignment is beyond the scope of this thesis work, once we have the syllables we can use rules of syllable weight assignment to assign stress based on the rules defined in relation with syllables and their corresponding weight. Therefore, having syllables and syllable weight for each syllable in the given word we can assign stress based on the rules specified.

CHAPTER FOUR

DESIGNING SPEECH SYNTHESIS SYSTEM

Speech synthesizers for different languages mainly differ because of the processing logic of the NLP component [1]. However, there is a difference as far as the NLP component is considered. Hence, the HMM based speech synthesis for Amharic is address the text, phonetic, and prosody analysis in line with the language's characteristics.

In the following sections, details of the process undertaken to design a TTS system for Amharic using hidden Markov models is discussed. The chapter also explains the whole system architecture starting with the selection of text and speech corpus. It then looks into the preparation of the files and data required for training the model. It further describes the software and tools required for experimentation and also discuss the step-by-step implementation phase of the research prototype.

4.1 Overall system architecture

As shows in **figure 4.1**, the synthesizer has two parts: The training and synthesis part. The training part includes data preparation, language processing, feature extraction, and building the HMM. Whereas, the synthesis part includes preparing labeled text from the text input, selecting appropriate HMMs, extracting speech parameters from HMMs, and finally generating the speech waveform from the speech parameters.

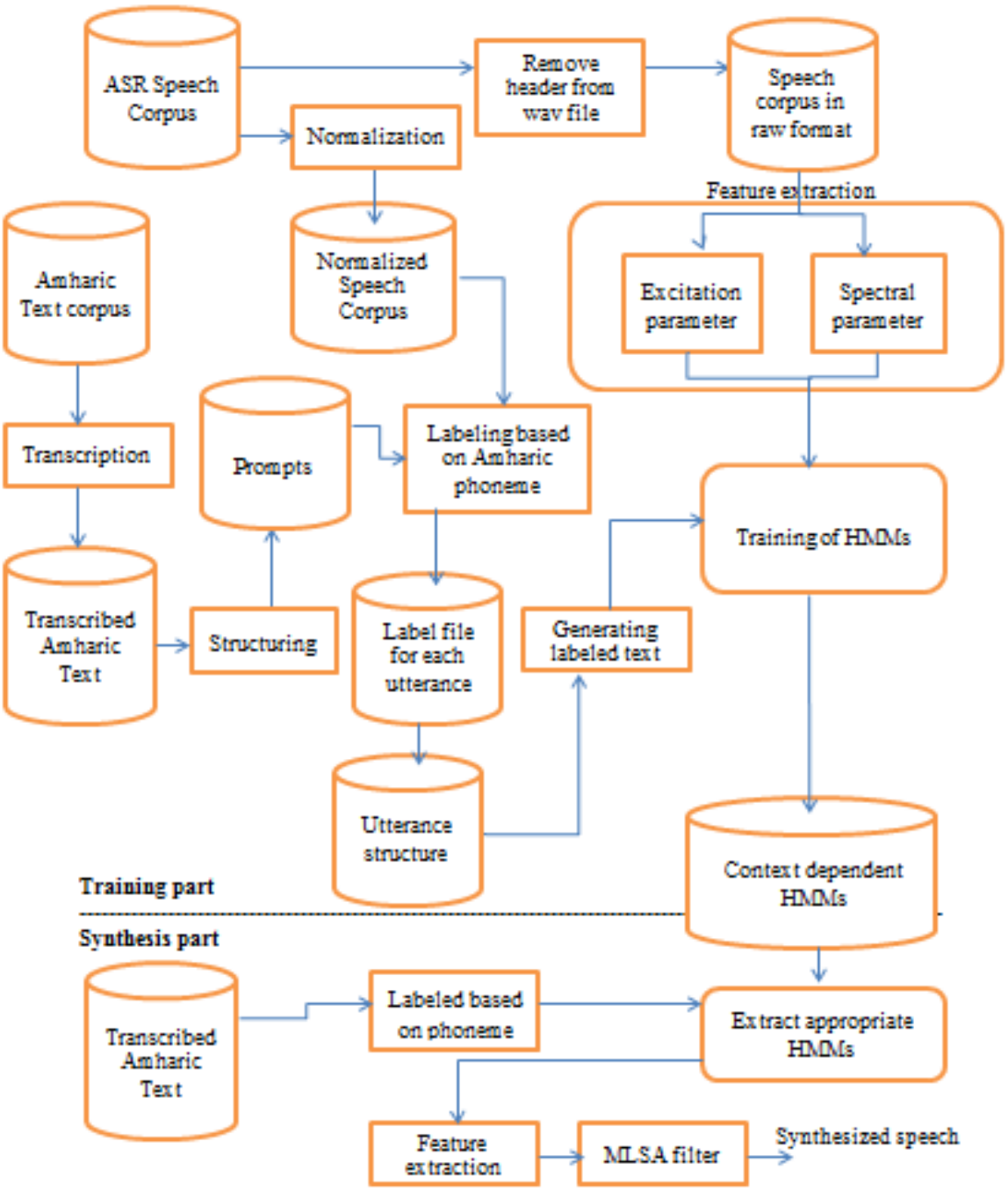

Figure 4.1: Overall system architecture

4.2 Description of the Architecture

This section describes the overall architecture and discusses the preparation details of the data and files necessary for successful experimentations conducted in this research prototype.

4.2.1 Data Collection and Preparation

The data selection method was as follow, a corpus of size 10,000+ ASR men and women speaker's speeches and corresponding sentences are used for selecting the training and testing dataset using random sampling technique. Many literature and scholars recommend that to train HMM model on average up to 2hr speech corpus was suggested. So as in this research, Out of 10,000+ ASR corpus 600 sentences and corresponding speech data by six female speakers is selected. These selected data had the following phonetic coverage. From 233 Amharic syllables 203 (87% of Amharic CV syllables) covered within selected 600 corpus and 32 (82% of Amharic phones) are covered from the total 39 Amharic phones as well. The frequency of the top 30 syllables and phones are as shown in **Appendix k**. From the total 600 corpus, 550 (total speech duration 1 and 1/2 hr.) were used to train the HMM model and 50 sentences is used for synthesis by labeling them for testing.

4.2.2 Normalization

The selected speech sample was recorded at 44.1 KHz stereo and the files stored in waveform format (.wav). The waveform files were normalized and changed to conform to 16 KHz, 32 bit, RIFF format as required by the festvox system and to make it easier to create raw files of small sizes. The command used to achieve this normalization is ./bin/get_wavs recording/*.wav. The recorded speech by six speakers – all are females – went through the .wav to .raw file conversion

process. The wave files were then converted to raw files by removing header using command $ESTDIR/bin/ch_wave -otype raw -F 16000 wav/$i.wav -o /project path/raw/$i.raw manually.

4.2.3 Transcription and Structuring

The sentences were collected from already available ASR corpus "Automatic speech recognition corpus for Amharic language" [48]. The collected sentences were then structuring suitable format for festival by transcribed and constructed using python scripts, then put into a festvox file called data.txt.done which is used for creating prompts. An example of the file data.txt.done can be found in **Appendix D**, and the file would be copied to the path aau_amh_bah/etc.

4.2.4 Labels and Utterances

From the sentences listed in data.txt.done, festvox generates prompts using the festvox command ./bin/do_build build_prompts. This command returns among other things the initial set of labels and utterances in the directories /prompt-lab and /prompt-utt respectively. This initial set of labels and utterances is not well aligned/transcribed and the labels are not context dependent. More accurate labels and utterances are then generated using the festvox commands ./bin/do_build label and ./bin/do_build build_utts respectively run consecutively. The newly generated labels will be used to generate utterances which will, in turn, generate context-dependent labels. These labels are automatically aligned/transcribed using an EHMM labeler and are found on the path aau_amh_bah/lab. The utterances are also generated automatically based on the label files in aau_amh_bah/lab and are themselves found on the path aau_amh_bah /festival/utts. There are several algorithms that can be used for automatic labelling in addition to EHMM labeler. An alternative to using automatic labelling can be to hand-label the speech corpus. Hand labelling is a very tiresome task and it requires expertise. For this research demo,

festvox's EHMM labeller was considered sufficient for its automatic labelling. Hand labelling can produce even better results especially when applied to improve on an already automatically labelled label. Hand labelling was not used for this demo due to time constraints and its need for special skilled human resources.

It is important to note that the word utterance is used in two different contexts in this document. The first is an utterance (.utt) file from which labels can be derived. The second refers to a speech interpretation synonymous to a person uttering a word or a machine rendering (uttering) a speech waveform.

- **Question set and phone set radio**

The properties of both consonants and vowels are significant to the design of both the phone set radio and the question set files. Both these files capture almost the same information, but the difference lies in the representation of the information. It is in these files that the classifications of vowels and consonants such as voicing, place of articulation, manner of articulation, lip rounding, etc., are captured.

The Festival system installation includes a file named radio_phones.scm in its library (lib) folder. This file needs to be changed to conform to the rules of the language being used in order to correctly train and synthesize the text entered for synthesis. This file is required to use the same language units (phonemes – vowels and consonants) as used in the question set file. The phone set radio (radio_phones.scm) was also changed to conform to Amharic language and a portion of this file is included in **Appendix E**. This file captures the details of the entire phoneme set using predefined special characters, e.g, +, -, 0, 1, 2, 3, s, l, d, etc., to represent certain properties.

A question set required for the HTS-demo was created for Amharic language using the English HTS-demo format for creating the question set. An example of the question set is included in **Appendix D.** The question set is found on the HTS-demo under the path /data/questions and the file is named questions_qst001.hed. The question set uses a method of grouping phonemes with the same properties together. The question set was created based on the language structure for Amharic.

- **Question file**

Some sort of criteria is required to tackle the problem of fewer training examples available per model. The numbers of training examples are few because if we look at the full-context style label format, then it reveals that the possible contextual occurrence of a single phoneme is quite huge. And to have this much context available in the training data is not possible, on the other side this much context is also rare in everyday speech.

To address these issues, a methodology known as clustering is employed. The notion of clustering is to group the phonemes which are acoustically similar and share a single model for closely related contexts. In clustering question file plays an important role, as they define how the grouping should be done.

The question file consists of a number of binary questions with YES or NO outcome related to segmental and prosodic context of the phoneme. It is helpful in the clustering process for spectrum, F0 and duration models[76]. It provides a basis for grouping a number of data points and hence handles data sparsity issues, which is common in speech synthesis as the numbers of unique models built are enormous.

In clustering first all the data points are placed into a single cluster and a list of questions is made. Then an objective function is defined. The cluster is split based on each question, and the question with which the objective function is minimized is selected as a successful candidate and is removed from the list. And the remaining questions are asked on the resultant clusters until a stopping criterion is met [77]. Another advantage is that these trees can also be used in the synthesis stage, as they are built on the acoustically similar properties of speech. In synthesis mostly the utterance that is required to be synthesized is unseen. Meaning that utterance was not available exactly the same way in the training data. So, by using the trees that incorporate the acoustic similarities we can trace them and select the closest alternative.

- **Generation of the Question file**

The questions are developed based on the similarities between the place and manner of articulation for segmental context. For prosodic context, the number of syllables in a word, their position and whether stressed or not etc. is taken into account. The idea is to group the phones that have a similar place and manner of articulation. These set of questions have a dual role. In the training process they are used to split the cluster nodes of the tree. Whereas during synthesis process, these generated trees are employed to trace the phoneme with un-seen context. Moreover these questions are created on a phoneset specific to language; we cannot employ the structuring specified for some other language (like English, Germen and Japanese).

- **Question format**

For example the question for phoneme before the previous to previous phoneme is defined as:

Field 1	**Field 2**	**Field 3**
QS	“LL-CONSONANT”	{A^*,B^*}

Table 4.1: question set format

The field 1 defines the label showing that it is a question. Field 2 specifies the grouping of various categories and finally the third column represents the possible phonemes for the defined categories.

- **Phonetic Analysis**

The phonetic analysis module takes the normalized word strings from the text processing module and produces a pronunciation for each word. The pronunciation is provided not just as a list of phones, but also a syllabic structure and lexical stress. The method for finding the pronunciation of a word is by a lexicon or/and by letter to sound rules. For developing Amharic synthesizer demo, the researcher used both lexicons and hand written letter to sound rule sets. Hand written letter to sound rules are context dependent re-write rules which are applied in sequence mapping strings of letters to strings of phones.

The Festival framework provides a lexicon subsystem upon which lexical analysis could be performed for the language. Using this framework, a compiled lexicon is prepared using the words intended for building the synthesizer prototype (training set). Festival also provides an alternative way of representing words pronunciation called the Adenda. But, since the Adenda is searched linearly when a new instance is encountered, it is less efficient than the compiled lexicon which follows a binary search mechanism.

In addition to the lexicon entries prepared explicitly for the system, a letter to sound rule is implemented as a mechanism to handle words whose pronunciation is not given in the lexicon. The compiled lexicon provides the pronunciation of the words in terms of the phonemes of the language prepared as a phoneset. Every entry in the compiled lexicon maintains three parts, the

root word to be pronounced, part of speech it belongs to and the actual pronunciation of the word. For example, in the following lexicon:

(lex.add.entry '("newix" nil (((ne)0)((wix)0))))

(lex.add.entry '("iitixyopxixya" nil (((ii)0)((tix)0)((yo)0)((pxix)0)((ya)0))))

(lex.add.entry '("ixnixde" nil (((ix)0)((nix)0)((de)0))))

the word ‘iitixyopxixya’ has the entry, (“iitixyopxixya” nil (((ii)0) ((tix)0) ((yo)0) ((pxix)0) ((ya)0)))) where “iitixyopxixya” indicates the root word, ‘nil’ represents the part of the speech which in our case is not considered followed by the pronunciation identifying syllable structure and stress markings. The lexicon look up process first checks the compiled Amharic language lexicon; if there is a full match (exactly matching the head word) the corresponding pronunciation is returned. If there is no match found in the compiled lexicon, the word is passed to the letter-to-sound rule that tries to map it into respective pronunciation.

In this research use and prepare pronunciation dictionary (lexicon) for standard words for all unique words in the **600** sentences i.e. **3430** words. However, the mapping process of every letter in the language to a set of possible sounds is challenging as it demands large corpus so that all the phonemes are classified in one of the letter to sound rule entries.

4.2.5 Feature Extraction

At this stage features are extracted, as discussed in chapter two, two parameters, the spectrum and excitation, are needed to train the model. The spectrum parameters consist of mel-cepstrum or linear prediction coefficient. On the other hand, the excitation parameter consists of the

fundamental frequency (F0). In this thesis work mel-cepstrum coefficients are used as spectrum parameters.

Using the raw data generated during data preparation, these speech parameters (features) are extracted using the tool SPTK-3.9 and then the delta and delta-delta values of the mel cepstrum coefficients are calculated. Similarly, delta and delta-delta values of the logarithm of F0 are calculated.

4.2.6 Training the Model

Training the model means estimating the HMMs parameters, which are the mean, the variance, and the transition probabilities based on the utterance structure and the extracted parameters (features). Once the parameters are extracted, training of HMMs is performed with the Hidden Markov Model Toolkit, which is software that provides a set of library modules and tools for building and manipulating HMMs. HTK supplies four basic tools for HMM parameter estimation: HCompV, HInit, HRest and HERest [45]. In addition to these tools one additional HTK tool is used for context analysis, which is HHEd.

HCompV and HInit are used for initialization of the model parameters. HCompV is used to set the mean and variance of every Gaussian component in a HMM definition. This tool is typically used for initializing the model if the speech utterance is not labeled. Alternatively, a more detailed initialization is possible using HInit which computes the parameters of a new HMM. In this research work HInit is used to initialize the model since the speech utterance is labeled. HRest and HERest are used to refine the parameters of existing HMMs using Baum Welch Re-estimation (forward/backward) algorithm [45].

The defined HMM for this thesis work incorporates both mel cepstrum coefficients and F0. Therefore, to model a segment of speech signal a pair of mel cepstrum and F0 values together with their delta and delta-delta values are needed. Hence, the mel-cepstrum and F0 values are combined together, so that they will be used to model the speech segment.

The first step in generating HMM parameters is to provide initial estimates for the parameters of every HMM model of the phoneme set. This process takes as input HMM definition, combined parameters, and label for each phoneme and initializes them one by one.

Initializing the HMM parameters requires some data (combined parameters). To circumvent this problem, the speech signal of each phoneme will be uniformly segmented and each segment will be used to initialize the parameters of each state of the model. Such initialization only makes sense if the HMM is left-to-right. In this work HInit is used for model initialization process.

The initialized model and the combined parameters selected from all utterances are used to refine the model parameters for a given phoneme using forward/ backward algorithm. The labeled utterance is used to select the phoneme boundary from all utterances and to collect the parameters that only belong to a phoneme. For refining the model parameters of each phoneme HRest tool is used.

The output of the model-parameters refining process will be combined together to have a master macro file (MMF) and used as input for the next process together with master label file. Master label file (MLFs) is prepared by combining all the transcriptions together. Both the previous processes, model initialization and refining model parameters, are used to reestimate the parameters of the model for each isolated phoneme separately. However, modeling a phoneme without considering the contextual factor will not give a good model of the utterance. Therefore,

model parameters should be estimated without considering the phoneme boundaries so as to include contextual factors. That is, unlike the processes described so far, this process simultaneously updates all of the HMMs of each phoneme in a given utterance. For this process HERest tool is used.

HERest processes each training file in turn and the whole utterance is considered rather than a single phoneme. It uses the associated transcription to construct a composite HMM which spans the whole utterance. This composite HMM is made by concatenating instances of the phoneme HMMs corresponding to each label in the transcription. When all of the training files have been processed, the new parameter estimates are formed and the updated HMM set is generated as an output.

The contextual factors that are considered in this thesis are phoneme identity factor, stress related factor, and location factors. Phoneme identity factor checks whether a given phone is vowel or consonant. Stress related factor is a factor that determines whether a given phoneme is stressed or not. Location factors deals with the location (position) of a given phoneme in a given word.

Considering the above contextual factor obviously enables to obtain appropriate models. However, as contextual factors increase, their combination also increases exponentially, which increases the searching and computation cost. To overcome this problem there are two methods: Data driven context clustering and Decision tree based context clustering. In this thesis work decision tree based context clustering is used by using the HTK tool HEEd. To do so, a question set for Amharic Language is prepared, which is needed to build the decision tree. As it is shown in **Appendix D**, a question set is a file that consists of a set of questions. Considering the first line of the question set,

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

It reads as question set “Is the left of left phoneme vowel?” and the answer will be given based on whether the phoneme is in the list or not. That is, during building a decision tree a question is raised, for example, “Is the phoneme to the left of left vowel?” To answer this question, the question set is checked and if the vowel to the left of left of the phoneme is in set of “QS L-vowel”, that is in QS "LL-Vowel" {a^*,e^*,ii^*,ix^*,ie^*,o^*,u^*}, then “yes” will be the answer. By doing so all contextual factors for given phoneme is considered. The question set given in **Appendix D** is only for left of left phonemes to the given phoneme. However, five positions are considered in this thesis. These are, LL (left of the left phoneme 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).

4.2.7 Synthesis Phase

As it is shown from the overall design, the synthesis phase includes activities like NLP, selecting appropriate models for the input text, speech parameter generation from the models and generating the synthesized speech from the generated parameters.

In the synthesis stage first the text to be synthesized is entered in desired format, then using the utilities developed in the training stage, converted to full-context style labels. The label format used in the synthesis part is similar to the training, except for the timing information which is absent it this case. By using these labels three different set of models are selected (Spectrum, Fundamental frequency and the Durations). From Spectrum and Duration models the optimal state sequence is selected. Finally the optimal state sequence along with the excitation signal is fed to the synthesis filter to produce the final waveform, as illustrated in **Figure 4.2**.

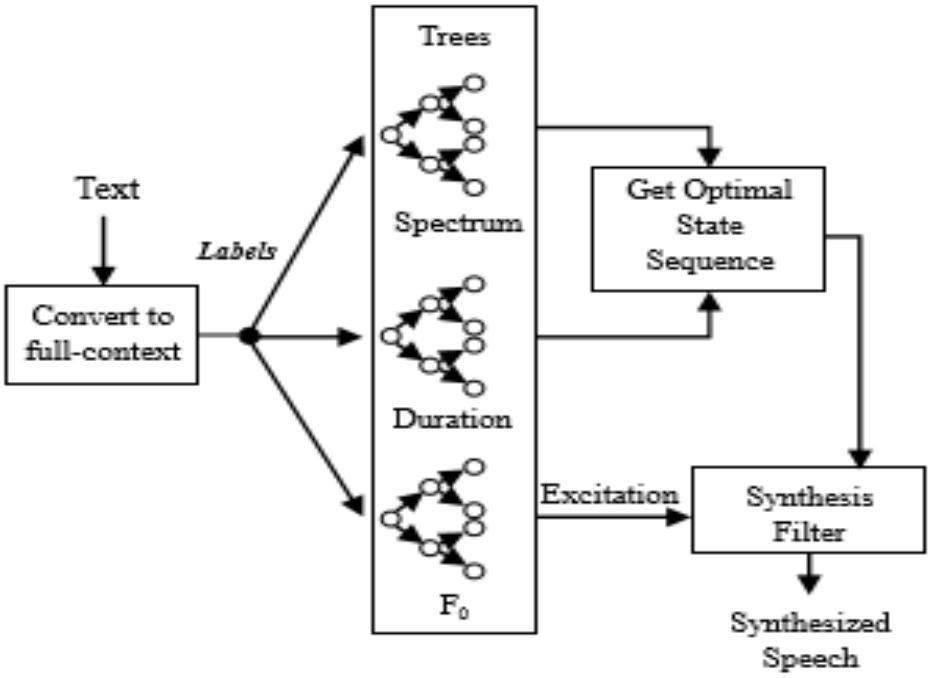

Figure 4.2: Overview of Synthesis Process

The texts that are going to be synthesized will not be given directly to the synthesizer. Rather the text will undergo through different pre-processing activities. However, a program is written using python to generate the labeled text; and then appropriate models were selected corresponding to each phonemes of the labeled text. From the selected models speech parameters, spectral and excitation parameters were generated. To select the appropriate models and to extract speech parameters from the models, HMGenS was used, a speech parameter generating tool, which is provided by HTS-2.3alpha open source tool kit or in our case hts_engine used. Finally, speech waveform is synthesized directly from the generated parameters by mel log spectral approximation filter using SPTK tools.

4.3 Software Packages and implementation

This section lists the software packages required for installation in order to set up the working environment for the research prototype. A brief descriptive discussion of the software packages is also given.

4.3.1 Software package listing

A few software installations were required in order to create a working environment for launching the present research demo. The experimental environment was set up on a normal laptop computer with 8GB RAM, Intel(R) _Core(TM) _i7-6500U_CPU_@_2.50GHz. The computer was running on windows 10 but the demo was running using virtual machine platform on OpenSuse 13.1 operating system. Several software packages necessary to the success of this research demo were acquired and/or downloaded and installed. Appendix G details the unpacking and installation of these software systems. The support for most of the packages can be found at their respective sites' mailing lists as one might face some challenges in using some of the software. For the majority of the software packages, even their recent versions would work well in the design of a similar project. The software components listed below were downloaded for installation as they were significant to the success of the research demo and they can be downloaded from their respective sites.

Key software packages:

- speech_tools-2.4-release.tar.gz [1]
- festival-2.4.tar.gz[1]
- festvox- 2.7.0-release.tar.gz[1]
- HTK-3.4.1.tar.gz [2]
- Hdecode-3.4.1.tar.gz[2]
- HTS-2.3alpha_for HTK-3.4.1.tar.gz [3]
- SPTK-3.9.tar.gz [4]
- hts_engine_API-1.10.tar.gz [5]
- ActiveTcl8.4.19.4.292682-linux-ix86.tar.gz [6]
- HTS-demo_CMU-ARCTIC-SLT.tar.bz2 [7]
- festlex_CMU.tar.gz[8]
- festlex_POSLEX.tar.gz[8]
- estvox_kallpc16k.tar.gz[8]
- festvox_cmu_us_awb_arctic_hts.tar.tar[8]
- sox-14.3.2.tar.gz [9]

Extra software packages:

- Praat
- Wavesurfer

http://festvox.org/index.html[1], http://htk.eng.cam.ac.uk/[2] , http://hts.sp.nitech.ac.jp/[3], http://sp-tk.sourceforge.net/[4], http://hts-engine.sourceforge.net/[5], https://www.activestate.com/activetcl/[7] , http://hts.sp.nitech.ac.jp/archives/[7], http://festvox.org/[8], https://sourceforge.net/projects/sox/files/sox/14.3.2/[9],

4.3.2 Software package description

Speech tools – A library of C++ functions for the speech processing of related speech objects. It is free software developed and maintained at the University of Edinburgh's Centre for Speech Technology Research. It is used for reading, writing, converting and supporting speech processing objects such as fundamental frequency, waveform, labels, etc..

Festival – It is a baseline system that provides a platform for developing TTS systems. It is a multilingual TTS system which is also a package that allows for the development of new systems with ease. Similar to speech tools, festival is free software with a licence that allows for unrestricted usage. Two lexicons (festlex_CMU and festlex_POSLEX) and voices (festvox_kallpc16k and festvox_cmu_us_awb_arctic_hts) were included in order to allow festival to speak. There are, however, many voices and lexicons available for download that can be used with festival.

HTK – It is a toolkit that was primarily designed for use in speech recognition research for building and manipulating hidden Markov models. HTK was originally developed at Cambridge University and can be used in many different applications including speech synthesis. HTK can be downloaded for free, but it requires one to first register with a valid email address and agree to its licence.

HDecode – HDecode is a package that is an add-on to HTK; it requires one to have registered as an HTK user, and also agree to its licence in order to download it.

HTS – HTS is an HMM-based Speech Synthesis System toolkit that is designed to be patched to HTK HTS is not a standalone software system, as a result it is required that once patched to HTK one should also agree to and obey the HTK licence.

SPTK – It is a software package that comprises speech signal processing tools. It is freely downloadable and is released under the Modified BSD licence.

hts_engine – It is software that is used to synthesize speech waveform from HMMs output by HTS. It is freely downloadable and is released under the Modified BSD licence.

ActiveTcl – TCL stands for Tool Command Language. It is a portable interpreter which is freely downloadable.

Sox – It is a command line utility that is famous for the conversion of different audio file formats. It can also be used to record, play and manipulate audio files. Sox is a free software.

HTS-demo – It is software used to train and synthesize a TTS system using HTS. Most of the software listed and discussed in this chapter are meant for use by this package. It is also freely downloadable.

Festvox – It is a project that is based on festival and is aimed at making the task of building a new voice easy. All the software components that it uses are free and unrestricted.

Praat – It is free software for sound manipulation, phonetic and acoustic analysis.

Wavesurfer – It is a toolkit used for sound visualisation and manipulation. It is very convenient in displaying waveforms, transcriptions, etc. It can also be used to convert speech files from one form to another. It is open source software.

4.4 Experiment Setup and implementation

This section takes one through the process taken to model TTS synthesis system by assuming that the installations, as outlined in **Appendix G**, of the packages discussed in the previous section have already been done. A detailed approach ranging from the actual data preparation to the commands used to run the demo is given.

4.4.1 Initial stages

Now that all the necessary software packages have been installed, a step-by-step narrative approach is taken into the development of the TTS synthesis system based on hidden Markov models. Firstly, a set of 600 sentences in Amharic was compiled. These sentences were collected mainly from the Amharic ASR corpus data and most of sentences were the general domain looking into some of the frequently used news sentences. This list was divided into two sets making up the training and testing sets. The initial set would be used later in the construction of a prompt list file called txt.done.data.

A phone set radio file was then created according to the language structure of Amharic. This file was named radio_phones.scm and copied to the path /festival/lib where it overwrote its counterpart. It could be a good idea to first rename the file radio_phones.scm under festival's lib directory before copying the newly created file therein. The old phone set radio file could also offer some light on how to create a phone set radio file for a different language.

4.4.2 Festvox

Festvox is actually based on festival. One can look at festvox as festival made easy in the creation of new voices for any natural language. All the processes defined in this section are documented in a manual called “Building Synthetic Voices”, by [75].

Every time one uses festvox, it is a requirement to first set the two environmental variables FESTVOXDIR and ESTDIR. These variables point to the paths where festvox and speech tools are located. In our case the following paths were used to initialize the variables.

```
export ESTDIR=/home/bahiru/HTSprototype/speech_tools

export FESTVOXDIR=/home/bahiru/HTSprototype/festvox
```

In order to start a new project, a new folder was created. The created folder is required to follow a particular standard, whereby, the folder name should start with the name of the institution, followed by the language, and concluded by the surname and name of the subject/owner in abbreviations. In this case the name of the institution is Addis Ababa University (aau), the language is Amharic (amh), and the surname and first name of the primary subject are in this case bahiru (bah). The newly created folder was then navigated into.

The following commands perform two operations which are to create a new folder and navigate into it respectively.

```
mkdir aau_amh_bah

cd aau_amh_bah
```

A template or skeleton of directories and files were then created. The template files and directories were automatically created by running the command.

```
$FESTVOXDIR/src/clustergen/setup_cg aau amh bah
```

After creating the template files, it is important to then transform them to suit the present project's specific and special needs. The collected list of sentences was used, in order to create a

file called txt.done.data stored under /aau_amh_bah/etc. A snapshot of the file can be found in Appendix F. An example of the structure used in creating the file is given by the following:

(bd_v_bah_001 " ")

(bd_v_bah_002 " ")

The above structure starts with an opening bracket, followed by the festvox demo dataset name, the speaker and the subject name with number separated by an underscore (_), the sentence is included as a string delimited by double quotes at the beginning and end, and concluded by a closing bracket. Another file called aau_amh_bah_lexicon.scm found under / aau_amh_bah / festvox was created. Other files which were modified include aau_amh_bah_phoneset.scm found under the same folder as the file aau_amh_bah_lexicon.scm. A grapheme-based rule was not defined for this demo and the system depended solely on A pronunciation dictionary which was created in aau_amh_tts_lexicon.scm.

The recorded files were then placed under the folder called /Recordings which is under /aau_amh_bah. The recorded wave files were then normalized and down sampled to conform to a 16 KHz, 16 bit, Resource Interchange File Format (RIFF) format. The resultant wave files are then stored on the path /aau_amh_bah/wav. The command used to achieve this is given below.

```
./bin/get_wavs recording/*.wav
```

Next, the prompt files were created, whose output was saved in the folders starting with the keyword prompt under /aau_amh_bah/etc. This command used the prompts data; the sentence list (txt.done.data), aau_amh_bah_phoneset.scm and the phone set radio and other files to create the prompts. The initial label and utterance files are a result of the following command:

```
./bin/do_build build_prompt
```

In order to have context-dependent labels, another command was executed; this command gave the final set of labels which would then be used to create utterances from which the desired labels would be obtained. These labels are automatically generated using the EHMM labeler. The resulting labels are stored on the path /aau_amh_bah/lab. These labels are used to create utterances by using a different command. The commands used for creating labels and utterances are given below in their sequence.

```
./bin/do_build label

./bin/do_build build_utts
```

The utterance files just created are in the format that can be used by the HTS-demo and those utterances can be found on the path /aau_amh_bah/festival/utts. The HTSdemo also requires raw (.raw) files. The raw files that are generated by using the newly generated wav files in /wav of /aau_amh_bah are much smaller in size as compared to those generated from the wav files in /recording of /aau_amh_bah. The raw files generated from the /wav directory worked fine on HTS, whereas the ones generated from the /recording directory gave problems with forced alignment due to their bigger size. Conversely, HTS seemed to prefer raw files generated from the wav files in /recording and returned an error/warning when using those generated using wav files in /wav. The reason for this is, used the sampling rate of 16 kilohertz (KHz), whereas HTS is defaulted to use the sampling rate of 48 KHz. The preliminary results used in the research, were based on raw files obtained from the 16 bit, 16 KHz, RIFF wav files which were down sampled from the 44.1KHz recordings using festvox‟s command (bin/get_wavs recording/*.wav).

4.5 HTS-demo

With most of the files already prepared the HTS-demo was unpacked using the tar command. The question set was then created, as described in generation of question file, as well as depicted in Appendix D. The question set named questions_qst001.hed was replaced with the question set relevant for the present project and having a name similar to the one of the deleted file, on the path /data/questions of the HTS-demo. All files on the paths /data/labels/gen, /data/utts, and /data/raw were deleted. Test labels, utterance files, and raw files specific/relevant to the present prototype were then copied into their respective emptied folders. The path /data/labels/gen contain context-dependent labels corresponding to the test sentences without time stamps.

Now that all the necessary files were in the right places, it was time to start running the HTS-demo. The first stage in running the HTS-demo is to configure it. Configuring the HTS-demo refers to setting the paths to the packages required to run it. The INSTALL file on the HTS-demo gives a guideline on how to set up these paths and alter certain parameters. It also indicates the packages required to run the HTSdemo and gives the universal resource locators (URLs) for their download. The configure statement used for this project is given as follows:

```
bahiru@linux-0bbs:~/HTSprototype/HTS-demo_CMU-ARCTIC-SLT>./configure --with-tcl-search-path=/home/bahiru/HTSprototype/ActiveTcl-8.4/bin \
> --with-fest-search-path=/home/bahiru/HTSprototype/festival/bin \  or
> --with-fest-search-path=/home/bahiru/HTSprototype/festival/examples \
> --with-sptk-search-path=/home/bahiru/HTSprototype/SPTK-3.9/bin \
> --with-hts-search-path=/home/bahiru/HTSprtotype/hts_patch/htk/bin \
```

```
> --with-hts-engine-search-path=/home/bahiru/HTSprototype/hts_engine_API-1.10/bin \
```

There are certain files which needed to be modified in order to run the demo successfully. In order to be able run the HTS-demo using data specific to the present study, values of certain variables had to be changed in the files makefile and config.pm found in the directories /data and /scripts respectively, which are on the main HTS-demo directory.

makefile # setting

SPEAKER = v , DATASET = bd

After all the initialization preparations were done, it was then time to run the HTSdemo. Using the bash terminal the main HTS-demo directory was navigated into by using cd command and the HTS-demo was run by using the make command. The commands used to achieve these two operations are:

```
cd HTS-demo_CMU-ARCTIC-SLT

make
```

At this stage features are extracted, as discussed in chapter two, two parameters, the spectrum and excitation, are needed to train the model. When the training process completes, the hts_engine synthesized the waveform of the test labels (sentences) which were on the path /data/labels/gen. The synthesized wave files are found on the path /HTS-demo_CMU-ARCTICSLT/gen/qst001/ver1/hts_engine. The demo usually takes several hours to days for successful completion on a normal laptop computer. After the make process is finished, then the training of HTS-demo run by using the training and synthesis perl script (training.pl and config.pm).

CHAPTER FIVE

THE EXPERIMENTAL RESULTS AND EVALUATION

This chapter discusses the methods used for evaluating the developed TTS synthesis prototype. It also discusses the criteria used to select test sentences and evaluation subjects. The actual evaluation of the results together with their analysis is then performed.

5.1 Testing Procedure

The most popular and effective way of evaluating TTS synthesis systems is through listening tests. This is significant in that potential end-users are part of the evaluation process. Moreover, the popularity of such systems is depends on whether ordinary users find them easy and pleasant to use. In TTS systems research/design, there are certain factors that system designers might consider important focus areas, whereas general users factors as insignificant; hence, the need to have listening tests.

Listening tests can be done through full sentence ratings. Furthermore, evaluators (or human test subjects) can match an utterance to a transcription or transcribe an utterance, in which case they already know what to expect. Evaluators then rate the speech signal produced based on the given categories (e.g., naturalness, intelligibility, etc.) on a scale of 1 (bad) to 5 (excellent). The average of the resulting response values is calculated. The computed average is referred to as the mean opinion score (MOS).

For this research the evaluation process employed is of two types. First, the overall performance of the prototype is measured in terms of total number of correctly pronounced words over each sentences played. In the second experiment we have evaluated intelligibility and naturalness of the synthesizer using MOS.

5.2 Evaluation Criteria

The enrollment process for evaluators or test subjects was structured as follows:

- Fluent Amharic speakers (not necessarily mother tongue speakers).

- Both male and female test subjects ranging between the ages 18 and 35.

- The number of test subjects was eight: 4 – female and 4 – male.

- People from: undergraduate and postgraduate students, sanitary personnel, security guards, etc.

Test sentence selection criteria:

- Sentences that do not form part of training set.

- The number of test sentences was fixed to 45 because the rest 5 for MOS test.

For this experiment, all evaluators were required to fill a consent form requesting their biographic information. Evaluators were made aware of the special nature of synthetic speech produced by the TTS systems so that they could appropriately adjust their expectations regarding the output from the system. They then listened to utterances and mapped them to their corresponding transcriptions. This was done to ensure that they were able to fairly judge the system as they would know what output to expect. Moreover, this also allows evaluators to comment on words not properly or wrongly synthesized.

The evaluation process is mainly based on the two concepts – naturalness, intelligibility, Naturalness has to do with the human-like sounding of the system, whereas intelligibility/ understandability has to do with the ability for one to hear the speech synthesized. Pleasantness – which is how enjoyable it is to listen to the synthesized speech is also evaluated. Lastly, the overall system performance impression is determined.

5.3 Evaluation Results and Analysis

The TTS synthesis system was trained using 550 ASR corpus recorded by Solomon [48] for Amharic automatic speech recognition system. Fifty sentences which did not form part of the training data. These forty five were used for the preference test of the TTS synthesis systems. The other five test sentences also were selected to test the TTS synthesis system based on the MOS method by the evaluators. Evaluators from diverse backgrounds and languages were used to test the system. Eight people, 4 males and 4 females, tested the system. The people that evaluated the system either spoke Amharic as their home language or could fairly hear and read the language.

The researcher listen one sentence played from the demo at a time and marks on the answer sheet for the synthesized sentence which is correctly pronounced or not. **Table 5.1** shows the analysis made on the performance evaluation of the test dataset.

Performance Measure on the Test Dataset						
	Correctly Pronounced		**Partly Pronounced**		**Bad Pronounced**	
	syllables	**phones**	**syllables**	**phones**	**syllables**	**phones**
No. of Sentences	**34**	**35**	**8**	**7**	**3**	**3**
% of Sentences	**75.56**	**77.78**	**17.78**	**15.56**	**6.67**	**6.67**

Table 5.1: Performance Measure of Amharic prototype

The overall performance of the prototype is measured in terms of total number of correctly pronounced words over the total number of words in each sentences played. Finally by calculating the number of sentences which are correctly pronounced the overall performance of the system is found to be 75.56% for syllable based and 77.78% for phone based system.

As observed from the analysis phonemes that are not found in the compiled lexicon or less frequent in the training data are not properly pronounced by the system. As per the sequence in the lookup process of Festival, if the pronunciation is not found in the compiled lexicon, the letter to sound rule is evoked to pronounce but not defined in the demo.

Basically, the letter to sound rule has entries that match each letters in the language alphabet to some possible sound forms that the letter can assume. However in this work, the letter to sound rule is not constructed. From the chosen 45 sentences, we observed that from results which improper pronunciation of the sentences, the phonemes contained in sentences not in the compiled lexicon. This outlines the main adverse effect in degraded performance of the system.

The MOS scores that used in the evaluation process ranged from one (bad) to five (excellent). The evaluation categories used for the MOS method were: understandability, naturalness, and overall system impression.

Value	**MOS**
5	Excellent
4	Very Good
3	Good
2	Fair
1	Bad

Table 5.2: Scales used in MOS

To evaluate the performance of the systems, the evaluators are invited. Then the evaluators provide their ranks based on the MOS scale. In both cases a questionnaire is used to collect the

evaluator's opinion. The questionnaire is attached in **Appendix I**. The results of the evaluation are presented in according to the format attached in **Appendix H**

The average intelligibility and naturalness of the synthesized speech for each sentence is presented in **Table 5.3** and **Table 5.4** for phone and syllable based respectively.

Sentence	**Intelligibility**	**Naturalness**
1	3.13	2.88
2	3	2.75
3	3.38	3
4	2.88	2.63
5	3	2.75
Average Scores	3.08	2.8

Table 5.3: Average MOS Scores of phone based Amharic speech demo

Sentence	**Intelligibility**	**Naturalness**
1	3	3
2	2.75	3.13
3	3.13	3.38
4	2.88	3.13
5	2.75	3
Average Scores	2.9	3.13

Table 5.4: Average MOS Scores of syllable based Amharic speech demo

The average MOS evaluation of the system from eight listeners for the five Amharic sentences is found to be 3.02 and 2.94 for phone based and syllable based respectively. Which means the synthesizer is ‘good’ for phone based and near to ‘good’ for syllable based as per the scale of the MOS test. The overall naturalness of the synthesizer found to be 3.13 and 2.8 for syllable based and phone based respectively which also ‘good’ MOS scale for syllable based and near to good for phone based. These values of intelligibility and naturalness look encouraging to come up with a better system. But the naturalness of the syllable based system somehow outperforms the phone.

The evaluations revealed that 37.5% of the people said that the system sounded like a human voice, 50% somehow sounded like human and the worst case being from one (12.5%) person saying that the system was not natural sounding. The evaluation revealed that 62.5% of the subjects found the system very pleasant to listen to and 37.5% of the subjects found the system horrible to listen to.

The evaluation results indicate that the overall system was found to be good on average. Only 25% of the subjects said that the overall system was excellent, 37.5% of the respondents said the system was good, and the other 37.5% said that the system was acceptable. The overall system, therefore, received a 100% acceptability rate.

Speech quality remains a concern. Several evaluators commented on the quality of the synthesized speech. A few others, even highlighted that voice clarity should be improved as they sometimes had to listen carefully to the synthesized speech in order to clearly hear the waveform rendition. Speech quality and clarity are very important to TTS synthesis systems and it should be what every TTS synthesis system designer strives for at the very least.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Speech Synthesis systems have been developed gradually over the last few decades and it has been integrated into several new applications. For most applications, the intelligibility and comprehensibility of TTS System have reached the acceptable level. Nevertheless, there is still much work and improvements to be done in prosodic, text preprocessing, and pronunciation fields to achieve more natural sounding speech.

In this thesis work, ASR corpus is used to develop a syllable based speech synthesis system for Amharic language using Hidden Markov Model. To come up with the new synthesis system, the nature of dataset required for speech synthesis and techniques was studied. Then, the datasets are randomly selected from ASR corpus with six female speakers' corpora as training data. Both text and speech with the size of each 600 were used. These corpus were split in to two, these are 550 (90%) for training and the rest 50 (10%) for testing data sets. And after, the researcher chose appropriate methods to handle such corpus. Lastly, statistical parametric approach (Hidden Markov Model) was selected. Every component of HMM based speech synthesis system was studied to identify those components that are dependent on the characteristics of a language. Having those components in mind, the Amharic language was studied. However, every feature of the Amharic language was not considered since it needs a lot of time and deep linguistic knowledge. Hence, this study considered only the characteristics and way of creation of Amharic phonemes.

The utterance structure generated by festival and festvox together with the parameters extracted from the raw wave data were used for training the model. Acoustic parameters, such as F0 and spectrum, as well as their dynamic characteristics are concerned in HMM model training. These models are then clustered by decision tree following MDL criterion. In synthesis stage, context information is generated by the text analysis procedure. Using context information, the system predicts HMM sequence by the decision tree. Finally, the speech parameter sequence, which is generated based on the predicted models, is used to synthesis the speech waveform by a vocoder. Basically, the text that is going to be synthesized was assumed to be transcribed. It means all the pre-process activities are done before it is given to the synthesis system. So, the synthesized speech is generated from the trained model based on the labeled input text.

Evaluation is done to see the overall performance of the demo and to measure how intelligible and natural synthetic speech is, based on a method called mean opinion score (MOS). According to the researcher evaluation, the systems register on the overall performance 75.56% for syllable based and 77.78% for phone based system; Preference evaluation result shows that Syllable based synthesis performs better in naturalness while Phone based TTS performs better in intelligibility with 550 sentences' training data. The MOS score shows, syllable based system 2.9 score in intelligibility and 3.13 score for naturalness and phone based system 3.08 score in intelligibility and 2.8 score for naturalness. However, as shown in MOS evaluation, when the training data size increases from 300 sentences to 550 sentences, the MOS score of Syllable based TTS system increases more than Phone based TTS system. To end with, the average MOS evaluation of the system from eight listeners/ evaluators for the five Amharic sentences is found to be 2.94 and 3.02 for phone based and syllable based respectively. Therefore, Syllable based

TTS system outperforms that the system using phone as basic unit with 550 sentences' training data.

According to the MOS results, the synthesizer is categorized as good in terms of both intelligibility and naturalness. The result looks encouraging and further improvement of intelligibility and naturalness depend on proper works in different context such as phoneme coverage, lexicon, and question set.

6.2 Recommendation

Noise free speech is very important in speech synthesis contrary to speech recognition which sometimes might require speech with noise. Unlike speech recognition, noise-free-speech in speech synthesis is a priority unless the TTS synthesis is to be trained using data collected for ASR or be used with an ASR, like this study.

Including an Amharic language question set, pronunciation dictionary with full context can also significantly positively affect the quality of synthesized speech. The creation of such a dictionary will be of great benefit to the language itself as it is difficult to find a pronunciation dictionary for Amharic. This will in turn further play a role in adding to the pool of essential components or resources of Ethiopian indigenous spoken language systems and the language itself. Incorporating intonation into the system will also undoubtedly better the quality of synthesized speech as it has been proven in several research articles.

Using a much better labeling algorithm than EHMM labeler or using hand-labeling, better algorithms are continuously being developed. It is, therefore, better to either design a more accurate labeling algorithm or use a state-of-the-art labeling algorithm.

In this study, we did not consider nonstandard words such as numbers, dates, abbreviations, etc. which are challenging in designing the speech synthesis unless there is a standard corpus. This is basically because of the nonexistence of specialized linguistic resources that can be used for developing letter to sound rule and compiled lexicon.

STRAIGHT vocoding could not be explored due to time constraints. STRAIGHT vocoding has, however, been proven to produce better results by several researchers that used the HTS toolkit. Further exploration of this method is therefore recommended towards achieving even better results.

Issues that should be looked at include but are not limited to those listed above. The method (HMM-based speech synthesis) used in this demo may be younger than most other speech synthesis methods, but it offers great opportunities for future research as it promises to outperform many of the other speech synthesis methods.

Finally, this research showed possibility to reuse speech data that requires huge amount of cost and time to prepare and annotate explicit TTS corpus. In addition, it uses as a basis for the following research work by acutely studding appropriate corpus selection techniques. For instances: Speaker adaptive TTS system, Unified ASR and TTS modeling and so on.

References

[1] Dutoit, T., “A Short Introduction to Text-to-Speech” Kluwer Academic Publishers, Dordrecht, Boston, London, (1997).

[2] Sebsibe H/Mariam, “Unit Selection Voice for Amharic using Festvox”, 5th ISCA Workshop on Speech Synthesis, Pittsburgh, USA, June 2004.

[3] CSA (Central Statistics Agency), Addis Ababa, Ethiopia: Central Statistics Agency. http://www.csa.gov.et, 2007.

[4] Lewis Paul, “ Language of the World. Sixteenth edition ”. Dallas, Texas: SIL International Publications. Https://www.ethnologue.com/world, accessed January 26, 2014.

[5] Hudson, Grover, “ The World’s Major Languages : Amharic. In The World’s Major Languages. Second Edition.”, Pp. 594–617. Oxon and New York: Routledge, 2009.

[6] Abyssinica dictionary, “Amharic, the Official Language of Ethiopia”, available at http://dictionary.abyssinica.com/amharic.aspx, 2015.

[7] Wolf Leslau, “ Introductory Grammar of Amharic.” Wiesbaden: Harrassowitz, 2000.

[8] Hudson, Grover, “ The World’s Major Languages : Amharic. In The World’s Major Languages. Second Edition.”, Pp. 594–617. Oxon and New York: Routledge, 2009.

[9] A. Stan, Romanian hmm-based text-to-speech synthesis with interactive intonation optimization," Ph.D. dissertation, Technical University of Cluj-Napoca, 2011.

[10] Solomon Teferra Abate. “ Automatic Speech Recognition for Amharic.” Ph.D. Thesis. Available at: http://www.sub.uni-hamburg.de/opus/volltexte/2006/2981/pdf/thesis.pdf, 2006.

[11] Benesty, J., Sondhi, M. M., and Huang, Y. A. , Springer Handbook of Speech Processing. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[12] Sebsibe, H., Kishore, S., Black, A., Kumar, R., & Sangal, R. , “Unit Selection Voice for Amharic Using Festvox”, Language Technologies Research Center International Institute of Information Technology, 2004.

[13] Bereket Kasaye, Developing A Speech Synthesizer For Amharic Language Using Hidden Markov Model, MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia, 2008.

[14] Tadesse Anberbir and Tomio Takara, “Development of an Amharic Text-to-Speech System Using Cepstral Method”, Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages AfLaT2009, pages 46–52, Athens, Greece, 31 March 2009.

[15] Michael Melese Woldeyohannis, Laurent Besacier, Million Meshesha,“Amharic Speech Recognition for Speech Translation ”, Addis Ababa University, Addis Ababa, Ethiopia, 2016.

[16] Henock Lulsegede, (2003). “Concatenative Text-to-Speech (TTS) synthesis for Amharic language”, MSc thesis, Addis Ababa University, Addis Ababa, Ethiopia.

[17] Lemmetty S., “Review of Speech Synthesis Technology”. MSc.Thesis laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Finland, 1999.

[18] Fujisaki, Y. ,”Sonority and Its Role for Syllabification.” Department of Humanities, Natural Language. Kochi University, Japan, 1995.

[19] Huang , X., “Speech Synthesis”, Prentice Hall PTR. Phonetics, 53-59, 2001.

[20] Rong-Wei, J., “Corpus-Based Unit Selection for Natural-Sounding Speech Synthesis”, Massachusetts Institute of Technology, Doctor of Philosophy in Electrical Engineering and Computer Science. Development, 2003.

[21] Morais, E., & Violaro, F. “Data-Driven Text-to-Speech Synthesis”, School of Electrical and Computer Engineering, University of Campinas, São Paulo, Brazil. Science and Language, 04-08, 2005.

[22] Daniel, Y., “Application of the Double Metaphone Algorithm to Amharic Orthography”, International Conference of Ethiopian Studies, Addis Ababa,Ethiopia, 2006.

[23] Honda, M., “Human Speech Production Mechanisms.” Ntt Technical Review, 1, 2003.

[24] Tesfay, Y., "Diphone based TTS synthesis system for Tigrigna Language”, MSc Thesis, Addis Ababa University, Faculty of Informatics, School of Information Science, Addis Ababa, Ethiopia, 2004.

[25] Taylor, P. ,Text-to-Speech Synthesis. Cambridge University Press, 2009.

[26] Dudley, H. , The Carrier Nature of Speech. The Bell System Technical Journal, 1940.

[27] Klatt, D. H., Software for a cascade/parallel formant synthesizer. Journal of The Acoustical Society of America, 67, 1980.

[28] Allen, J., Hunnicut, S., and Klatt, D., From Text to Speech: the MITalk System. Cambridge University Press, 1987.

[29] Apopei, V. and Jitc_a D. , Module for F0 Contour Generation Using as Input a Text Structured by Prosodic Information. In Proceedings of SPED 2007.

[30] Benesty, J., Sondhi, M. M., and Huang, Y. A. , Springer Handbook of Speech Processing. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[31] Bickley, C., Stevens, K., and Williams, D. , A framework for synthesis of segments based on pseudoarticulatory parameters, 1997

[32] Lambert, T. and Breen, A. P. , A database design for a TTS synthesis system using lexical diphones. In Proceedings of Interspeech, 2004.

[33] Olive, J. P., Greenwood, A., and Coleman, J. Acoustics of American English speech: a dynamic approach. Springer, 1993.

[34] Black, A. and Campbell, N. , Optimising selection of units from speech database for concatenative synthesis. In Proc. EUROSPEECH-95, 1995.

[35] Moulines, E. and Charpentier, F. , Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication, 1990.

[36] Falaschi, A., Giustiniani, M., and Verola, M. , A hidden Markov model approach to speech synthesis. In Proceedings of Eurospeech, volume 1989

[37] Zen, H., Toda, T., Nakamura, M., and Tokuda, K., Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005.

[38] Toda, T. and Tokuda, K. , A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Trans. Inf. & Syst., 2007.

[39] Acero, A. Formant analysis and synthesis using hidden Markov models. In Proc. of Eurospeech, 1999.

[40] Kawahara, H., Masuda-Katsuse, I., and Cheveigne, A., Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds. Speech Communication, 1999.

[41] Yamagishi, J. Average-Voice-Based Speech Synthesis. PhD thesis, Tokyo Institute of Technology, Tokyo, 2006.

[42] Tokuda, K., Masuko, T., Miyazaki, N., and Kobayashi, T., Multi-space probability distribution HMM. IEICE Trans. Inf. & Syst., 2002a.

[43] Imai, S., Sumita, K., and Furuichi, C., Mel log spectrum approximation (mlsa) filter for speech synthesis. Electronics and Communications in Japan (Part I: Communications), 1983.

[44] K. Tokuda, H. Zen, J. Yamagishi, T. Masuko, S. Sako, A. B. Black, and T. Nose, "The HMM-Based Speech Synthesis System (HTS) Version 2.1." [Online]. Available: http://hts.sp.nitech.ac.jp/, 2016.

[45] Young, S., Everman, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. , The HTK Book Version 3.1., 2001.

[46] Ohtani, Y., Toda, T., Saruwatari, H., and Shikano, K. , Maximum likelihood voice conversion based on GMM with STRAIGHT mixed excitation. In Proceedings of Interspeech 2006.

[47] Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J., Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. IEEE Audio, Speech, & Language Processing, 2009.

[48] Solomon Teferra Abate, Martha Yifiru Tachbelie, Wolfgang Menzel ,“Amharic Speech Recognition: Past, Present and Future”, University of Hamburg, Department of Informatics, natural language systems division.

[49] D. H. Klatt, “Review of Text to Speech Conversion for English”, Journal of the Acoustic Society of America, 1987.

[50] K. Tokuda, et al. ,“Multi-Space Probability Distribution HMM”, IEICE Trans. Inf. & System, Vol. E85-D, No.3, pp. 455-464., 2003.

[51] Jurafsky, D., and Martin, J. H., “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.”, 2006.

[52] Duanmu, S. ,”Syllable Structure the limit of variation.” Oxford University Press.UK., 2008.

[53] Aster Taddese, “The syllable structure of Amharic and syllabification of Medial Consonant Clusters and Geminates.” B.A thesis in Linguistics, Addis Ababa University, Addis Ababa, Ethiopia, 1981.

[54] Mulugeta Seyoum, “The syllable Structure and Syllablification in Amharic. Masters of philosophy in general linguistic thesis.” Department of Linguistics, Trondheim, Norway, 2001.

[55] Yule, G. ,”The study of language: an introduction (3rd edition).” Cambridge: Cambridge University Press, 2006.

[56] H. Nirayo, " Modeling improved Amharic syllbification algorithm (unpublished)," Addis Ababa University, computer science department, 2011.

[57] Baye Ymam, “አጭርና ቀላል የአማርኛ ሰዋስው”: (“Short and simple Amharic Grammar”). Addis Ababa, Ethiopia, 2010.

[58] Tadesse Anberbir and Gasser, M, “Grapheme-to-Phoneme Conversion for Amharic Text-to-speech System.” Conference on Human Language Technology for Development. Alexandria, Egypt, 2011.

[59] Hudson, G., “Phonology of Ethiopian Languages: The Handbook of Phonological Theory.” Glodsmith, John A. Blackwell Publishing, 1996.

[60] Rose, S., “Theoretical issues in comparative Ethio-Semitic Phonology and Morphology, PHD thesis,” Department of Linguistics. McGill University, Montreal, 1997.

[61] Laine Berhane, “Text To speech Synthesis of the Amharic Language”, Master’s Thesis, Addis Ababa University, 1998.

[62] Yibeltal Tefera, “Formant-Based Speech Synthesis: A Case of Amharic Words”. MSc Project, Faculty of Informatics, Addis Ababa University, Ethiopia, 2008.

[63] Habtamu Haye, “Amharic concatenative Text- To-Speech (TTS) synthesis system using syllabic unit”, MSc project, Addis Ababa University, Addis Ababa, Ethiopia, 2007.

[64] Henock Lulsegede, “Concatenative Text-to-Speech (TTS) synthesis for Amharic language”, MSc thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2003.

[65] Firdyiwek, Yitna, and Daniel Yaqob. "The system for Ethiopic representation in ASCII." ,1997.

[66] Wolf Leslau , “Introductory Grammar of Amharic. Wiesbaden: “Harrassowitz, 2000.

[67] Y. Baye,የአማርኛ ሰዋሰው,1986.

[68] http://www.arts.gla.ac.uk/ipa/ipachart.html, Last Accessed on May 01, 2017.July 25, 2008.

[69] http://www.phon.ucl.ac.uk/home/sampa/, Last Accessed on May 01, 2017.July 25, 2008.

[70] A. Getahun, ዘ ሞና ዊ የአማረኛ ሰዋስው በቀላል አቀራረ ብ , 1989.

[71] Roach P., “A Little Encyclopaedia of Phonetics”, Available at http://www.linguistics.reading.ac.uk/staff/Peter.Roach/PAPERS/encyc2.pdf, Last Accessed on June 1, 2006.

[72] Y. Baye,የአማርኛ ሰዋሰው,1997.

[73] W. Leslau, "Introductory Grammar of Amharic," Wiesbaden: Harrassowitz, 2000.

[74] Alemayehu Hailu, “Lexical Stress in Amharic. In Journal of Ethiopian Studies:” Vol XX, Addis Ababa University. Addis Ababa, Ethiopia, 1987.

[75] A.W. Black and K.A. Lenzo, Building Synthetic Voices, Statistical Parametric Synthesis, Building a CLUSTERGEN Statistical Parametric Synthesizer, Jan 2007.

[76] S. J. Young, J. J. Odell and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in proc. of the workshop on Human Language Technology, Plainsboro, New Jerey, USA, March, 1994.

[77] K. Shinoda and T. Watanabe, "Acoustic Modeling Based on the MDL Principle for speech recognition," in proc. of EuroSpeech-97, Rhodes, Greece, September, 1997.

Appendixes

Appendix A: List of all Amharic core characters

First	Second	Third	Fourth	Fifth	Sixth	Seventh
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

Appendix B: Amharic characters ASCII transliteration

IPA	Transcription	Amharic equivalence
Consonants		
[p]	[p]	ፐ
[t]	[t]	ት
[k]	[k]	ክ
[b]	[b]	ብ
[d]	[d]	ድ
[g]	[g]	ግ
[p']	[px]	ጵ
[t']	[tx]	ጥ
[c']	[cx]	ጭ
[q]	[q]	ቅ
[f]	[f]	ፍ
[s]	[s]	ስ
[ʃ]	[sx]	ሽ
[h]	[h]	ህ
[s']	[xx]	ጽ
[tʃ]	[c]	ች
[g']	[j]	ጅ
[m]	[m]	ም
[n]	[n]	ን
[n']	[nx]	ኝ
[l]	[l]	ል
[r]	[r]	ር
[j]	[y]	ይ
[w]	[w]	ው
[v]	[v]	ቭ
[z]	[z]	ዝ
[z']	[zx]	ዥ
Vowels		
[ɛ]	[e]	ኧ
[ʊ]	[u]	ኡ
[ɪ]	[ii]	ኢ
[ɑ]	[a]	ኣ
[ɐ]	[ie]	ኤ
[ɨ]	[ix]	እ
[ɔ]	[o]	ኦ

Appendix C: ASCII Translation python code

```
# encoding:utf-8
import codecs, sys, string
worddict = {}
mapfile = codecs.open("chartable.txt", 'r', 'utf-8')
corfile = codecs.open("dtxtbahiru.txt", 'r', 'utf-7')
outfile = codecs.open("txtout.txt", 'w', 'utf-8')
maps = mapfile.read().encode("utf-8")
corpus = corfile.read().encode("utf-8")
def autodecode(corpus):
    if corpus.startswith(codecs.BOM_UTF8):
        out = corpus.decode( "utf8")
        return out[1:]
    else: return corpus.decode( "utf-8")
mapstripped = autodecode(maps)
corstripped = autodecode(corpus)
for line in mapstripped.split("\n"):
   (i,j) = line.split()
   worddict[j] = i
   print worddict[j]
#for line in corstripped.readline():
for char in corstripped:
   if (char == ".") :
        outfile.write("\n")
   elif worddict.__contains__ (char):
        outfile.write(worddict[char])
   else:
        outfile.write(" ")
    #outfile.close()
```

Appendix D: part of left of left of phoneme question set example

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

QS "LL-Consonant"
{axa^*,axu^*,axii^*,axie^*,axix^*,axo^*,be^*,bu^*,bii^*,ba^*,bie^*,bix^*,bo^*,ce^*,cu^*,cii^*,ca^*,ci
e^*,cix^*,co^*,cxe^*,cxu^*,cxii^*,cxa^*,cxie^*,cxix^*,cxo^*,de^*,du^*,dii^*,da^*,die^*,dix^*,do^*,fe^
*,fu^*,fii^*,fa^*,fie^*,fix^*,fo^*,ge^*,gu^*,gii^*,ga^*,gie^*,gix^*,go^*,ha^*,hu^*,hii^*,hie^*,hix^*,ho
^*,je^*,ju^*,jii^*,ja^*,jie^*,jix^*,jo^*,ke^*,ku^*,kii^*,ka^*,kie^*,kix^*,ko^*,le^*,lu^*,lii^*,la^*,lie^*,lix
^*,lo^*,me^*,mu^*,mii^*,ma^*,mie^*,mix^*,mo^*,ne^*,nu^*,nii^*,na^*,nie^*,nix^*,no^*,nxe^*,nxu^
*,nxii^*,nxa^*,nxie^*,nxix^*,nxo^*,qe^*,qu^*,qii^*,qa^*,qie^*,qix^*,qo^*,pxe^*,pxu^*,pxii^*,pxa^*,p
xie^*,pxix^*,pxo^*,pe^*,pu^*,pii^*,pa^*,pie^*,pix^*,po^*,re^*,ru^*,rii^*,ra^*,rie^*,rix^*,ro^*,se^*,su
^*,sii^*,sa^*,sie^*,six^*,so^*,sxe^*,sxu^*,sxii^*,sxa^*,sxie^*,sxix^*,sxo^*,te^*,tu^*,tii^*,ta^*,tie^*,tix
^*,to^*,txe^*,txu^*,txii^*,txa^*,txie^*,txix^*,txo^*,ve^*,vu^*,vii^*,va^*,vie^*,vix^*,vo^*,we^*,wu^*,
wii^*,wa^*,wie^*,wix^*,wo^*,ye^*,yu^*,yii^*,ya^*,yie^*,yix^*,yo^*,ze^*,zu^*,zii^*,za^*,zie^*,zix^*,zo
^*,zxe^*,zxu^*,zxii^*,zxa^*,zxie^*,zxix^*,zxo^*,xxe^*,xxu^*,xxii^*,xxa^*,xxie^*,xxix^*,xxo^*}

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-h#" {h#^*}

QS "LL-brth" {brth^*}

Appendix E: part of phone set definition

```
(defPhoneSet
 radio
 ;;;  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  -)
(pe  -  0  0  0  -  s  l  -)
(pu  -  0  0  0  -  s  l  -)
(pii  -  0  0  0  -  s  l  -)
(pa  -  0  0  0  -  s  l  -)
(pie  -  0  0  0  -  s  l  -)
(pix  -  0  0  0  -  s  l  -)
(po  -  0  0  0  -  s  l  -)
(te  -  0  0  0  -  s  a  -)
(tu  -  0  0  0  -  s  a  -)
(tii  -  0  0  0  -  s  a  -)
(ta  -  0  0  0  -  s  a  -)
(tie  -  0  0  0  -  s  a  -)
(tix  -  0  0  0  -  s  a  -)
(to  -  0  0  0  -  s  a  -)
.
.
. (h#  -  0  0  0  0  0  0  -)
  (brth -  0  0  0  0  0  0  -)
 )
)
(PhoneSet.silences '(pau h# brth))
```

Appendix F: sample prompts snap shoot

```
(  bd_v_bah_001  "yanixdenxa  dereja  tixmixhixrixtacewixnix
gonixderix temixrewalix" )
( bd_v_bah_002 "yeteleqequtix mixrixkonxocix beakababiiyacewix
selamawii nuro ixnixdiinoru yetixranixsixporixtixna megxgxzxa
genixzebix tesetxixtwacewix mesxenxetacewixnix amelixkixto
beyezonacewix ixnixdederesu meqxqxmiiya ixnixdemiisetxacewixmix
asixtawixqxlix" )
( bd_v_bah_003 "beadiisix abebawix sixtadiiyemix betekahiedutix
huletix gixtxixmiiyawocix bemejemeriiya yetegenanxutix medixnixna
mugerix siimiinixto siihonu bewixtxietumix sosixtix lesosixtix
teleyayixtewalix" )
( bd_v_bah_004 "weriewixnix werie yaderegu mixsixtxiirenxocix
nacewix" )
(  bd_v_bah_005  "iitixyopxixyawiitwa  bebixhierawii  bahixlawii
alebabesix kealemix anixdenxixnetixnix teqedajecix" )
(  bd_v_bah_006  "ketixmixkixhixtix  ixnixdayixqotxerixbixnix
ixnixjii  bealemix  tariikix  wixsixtxix  benecxocix
yalixteregetxecix agerix iitixyopxixya natix" )
( bd_v_bah_007 "ixhixtocu yeierixtixra ziegocixna yesxaixbiiya
degafiiwocix nacewix" )
(  bd_v_bah_008  "ixnanixtemix  meqeberiiya  ixnixdatatxu
tetxenixqequ" )
(  bd_v_bah_009  "anixtonielii  beaxxie  mixnixliikix  fiitix
yefexxemewix  dixfixretix  beiitxaliiyanix  mixkixrix  bietix
asixtecewix" )
( bd_v_bah_010 "gixnix wede hxlawix layix iisayasix ixnixde
lixmadacewix  hulunixmix  yemelixkefix  diipixlomasiiyacewix
ixsixraielixnixmix yasixwerixfacewix jemerix" )
```

For the following corresponding sentences

.ያንደኛ ደረጃ ትምህርታቸውን ጐንደር ተምረዋል

.የተለቀቁት ምርኮኞች በአካባቢያቸው ሰላማዊ ኑሮ እንዲኖሩ የትራንስፖርትና መጓጓዣ ገንዘብ ተሰጥቷቸው መሸኘታቸውን አመልክቶ በየዞናቸው እንደደረሱ መቋቋሚያ እንደሚሰጣቸውም አስታውቋል

.በአዲስ አበባው ስታዲየም በተካሄዱት ሁለት ግጥሚያዎች በመጀመሪያ የተገናኙት መድንና ሙገር ሲሚንቶ ሲሆኑ በውጤቱም ሶስት ለሶስት ተለያይተዋል

.ወሬውን ወሬ ያደረጉ ምስጢረኞች ናቸው

.ኢትዮጵያዊቷ በብሄራዊ ባህላዊ አለባበስ ከአለም አንደኝነትን ተቀዳጀች

.ከትምክህት እንዳይቆጠርብን እንጂ በአለም ታሪክ ውስጥ በነጮች ያልተረገጠች አገር ኢትዮጵያ ናት

.እህቶቹ የኤርትራ ዜጐችና የሻእቢያ ደጋፊዎች ናቸው

.እናንተም መቀበሪያ እንዳታጡ ተጠንቀቁ

.አንቶኔሊ በአጼ ምንሊክ ፊት የፈጸመው ድፍረት በኢጣሊያን ምክር ቤት አስተቸው

.ግን ወደ ኋላው ላይ ኢሳያስ እንደ ልማዳቸው ሁሉንም የመልከፍ ዲፕሎማሲያቸው እስራኤልንም ያስወርፋቸው ጀመር

Appendix G: Configuration of HTS system

Create a directory or folder on the desktop and move to that directory.

- *mkdir HTSmodel*
- *cd HTSmodelt*

Copy all the required downloaded tools into the HTSmodel folder

Installation of speech tools, festival and festvox

To build HMM models, the utterance files which consist of textual features and the duration of each unit in the text to be synthesized are required. To generate these utterance structures, speech tools, festival and festvox are needed. Run the commands given, for the installation of speech tools, festival and festvox respectively.

- *Installation of speech tools*
- *tar –xvf speech_tools-2.4-release.tar.gz*
- *cd speech_tools*
- *./configure --prefix=/home/bahiru/HTSmodel/speech_tools/*
- *Make*

The first command is for extraction tar files. The most popular archiving tool used in UNIX and Linux is the "tar" command. The second command gives the path to the speech files folder. Next command is for running the configure script. "Configure" is an executable script designed to help in developing a program to be run on computers and matches the libraries on the user's computer, with those required by the program, before compiling it from its source code.

Installation of Festival

- *cd..*
- *tar –xvf festival-2.4-release.tar.gz*
- *cd festival*
- *./configure --prefix=/home/bahiru/HTSmodel/festival/*
- *make*

Installation of Festvox

- *cd..*
- *tar -xvf festvox-2.7.0-release.tar.gz*
- *cd festvox*
- *./configure --prefix=/home/bahiru/HTSmodel/festvox/*
- *Make*

Installation of HMM Toolkit (HTK) and Patch for HTS

HTK along with patch files provided for HTS are used to train context independent and context-dependent hidden Markov models. The procedure for installing these tools is given below:

- *cd..*
- *mkdir hts_patch*
- *tar –xvf HTK-3.4.1.tar.gz -C ./hts_patch*
- *tar –xvf Hdecode-3.4.1.tar.gz -C ./hts_patch*
- *tar -xvf HTS-2.3alpha_for_HTK-3.4.1.tar.bz -C ./hts_patch*
  (Copy all the extracted files into a folder and name it as hts_patch)
- *cd hts_patch/htk*

  Run the command given below to include a patch file for HTS.

- *patch -p1 -d . < ../hts_patch/HTS-2.3alpha_for_HTK-3.4.1.patch*

The executable such as Hcopy, HList, HInit, will be compiled in /usr/local/HTS-2.3alpha/bin

- *./configure*

  or
- *./configure --prefix=/home/bahiru/HTSmodel/hts_patch/htk/*
- *make*
- *sudo make install*
- *sudo make hlmtools install-hlmtools*
- *sudo make hdecode install-hdecode*

Installation of HTS Engine

This synthesize speech waveform from trained HMMs

- *cd..*
- *mkdir hts-engine*
- *tar –xvf hts_engine_API-1.10.tar.gz*
- *cd hts_engine_API-1.10*
- *./configure –prefix=/home/bahiru/HTSmodel/hts_engine_API-1.10/*
- *make*
- *sudo make install*

Installation of SPTK (Signal Processing Toolkit)

The SPTK functions such as mgecp, x2x, lsp2lpc, etc will be compiled in /home/…/SPTK/bin

- *cd..*
- *mkdir SPTK*
- *tar –xvf SPTK-3.9.tar.gz*
- *cd SPTK-3.9*
- *./configure --prefix=/home/bahiru/HTSmodel/SPTK-3.9/*
- *make*
- *sudo make install*

Installation of Active Tcl

Execute the following commands to install Active Tcl in /home/bahiru/HTSmodel/bin/

- *cd ..*
- *tar –xvf ActiveTcl8.4.19.4.292682-linux-ix86.tar.gz*
- *cd ActiveTcl8.4.19.4.292682-linux-ix86*
- *sudo ./install.sh*

Finally, configuring HTS Demo with those tools, start training and synthesis process.

Appendix H: Evaluation format

Test Data (Sentence)	***Ranks given by different evaluators for intelligibility/naturalness***							
	Person 1	***Person 2***	***Person 3***	***Person 4***	***Person 5***	***Person 6***	***Person 7***	***Person 8***
1								
2								
3								
4								
5								

Appendix I: Evaluation Questionnaire

Questionnaire

Addis Ababa University

School of Information Science

Users' Evaluation of syllable based speech synthesis using HMM for Amharic Language.

Educational Background	**sex**	**age**

The aim of this questionnaire is to evaluate the performance of the syllable based speech synthesis using Hidden Markov Model for Amharic. So, we kindly request you to consider each question critically and give the rank honestly.

The following two questions deals in measuring the intelligibility and naturalness of the synthesized speech. Intelligibility measures the understandability of the synthesized speech and naturalness measure to what extent that synthesized speech looks like human speech.

1. How do you judge the understandability of the synthesized speech?

a. Excellent

b. Very Good

c. Good

d. Fair

f. bad

2. How do you judge the naturalness of the synthesized speech?

a. Excellent

b. Very Good

c. Good

d. Fair

f. bad

Thank you

Appendix k: the frequency of syllables and phones covered within selected dataset

Order number	Amharic syllables	Transcribed syllables	frequency	Amharic phones	Transcribed phones	frequency
1	ኀ	nix	1600	ኀ	n	1600
2	ት	tix	1135	ት	t	1135
3	ዉ	wix	990	ዉ	w	990
4	ስ	six	824	ስ	s	824
5	ያ	ya	733	ል	l	605
6	የ	ye	694	ር	r	598
7	ተ	te	685	እ	ix	594
8	በ	be	661	ም	m	488
9	አ	ixa	652	ች	c	476
10	ለ	le	623	ይ	y	372
11	ል	lix	605	ግ	g	300
12	ር	rix	598	ድ	d	283
13	እ	ix	594	ኢ	ii	261
14	መ	me	501	ክ	k	247
15	ም	mix	488	ብ	b	228
16	ች	cix	476	ህ	h	181
17	ና	na	440	ጵ	px	174
18	ገ	ge	438	ቅ	q	174
19	ደ	de	438	ጥ	tx	171
20	ነ	ne	426	ፍ	f	108
21	ማ	ma	387	ሽ	sx	85
22	ይ	yix	372	ዝ	z	52
23	ባ	ba	321	ኝ	nx	51
24	ግ	gix	300	ጵ	xx	47
25	ሚ	mii	300	ጅ	j	38
26	ድ	dix	283	ጭ	cx	31
27	ላ	la	283	ኪ	kx	20
28	ራ	ra	268	ጓ	gx	17
29	ቸ	ce	264	ኧ	hx	10
30	ኢ	ii	261	ኡ	u	8