Addis Ababa University

College of Natural and Computational Sciences

Text to Speech Synthesizer for Afaan Oromoo using Statistical Parametric Speech Synthesis

Muhidin Kedir Wosho

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

Addis Ababa, Ethiopia

June, 2020

Addis Ababa University

College of Natural and Computational Sciences

Muhidin Kedir Wosho

Advisor: *Dida Midekso (PhD)*

This is to certify that the thesis prepared by *Muhidin Kedir Wosho*, titled: *Text to speech synthesizer for Afaan Oromoo using statistical parametric speech synthesis based on HMM* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the examining committee:

	Name	Signature	Date
Advisor:	Dida Midekso (PhD)	_____	_____
Examiner:	Solomon Gizew (PhD)	_____	_____
Examiner:	Minale Ashagrie (PhD)	_____	_____

Abstract

Speech synthesis systems are concerned with generating a natural sounding and intelligible speech by taking text as input. Speech synthesizers are very essential in helping impaired people, in teaching and learning process, for telecommunications and industries. Nevertheless, it has been a lot of challenging such as text processing, grapheme to phoneme and modeling prosody for years. Text preprocessing includes tokenization and normalization and then converting the grapheme representation of sounds to their phonetic representation and modeling prosodic features of various speaking styles. To address these challenges, different techniques have been studied and implemented. Speech synthesizers using statistical parametric speech based on hidden Markov model (HMM) are done for foreign languages which are not applicable for Afaan Oromoo language since the Afaan Oromoo language's special characteristics are not considered in foreign synthesizers. Statistical parametric speech synthesis based on HMM techniques is chosen for these research because it is a model based that require less storage, it learn properties of data rather store the speech, small run time, and easy to integrate with small handheld devices. The Afaan Oromoo text to speech synthesis system has been developed using statistical parametric speech synthesis based on a hidden Markov model. The synthesizer has two main components: training and testing phases. In the training phase, source and excitation parameters of the speech are extracted from speech database. The speech and phonetic transcriptions are automatically segmented using EHMM labeling. During testing phase, the input text is processed to form phonetic strings along with the trained models. Finally, the synthesized speech is generated from speech parameters. In order to train the system being developed, we collected four hundred sentences and speeches. Additionally, we used ten sentences to test the performance of the system. In this study, the subjective Mean Opinion Score (MOS) and objective Mel Cepstral Distortion (MCD) evaluation techniques are used. The subjective results obtained using the mean opinion score (MOS) is 4.3 and 4.1 in terms of the intelligibility and naturalness of the synthesized speech respectively. The objective result obtained using mean opinion score is 6.8 out of 8 which is encouraging.

Keywords: Statistical Parameter Speech Synthesis, Text to Speech, Afaan Oromoo, Hidden Markov Model based speech synthesis.

Dedication

I would like to dedicate this paper to my family.

Acknowledgments

First and for most, I would like to express my heartfelt thanks to the Almighty Allah, for giving me the strength, determination, endurance and wisdom to bring this thesis to completion. I would also like to express my heartfelt and deepest sense of gratitude to my advisor Dr. Dida Midekso for his constructive comments, suggestions, guidance, inspiring and enlightening ideas. Starting from shaping and reshaping the tittle of this thesis, his support and encouragement is high and without all these efforts, this work wouldn't have been a success at all.

I would also like to thank Dr. Mohammed Kalid, Dr kalil Abu, Ato Abdurazek Kedir, Ato Mubarek Dubiso, Ato Mohammed Ebrahim, Ato Argaw Korsa, Ato Luel Negash, Ato Gobena Haso, Ato Edao Abdi, Ato Fitsum Andresom, Ato Sultan Esmu, W/o Misra Kedir and W/o keebeki Mulata for their help on linguistic parts and suggestions on this study.

I am also grateful to my family, especially my mother Rare Rike and my father Kedir Wosho for their love and endless support and encouragement throughout my academic life. Using this opportunity, I would like to express my deepest gratitude to my wife Zuleyka Woliyi Dalu for her love, care, patience and moral encouragement she has given me throughout my study.

I feel happy to thank Mizan-Tepi University for sponsoring me. Last, but not least, I would like to thank my friends for sharing me their knowledge and experience, and suggestions.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

AOTTS- Afaan Oromoo Text to speech synthesizer

ASR-Automatic Speech Recognition

CALL- Computer Aided Learning Language

CART- Classification and Regression Tree

DSP - Digital Signal Processing

LPC- Linear predictive Code

LTS- Letter to Sound

EHMM- Ergodic hidden Markov models

HCI- Human Computer Interaction

HMM- Hidden Markov Model

HRL- High Resource Language

HTK- Hidden Markov Model Tool Kit

HTS- HMM-based speech synthesis system

IPA-International Phonetic Alphabet

MCD - Mel Cepstral Distortion

MFCC - Mel frequency cepstral Coefficients

NLP- Natural Language Processing

NSW – Non Standard Words

SGD- Speech Generating Device

SPSS – Statistical Parametric Speech Synthesis

SPTK – Speech Processing Tool Kit

TTS- Text To speech

UTS –Unit Selection Technique

Chapter One: Introduction

1.1 Background

Speech is the most natural way of human communication, which has been the driving force underlying several significant advances in speech technology [1]. As computers become more functional and prevalent, demands for technologies in speech processing area, such as speech recognition, dialogue processing, speech understanding, natural language processing, and speech synthesis are increasing to establish high-quality human-computer communication with voice [2].

Natural language processing (NLP) is a field which employs computational techniques for the purpose of learning, understanding and producing human language content [3] at the intersection of computer science, artificial intelligence and computational linguistics. It is used for both generating human readable information from computer system and converting human language into more formal structures that a computer can understand [4] . Therefore, it is important for the purpose of scientific, economic, social and cultural reasons. It is also undergoing rapid growth in its areas where its theories and methods are deployed in a variety of new language technologies [5].

The goal of NLP is to design and build software that will analyze, understand and generate languages that humans use in the natural manner [3]. A major limitation of NLP today is the fact that most NLP resources and systems are available only for high-resource languages (HRLs), such as English, French, Spanish, German, and Chinese. In contrast, many under-resource languages such as Amharic, Afaan Oromoo, Tigrinya and etc are written by many people with reference to under resources where no systems exist [4]. The future challenge for the language community is how to develop resources and tools for thousands of limited resource languages.

Text-to-speech synthesis (TTS) is one of the key technologies in speech processing techniques for creating speech signal from arbitrarily given text in order to transmit information from a machine to a person by voice [2]. The main goal of TTS synthesis is to produce a natural and intelligible sounding speech from arbitrary text [6]. Moreover, the current trend in TTS research calls for systems that enable producing speech in different speaking styles with different voice characteristics with small run time and emotions [7]. In

order to fulfill these requirements, the best suitable approach is statistical parametric speech (SPS) synthesis system technique based on hidden Markov model (HMMs) [8].

Samson Tadese [9] developed a model with reference to Afaan Oromoo text to speech using a diphone based concatenative approach speech synthesis. But, there are various limitations in his work concerning distortion from discontinuities in concatenation points, the memory requirements are typically very high, hence it requires huge data for training and recording speech time consuming. Moreover, the researchers didn't consider the non- standard words, germination and epenthesis, which is necessary parts of the language. Data collection, labeling of the speech and real time processing ability is also another issue with such systems. In contrast to concatenative approach, the statistical parametric speech based on HMM technique only requires the statistical model parameters to be stored (machine learning), requires less data for training, no need of memorizing all data hence we can easily modify model to convert the various voice quality. Therefore, it is important to study the statistical parametric speech synthesis based on hidden Markov model (HMM) technique for Afaan Oromoo.

Generally, modern speech synthesizers are able to achieve high intelligibility. However, they still undergo from a high quality or unnatural speech. In order to, increase the naturalness of TTS there has been a noticeable shift from rule based towards corpus, based on statistical parametric speech synthesis [6] . The corpus based unit selection technique (UST) is by far the best method of producing the natural speech. However, this method requires a large database in terms of size in gigabyte with single speakers support and limited to domains. Therefore, HMM based speech synthesis has proven to be an efficient and best parametric model of the speech acoustics in the circumstances of speech synthesis [7]. It requires small database size (less memory) and ability to produce an intelligent and natural speech that supports different voice characteristics and easy for adaptation which is integrated with small handheld device, also requires small data for training, they are robust and flexible [2, 7]. As far as the researchers' knowledge is concerned, there were no attempts made to develop TTS for Afaan Oromoo using the statistical parametric speech synthesis based on HMM technique.

1.2 Motivation

The total number of Afaan Oromoo speakers is estimated to be more than 40 millions ranging from Ethiopia to Tanzania according to 2007 Ethiopian Statistical Agency [10]. In addition to this, the language is also used as a medium of instruction in the primary schools in the Oromia region and as an official language in the region and some regional offices in Addis Ababa as well [9]. Nowadays, literatures, books, newspapers published in Afaan Oromoo have been increased over the years [10].

The major challenge faced is by the visually impaired students to learn with their students as they are naturally or accidentally disabled by the sight. Text to speech is also used in different tasks such as e-mail reading, reading fictions and nonfiction books, reading message sent from telecommunication, and newspapers. But, these said activities become tedious because reading is time consuming and challenging for the disabled people. Therefore, the motivation behind this work is to build a high quality TTS system for helping blind, visually impaired, vocally challenged people as well as foreigners to access the written text.

1.3 Statement of the Problem

Speech synthesis is becoming one of the most important steps towards improving the human interface to a computer. Speech synthesis system that generates natural sound and intelligible speech with small resource requirement is vital for many application areas. The speech synthesis systems that are developed for one language are not applicable for others, because speech synthesis systems are highly dependent on the characteristics of a language.

Presently, there are many functional speech synthesizers which are developed and used in foreign languages such as Japan, Vietnam, Bengali and English [12, 13, 36, 53]. In addition to other foreign languages, Ethiopian researchers have been able to develop a prototype TTS synthesizer model for regional languages such as Tigrigna , Afaan Oromoo, Amharic and Wolayta language respectively [5, 9, 16,55]. Samson Tadese [9] developed a model with reference to Afaan Oromoo text to speech using a diphone based concatenative approach speech synthesis. But, there are various limitations in his work concerning distortion from discontinuities in concatenation points, the memory requirements are typically very high, hence it requires huge data for training and recording speech time consuming. Moreover, the

researchers didn't consider the non- standard words, germination and epenthesis, which is necessary parts of the language. Data collection, labeling of the speech and real time processing ability is also another issue with such systems. In contrast to concatenative approach, the statistical parametric speech based on HMM technique only requires the statistical model parameters to be stored (machine learning), requires less data for training, no need of memorizing all data hence we can easily modify model to convert the various voice quality. Therefore, it is important to study the statistical parametric speech synthesis based on hidden Markov model (HMM) technique for Afaan Oromoo.

Hence, the aim of this research to come up with a text-to-speech synthesis for Afaan Oromoo language that generates natural sounding and intelligible speech which is vital for many application areas. This research tries to answer the following research questions:-

- ✓ To what extent is it possible to develop a TTS synthesizer for Afaan Oromoo language?
- ✓ How effective is synthesizing Afaan oromoo using HMM?
- ✓ How to measure the performance of text to speech synthesizer for Afaan Oromoo language?

1.4 Objective

General Objective

The general objective of this study is to design a text to speech synthesizer for Afaan Oromoo using statistical parametric speech synthesis based on HMM.

Specific Objectives

The specific objectives of this research are as follows:

- ➢ To review literatures on the area of text to speech synthesis.
- ➢ To study the structure of Afaan Oromoo language.
- ➢ To prepare Afaan Oromoo corpus.
- ➢ To design a text to speech synthesizer for Afaan Oromoo language.
- ➢ Evaluate the performance of the developed system with different data sets.

1.5 Methods

Literature Review

To gain clear picture of the work, literatures will be reviewed on the area of text to speech synthesis. The literature review will come from various sources such as published papers, books, and other related materials to understand the area in detail.

Data Collection

The main objective of the data collection is to prepare corpus for training and testing the data set for the desired system. In this research, we will collect data from different sources such newspapers, Holy books, magazines, and others.

Development tools

In order to test the proposed solution, a sample of system will be developed.

Evaluation

Basically, there are subjective and objective evaluation metrics of TTS, where both are assessed based on the intelligence and naturalness of user respondents. The intelligibility of a system is how much of the synthetic output are understandable by users while the naturalness of a system is the measure of how much the artificially produced sound resembles the natural sound.

1.6 Scope and Delimitations of the study

This study focuses on a TTS system for Afaan Oromoo language based on HMM synthesis. However, voice conversation mechanism, spoken language idioms such as, time, acronym, abbreviation, date and other numeric are not considered.

1.7 Application of Results

Text-to-speech systems have various range of applications. The most important and useful application field in speech synthesis is the reading systems for the blind and vocally handicapped, where a system would read some text from a book and converts it into speech. In addition, it is highly applicable for language learning. A high quality of TTS synthesis can be coupled with a computer aided learning system, and provides tools to help a child or a student to learn the correct pronunciation of words. The application of this study will also benefit anyone who wants to synthesize speech of Afaan Oromoo and

wants to conduct further research in other NLP applications with the language. Therefore, it will have a great contribution for the development of a full-fledged text to speech synthesis for Afaan Oromoo language.

1.8 Organization of the Thesis

This thesis is organized into six chapters including the current one. The Second Chapter focuses on the literature review. It deals with the fundamental speech production system, basics of text to speech synthesis, synthesis methods and overview of Afaan Oromoo structure. The Third Chapter discusses the works related to this study. In Chapter Four, we focused on the system design and its prototype. Chapter Five deals with the experimentation and evaluation of the prototype. Finally, conclusion and future works are discussed in Chapter Six.

Chapter Two: Literature Review

This chapter describes the human speech production mechanism, fundamental concepts of speech synthesis, the history and development of speech synthesis techniques. The basic architecture of a TTS system and the roles of the linguistic front-end and the signal processing back-end are also described.

2.1 Fundamental of Speech Production System

A speech production is a model that simulates the physical process of human speech production comprising of lungs, vocal cords, and the vocal tract [9]. Speech production can be described as a result of three main components, namely the respiratory system, larynx, and the vocal tract. Speech is produced by regulating the airflow from lungs through throat, nose and mouth [11]. The produced speech sounds can be basically classified into three categories. Firstly, voiced speech sounds are produced by using the air pressure to get the vocal folds into vibratory motion. Secondly, unvoiced sounds are produced by constricting the airflow somewhere in the vocal tract. Thirdly, unvoiced stop consonants are produced by completely stopping the airflow in the vocal tract [11]. Figure 2.1 shows human speech production system [12,33].

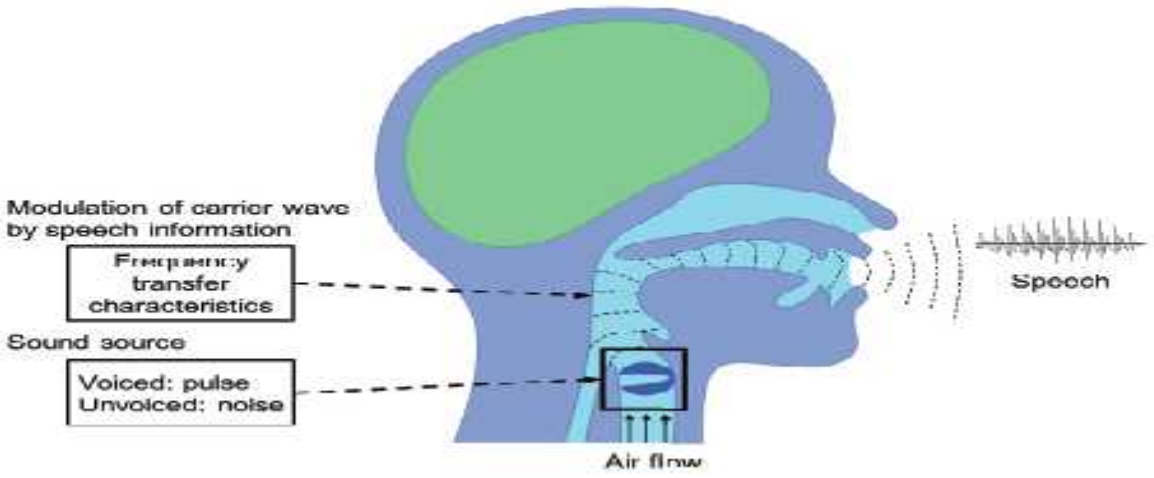


Figure 2. 1: *The human speech production system [33].*

2.2 History of Speech Synthesis

The first successful attempts to produce artificial speech was based on imitating the speech production organs. In 1791 Wolfgang von Kempelen wrote a book that explored speaking machine, a vibrating reed (glottis), which consisted of bellows (lungs), and a rubber tube modeling the vocal tract [7]. In early 1920s, research speech synthesis moved direction from physical modelling to use electric circuitry. In 1939, Dudley developed the first machine that could produce continuous speech [1]. Starting from 1960s, speech synthesis techniques split into two paradigms: signal-based and articulatory-based synthesis [13]. In signal- based synthesis, the speech signal itself is modelled in order to reconstruct the signal so that it is perceptually close to the original one. In parallel with the development of speech processing techniques, the generation of speech from text instead of mere speech analysis and synthesis was gaining more interest. Development work was done both at the signal and linguistic levels, and the first full TTS synthesis system was developed by Umeda et al. in 1968 [3]. In 1973, Holmes [4] also proposed the use of inverse-filtered glottal-flow wave forms to improve the naturalness of synthetic speech.

In the 1980s, the paradigm started to shift from light and rule-based expert systems to database (corpus) based systems, as speech data could be recorded, stored, and processed more efficiently. This was enabled by the reduced price in memory and increased computing power, and motivated by the potential to achieve better quality using concatenative synthesis methods [1]. The high quality concatenative speech synthesis appeared in the 1990s, which was made possible again by the increased computational capacity and new methods in signal and natural language processing. Purely software based synthesizers also became feasible for the same reasons, such as ATR's CHATR and University of Edinburgh's Festival speech synthesis system [4].

In the 1990s, statistical parametric speech synthesis (SPSS) using HMMs was introduced by Tokuda et al [3], and it has been the most researched paradigm in speech synthesis. In SPSS, speech is converted into parameters that are statistically modelled and generated for synthesis use. The most widely used technique for SPSS is the HMM-based speech synthesis systems (HTS, 2014) developed in Japan [1].

2.3 The challenges and importance of text to speech synthesis

The process of converting text to speech has a lot of challenges such as text normalization, grapheme to phoneme, evaluation, Prosodic and emotional content [14]. Texts are full of heteronyms, number and abbreviations that all require expansion into a phonetic representation. Finding the appropriate pronunciation for the given texts having its own language structure, evaluation of TTS system is lack standard evaluation metrics and modeling correct acoustic features also another challenges in speech synthesis. One of the biggest challenges in the speech synthesis the units connect to each other in a continuous way so that the amount of audible distortion.

The process of converting TTS using a machine is less complex than the corresponding human process due to many simplifications [1]. Firstly, machines are not yet capable of extracting high level meaning from text, so this step is generally omitted in TTS synthesis. Secondly, speech generated by the physiological speech production mechanism is governed by the laws of physics as the air-flow from the lungs is modulated in the larynx and the vocal tract [1]. Despite the difficulty of speech synthesis, present day high quality TTS systems are approaching many of the qualities of human speech, which makes them useful and acceptable for increasingly many purposes [15].

TTS is often used as a communication aid for people with difficulties in producing speech due to vocal disorders or dyslexia. For example, a Speech Generating Device (SGD) can be used as a personal device carried by the user to communicate with people through synthetic speech. Probably, the most famous user of an SGD is Stephen Hawking, an English theoretical physicist and cosmologist who is also known for his work in popular science [1].

Speech synthesis is also widely used in various telecommunication services, such as in information retrieval systems. TTS synthesis also finds applications in Computer Assisted Language Learning (CALL) in the creation of audio books, talking toys, and entertainment products such as games and animations. In connection with other speech technologies, such as Automatic Speech Recognition (ASR), speech synthesis is also increasingly used in Human Computer Interaction (HCI), such as in mobile phones, where a user can give commands or ask questions, based on which the phone can give answers and feedback through synthetic

speech such as Google. As a result, speech synthesis provides a fundamental tool for research in the production and perception of speech and language [1].

2.4 Basic Architecture of Text-to-Speech (TTS) Synthesis

Speech is the most widely used comprehensive means of communication between people [16]. The basic architecture of a TTS system has two main parts with four components [17]. Figure 2.2 shows the basic architecture of TTS [13].

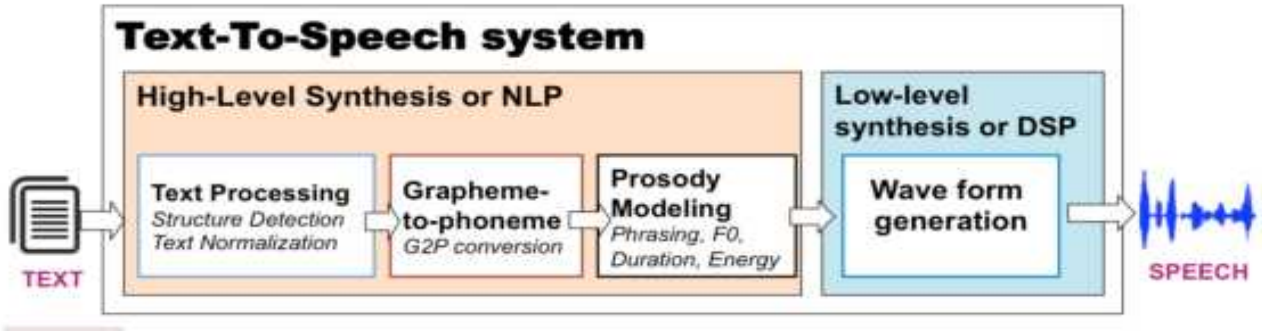


Figure 2. 2: *Basic Architecture of a TTS system [13].*

The NLP part which acts as *front-end* has three components and a Digital Signal Processing (DSP) part acts as *back-end system* with one component. The components of each part are described in the following sub-sections.

2.4.1 Natural Language Processing (NLP)

The aim of NLP is to produce phonetic transcription of a text together with the desired intonation and rhythm. In any written text of a language, the presence of symbols like numbers, abbreviations and acronyms are very common. Since the ways we write and speak are different, there is a need to design a mechanism that enables symbols converted to their equivalent speech forms. In addition, phonetic transcription of the text with correct prosody and pronunciation also needed and addressed in this phase even if it is difficult because of written text does not contain explicit emotions. In essence, NLP phase consists of three processing stages: text processing, Grapheme to Phoneme and prosody modeling [18, 19] .

Text Analysis is one of the most influential and complex tasks of NLP module which involves a number of modules such as preprocessing module, morphological analysis module,

contextual analysis module and syntactic prosodic parser, to enable analysis of raw text into pronounceable words [20]**.**

It is the process in which a given input text is first analyzed, normalized and transcribed into a phonetic or some other linguistic representation or converting the Nonstandard Words (NSWs) such as abbreviations, numbers, acronyms into its equivalent Standard Words (SWs**)** [21, 22, 23].

Phonetic Analysis involves investigation of the underlying pronunciation elements of a word. The module can be organized in many ways, often roughly classified into dictionary based and rule based strategies [17]. In dictionary based approach, a maximum of phonological knowledge is stored in a lexicon in limited boundary. In rule-based approach, a set of letters to sound or grapheme to phoneme rules are generated to handle exceptions or unlimited words. Thus, a combination of the lexicon and rules is used in most text to speech systems. Primarily, a lexicon is used to store pronunciations, and rules are used as a resort for missing entries in the lexicon [24].

Prosody Generation refers to extraction of supra-segmental features such as intonation, stress, and timing from written text [25]. These features are considered to be the melody, rhythm, and emphasis of the speech at the perceptual level. The intonation tries to estimate how the pitch pattern or fundamental frequency changes during speech. The term prosody refers to certain properties of the speech signal which are related to audible changes in pitch, loudness, and syllable length. The following can be mentioned as factors for contributing to the prosodic features: feelings (anger, happiness, and sadness), speaker characteristics (gender, age, and dialects), the meaning of the sentence (neutral, imperative, and question) and fundamental frequency (duration and stress).

2.4.2 Digital Signal Processing

Digital signal processing (DSP) module receives input from the NLP module which produces a phonetic transcription and the desired prosody of text that is read, and transforms the symbolic information it receives into speech. The input of traditional speech synthesis systems is a phonetic transcription with its associated prosody. The input can also include the original text with tags; this may help in producing higher-quality speech.

Speech generation is the production of acoustic output from phonetic and prosodic information [25]**.** The preprocessed and analyzed phonetic representation of phonemes with correct intonation, duration and stress are used to generate speech. Speech can be well characterized by its prosody. The rate of vibration of the vocal folds is called fundamental frequency, which is denoted by *f0*. The term pitch is used for the rate of vibration that is perceived by the listener. Prosody is simply the way we speak out words and this depends on ones feeling at the time of speaking. For example, when someone speaks out the word **Come** or "**Kootu"** the sound will be different depending on whether he/she is angry or not. The pronunciation is provided as a list of phones, a syllabic structure and lexical stress. The method for locating the pronunciation of a word is either by a lexicon or by letter to sound rules [26].

For example, the word five "**shan**" is changed into phonetic sequence **|s|, |h|, |a|, |n|.** So, in order to describe the correct pronunciation some kind of symbolic presentation is needed, which is the phonetic analysis. This step is also called grapheme to phoneme conversion (G2P). After the text and phonetic analysis are finished, the next step is prosodic analysis.

The task of finding the correct prosodic features is a very challenging problem. It is a must to do it seriously to get intelligible and natural sounding synthetic speech. During prosodic analysis, appropriate prosodic features will be attached to the phonetic sequence. Finally, the speech synthesis component takes the phonetic sequence along with the prosodic information from the fully tagged phonetic sequence to generate the corresponding speech waveform. Figure 2.3 shows the process for **Kootu** word.

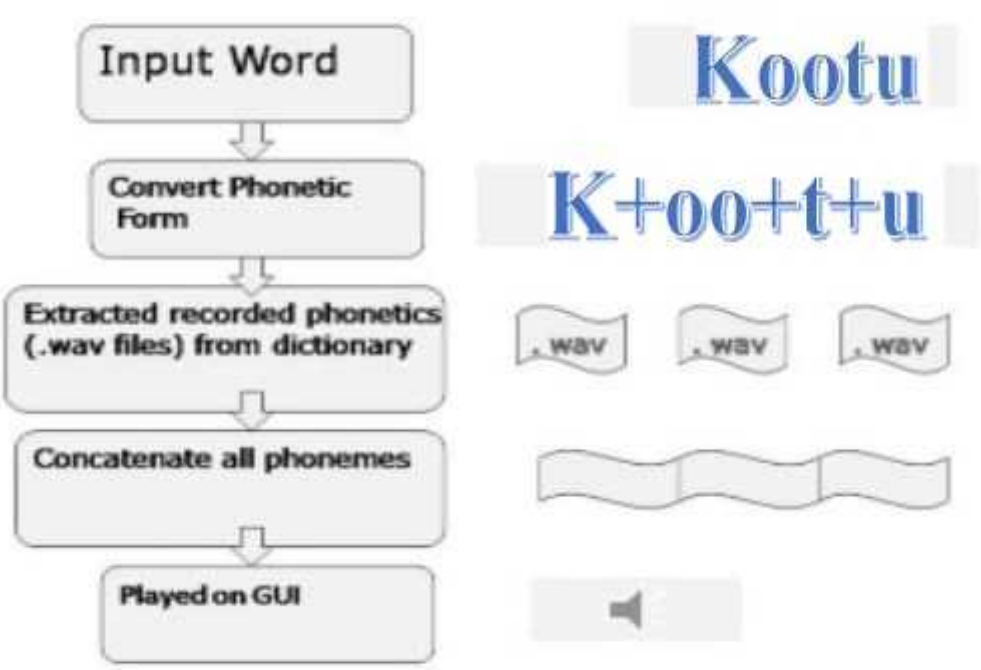


Figure 2. 3: *The flow process for 'Kootu'.*

2.5 Speech synthesis methods

There are different kinds of synthesis methods that can be used to build a TTS synthesis system. Some of these methods require a set of rules to drive the synthesizer whereas others depend on parameters from the recorded speech corpus [27].

2.5.1 Formant synthesis

Formant synthesis is one of the oldest synthesis techniques also known as rule-based synthesis. It is based on a set of parameters to create speech, such as formant frequencies, amplitudes, and bandwidths as well as fundamental frequency, voicing, and amount of aspiration noise [28]. The rules are often created manually by human experts, which often also require laborious trial and error work by comparing the original and synthesized utterances to optimize the rules. The naturalness of formant synthesis is generally poor due to the limited set of rules constructed by human experts and due to the overly simplified synthesis techniques. Theoretically, formant synthesis can produce high quality synthetic speech by creating a synthetic speech sample sounding. However, in practice, it is difficult to build a comprehensive set of rules in order to yield high quality TTS synthesis.

2.5.2 Articulatory synthesis

Articulatory synthesis aims to model the speech production mechanism as accurately as practically possible. It is motivated by how the human articulators such as vocal tract, nasal tract, lungs and larynx generate speech [27]. Theoretically, this method is the best synthesis method as it focuses on the human speech production system. However, it is the most difficult to implement and it is computationally very expensive [28]. This method is not being given much research attention at the moment, but is more of a historical treasure.

2.5.3 Concatenative synthesis

Concatenative synthesis provides a different approach to speech synthesis. Instead of artificially generating speech, a prerecorded speech corpus is first split into (small) speech segments, which are then concatenated smoothly in the synthesis phase to generate new sentences. It also provides high-quality speech output, but it may often suffer from audible (glitches) in the output due to imperfect concatenation of the units [27]. Currently, concatenative synthesis is categorized into Diphone and Unit selection speech synthesis [16]. In diphone synthesis, a minimal speech database is constructed that contains all the diphones (phone-to-phone transitions) occurring in a language. However, diphones synthesis often suffers from anomalies when concatenating two diphones that are not compatible with each other [27]. The naturalness of diphone synthesis can also suffer from artifacts stemming from signal processing methods, or modify the diphones in order to aid the concatenation [16].

Unit selection synthesis is the most widely used concatenative speech synthesis method. It can provide a highly natural and intelligible speech. However, the concatenation points may still cause audible glitches or anomalies in prosody, especially when using smaller corpora [27]. Despite the high quality, only a single speaking style, usually read-aloud, can be produced using one extensive speech corpus. Sampling-based approaches, such as unit selection, are inherently inflexible and limited by the available samples in the database, which limits the ability of the system to change, for example, voice quality, speaking style, or expression. Designing and recording a unit selection corpus that includes all the desired variation is highly impractical and expensive [29].

2.5.4 Statistical parametric speech synthesis (SPS)

The HMM-based speech synthesis is a statistical parametric model that extracts speech parameters from the speech database, trains them and produces the sound equivalent of the input text [30]. This method has the advantage of being able to synthesize speech with various speaker characteristics, speaking styles, emotions, and still produces reasonably natural sounding synthetic speech [13]. The decision tree clustered context-dependent HMMs are utilized for modelling the time varying speech parameters, and the SPS is sometimes called HMM based speech synthesis [8].

Unlike the unit selection method, SPS is able to generate speech that is not included in the original corpus by predicting the parameter values for a new context. It is also flexible in the sense that can be adapted to a different voice quality, speaking style, or speaker identity by using a small amount of corresponding speech material. In addition, it does not require as large a speech database as the unit selection methods, and the footprint is very small. However, due to the parametric representation of speech, SPS suffers from lower segmental speech quality than unit selection synthesis [1].

2.6 Text-to-Speech Synthesis System Characteristics

The purpose of this section is to outline properties or characteristics that are key to the understanding of TTS synthesis systems. The characteristics include: naturalness and intelligibility, font-end and back-end, speaker dependent and independent, limited and open domain.

2.6.1 Naturalness and intelligibility

There are two key properties which are expected of any TTS synthesis system. One such property is naturalness which is the degree to which the synthesized speech sounds close to speech uttered by humans [27] . The other is intelligibility which is the degree of ease with which people understand the synthesized speech. Understandability is sometimes used in place of intelligibility [27]. These properties can also be complemented by three other concepts called flexibility, pleasantness, and similarity to original speaker.

Flexibility has to do with how well the system handles symbols which need translation. For example, time phrases, out-of vocabulary words and others. Pleasantness on the other hand, deals with the desirability and pleasure that one associates with listening to the synthesized voice sound. Similarity to original speaker deals with how close the synthesized voice compares to that of the original speaker.

Table 2.2 compares the speech synthesis methods [14] .

Table 2. 1: *A comparison of speech synthesis methods*.

Synthesis methods	Criterion			
	Intelligibility	**Naturalness**	**Memory/CPU/Voices**	**Expressivity**
Rule_based	High	Low	High	Low
Diphone concatenation	High	Very low	High	Low
Unit selection method	High	High	Very low	Very low
SPS based on HMM	High	Medium	High	Medium

2.6.2 Front -end and back-end

A speech synthesis system is an electronic system that receives typed or stored text as input and produces or synthesizes the corresponding speech waveform as output. A TTS synthesis system trained for a particular language can be used to synthesize arbitrary text in that language. It is a common practice to view TTS synthesis systems as consisting of a front-end and a back-end. The former is responsible for text analysis, where by tokenization converts ambiguous symbols like dates to their equivalent word format, and then these words are also assigned their corresponding phonetic transcriptions by a process called grapheme-to-phoneme conversion [27]. The latter is responsible for converting the output (phonetic transcriptions and prosodic information) of the front-end to the corresponding waveform rendition.

2.6.3 Speaker-dependent and speaker independent systems

Speech synthesis systems can be trained for speaker-independent, speaker dependent or adaptive platforms [27]. A speaker-dependent system is one that is trained on data from one particular speaker. A speaker-independent system on the other hand is trained on data from several speakers and can be used to synthesize text using any of the trained voices. An adaptive TTS synthesis system is one that allows a new speaker to be adapted based on trained data of a speaker-independent system hopefully using only minimal data from the target speaker [27].

2.6.4 Limited domain and open vocabulary

TTS synthesis systems can be developed for either limited domain or open vocabulary platforms [27]. Limited domain speech synthesis systems are those trained using data from a particular domain (e.g., medicine) to be used only for purposes relating to that domain [27]. Moreover, limited domain systems do not require a huge database, they cannot synthesize words not in their lower case database.

An open vocabulary TTS synthesis system is one that is trained on general purpose data from a particular natural language to be used for general purpose applications [29]. Unlike limited domain systems, open vocabulary systems are flexible in that they can synthesize even words not in their database. However, open vocabulary systems, often require a huge database, more training data, and produce less natural speech than that produced by limited domain systems.

2.7 Fundamental concepts of statistical parametric speech synthesis

SPS is one of the most widely used speech synthesis technologies today[8]. The flexibility and robustness makes an attractive method for almost any speech synthesis application. Using the parametric representation of speech and the linguistic information extracted from text, the context-dependent statistics of the speech sounds can be modelled. Statistical parametric synthesis is simply described as speech synthesis method that enables generating an average of some set of similarly sounding speech segments via using parametric models [31]. Statistical parametric synthesis method differ from the traditional parametric method by using machine learning techniques to learn the parameters and any associated rules of co-articulation and prosody from speech data [32]. This is to mean that statistical parametric

speech synthesis method produces speech from a set of parameters learned from a speech data given as input while the traditional parametric synthesis method performs parameters that require manual specification and hand-crafting, whereas statistical parametric synthesis method implements machine learning models**.** Moreover, SPS has the ability of synthesizing speech with various voice characteristics such as speaker individualities, speaking styles, emotions and the like that the other two methods cannot [33]. In traditional parametric synthesis methods, the parameters are derived manually and rules are prepared manually to incorporate co-articulation and prosody. Even though, the new synthesis statistical parametric synthesis methods differ from the traditional parametric by using machine learning techniques to learn the parameters and any associated rules of co-articulation and prosody from the data. One of the early works on statistical parametric synthesis is based on Hidden Markov Models (HMMs) [12].

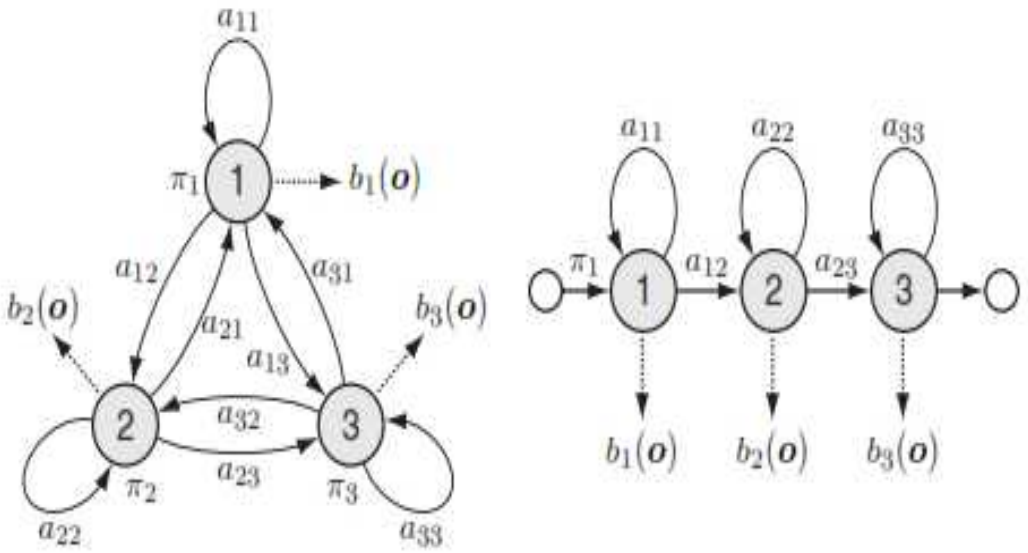
2.8 Machine learning models used in SPS

Machine learning is to learn from data in an automated fashion (ideally without human effort) to build a model that can identify patterns and make accurate judgments. In other words, Machine Learning is the branch of computer science that aims to the development of effective computer programs that are solely operates based on the information extracted from the data. Machine Learning is the subfield of computer science that gives the computer the ability to learn without being explicitly programmed.

2.8.1 Hidden Markov Models

The HMM is the most powerful statistical tool models under statistical parametric speech synthesis method [29]. A HMM can be described as a finite state machine which generates a sequence of time observations. At each time unit (i.e., frame), the HMM changes states according to state transition probability distribution, and then generates an observation $\boldsymbol{o}_t$ at time t according to the output probability distribution of the current state [34]. Hence, the HMM is a doubly stochastic random process model. The HMM is one of the statistical models that is used in TTS synthesis systems for modeling the features of speech signals. A time observation is generated by first making a decision to which state to proceed, and then

generating the observation according to the probability density function of the current state. The system modeled by an HMM is assumed to be a Markov process, in which the probability of a state transition depends only on the path of the past states. This characteristic is called the Markov property [34]. Figure 2.4 shows types of HMM [30].



(a) Ergodic (full connected) **(b)** Bakis (left-to-right)

Figure 2. 4: *Types of hidden Markov models*

In the Figure 2.4, figure (**a)** shows 3-state ergodic model, in which every state of the model could be reached from every other state of the model in a single step, and figure (**b)** shows a 3-state left-to-right model, in which the state index increases or stays the same as time increases. In speech processing, the left-to-right models are often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successively change.

Each state has a probability distribution over the possible output observation. Therefore, the sequence of discrete time observations generated by an HMM gives some information about the hidden state sequence, the parameters of the model may be known. An *N*-state HMM is defined by state transition probability distribution.

$\mathbf{A} = \{ a_{ij} \}_i^N{}_j = 1$ State transition probability matrix……..*(1)*

$\mathbf{B} = \{b_j (O)\}\ j^N = 1$ Observation probability distribution ……..... *(2)*

$\mathtt{m} = \{ \Pi_i \}_i^N = 1$ Initial state distributions ……..…… (3)

Where A**,** state transition probability matrix (a_{ij}) i and j are axis, B; observation probability distribution; O is output probability distribution either discrete or continuous observation; Π; initial state distributions N; number of hidden states. For convenience, the compact notation $\lambda = (A, B, \Pi)$ is used to indicate the parameter set of the model. Figure 2.5 shows the typical architecture TTS using HMM [35].

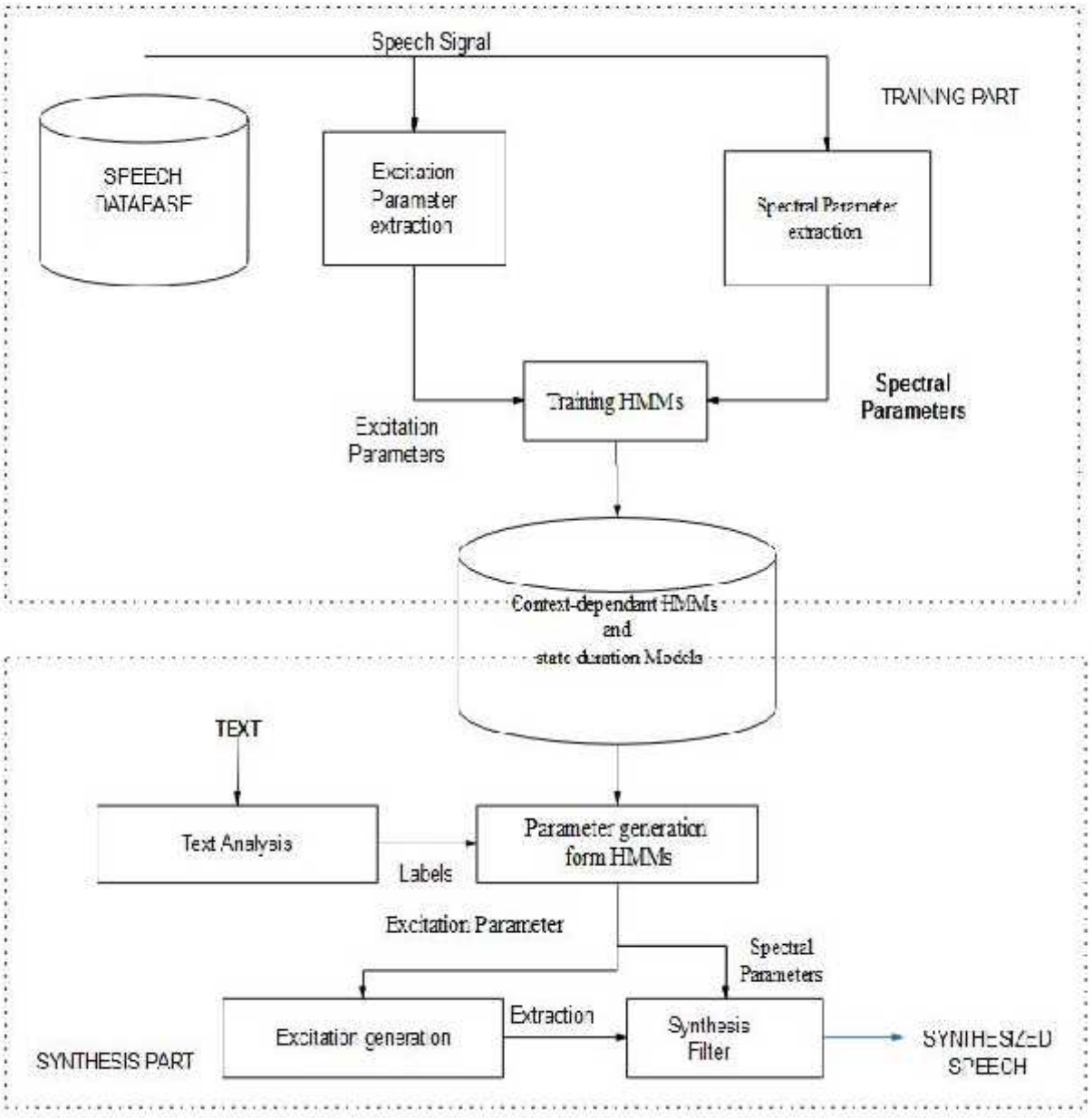


Figure 2. 5: *Typical text to speech using Hidden Markov Model [35]*.

From the Figure 2.5, HMM-based speech synthesis consists of training and synthesis parts. In the training part two parameter feature vectors are extracted, the spectrum and excitation. Before going to extract the parameter features from the raw speech, first we have to prune the given data to erase noise and generate the labelled speech from the raw speech database.

The spectrum parameters has a mel frequency cepstrum coefficients (MFCC) whereas the excitation parameter has of the fundamental frequency (F0, source) [36]. The MFCC are used as spectrum parameters and the fundamental frequency is used as a source of speech [37]. Features are used to define, train the HMM state and are categorized into two streams of data.

The first stream contains the spectrum part along with the delta and delta of delta values, and the second stream includes the logarithm of the fundamental frequency (logF0) along with their delta, and delta of delta values respectively. Each stream of information is modeled separately and delta and delta of delta values are used to model the dynamic nature of the speech. The Multi Space Distribution Hidden Markov Model probability density (MSD-HMM) is used to model for both discrete values (texts) that has a single value and continuous values (speech) to generate noise free fundamental frequency data [38]. Finally, using the Speech Signal Parametric Toolkits (SPTK) parameter features vectors are extracted. Here, the delta and delta of delta values of the MFCC are calculated as follows.

$$\Delta c_t = \frac{c_{t+1} - c_{t-1}}{2} \qquad \textit{Spectrum part delta.....4}$$

$$\Delta^2 ct = \frac{1}{4} c_{t-1} - 1 - t + \frac{1}{4} ct + 1 \qquad \textit{Spectrum part delta of delta}..... 5$$

Where C_t is static feature vector for mel-cepstrum coefficient at time unit (frame) t and Δc_t, Δ^2 are dynamic features i.e.., delta and delta-delta mel-cepstrum coefficients, at each time unit **t** (frame).

Similarly, delta and delta-delta values of the logarithm of F0 are calculated by:

$$\Delta p_t = \frac{pt+1 - pt\_1}{2} \qquad \textit{Pitch part delta formula.....6}$$

$$\Delta^2 p_t = \frac{1}{4} p_{t\_1} - \frac{1}{2} p_t + \frac{1}{4} p_{t+1} \qquad \textit{Pitch Part delta of delta formula.....7}$$

Where P_t is the static feature vector for logarithm of F0, at time t and delta and delta of delta are dynamic features. Hence delta and delta-delta mel-cepstum are used as constraints to obtain a smooth sequence of mel-cepstrum vectors. The approach to generate speech from mel-cepstrum is similar to vocoder,

The mel-cepstrums are passed Mel Log Spectral Approximation (MLSA) and are excited with white noise or pulse train to generate the speech signal. Fundamental frequency (log F0) can also termed as vocal source, the rate of vibration of the vocal folds. Mel frequency cepstral coefficients (MFCC) are spectrum (vocal tract) features that extract an acoustic signal before any training. MFCC are coefficients that have strong correlation with the human vocal tract physical state and from this physical state it is possible to determine which phone is being uttered. In the synthesis phase first, an arbitrarily given text to be synthesized is converted to a context-dependent label sequence, and then a sentence HMM is constructed by concatenating the context dependent HMMs according to the label sequence. Second, state durations of the sentence HMM are determined based on the state-duration distributions. Third, the speech parameter generation algorithm generates sequences of spectral and excitation parameters that maximize their output probabilities under the constraints between static and dynamic features [37].

Lastly, a speech waveform is synthesized directly from the generated spectral and excitation parameters using a speech synthesis filter by MLSA [39, 40]. Therefore, HMM-based speech synthesis is very flexible and is able to achieve various speaker characteristics, varying emotional and speaking styles, and articulatory control based on our prior knowledge. For instance, the initial probabiity of phoneme **KON**= 1.0**%**, **KO**=0.0**%, LA=0.0%, TA= 0.0%** found on database respectively. So having this mathematical probability estimation, we choose the first letter that have highest probability **KON** as initial.

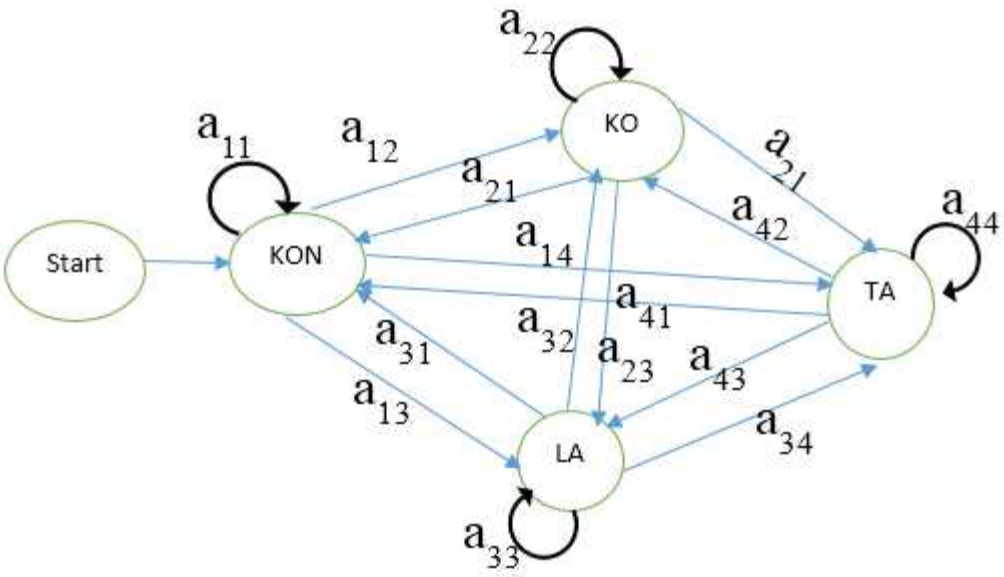


Figure 2. 6: *HMM State for the word "KONKOLATA ".*

	KON	KO	LA	TA
KON	**0.3**	**0.9**	**0.6**	**0.5**
KO	**0.1**	**0.5**	**0.8**	**0.4**
LA	**0.3**	**0.2**	**0.7**	**0.8**
TA	**0.2**	**0.5**	**0.6**	**0.7**

By using Bayes rule we can calculate the probability of each phoneme concatenation and choose the highest confidence probability as updated HMM state for the next sequence and continues by follow the above step until the full concatenation of grapheme with the language grammatical structure i.e., KONKO, *KONKOLA*, *KONKOLATA.*

Typically SPSS system consist of training and synthesis stages.

At training stage, acoustic (real-valued) and linguistic (discrete) features sequences are extracted from the speech waveforms and its transcription respectively [41]. Then the acoustic model is trained to model the conditional distribution of an acoustic feature sequence given a linguistic features as following formula.

$$\hat{\lambda} = arg\,\max_{\lambda} P(o|l, W) \quad \text{..... (8)}$$

Where O is acoustic feature sequence l is the linguistic features and λ is the acoustic *model.*

In equation (8), the fundamental problem that needs to be solved in corpus-based speech synthesis, i.e., finding the most likely speech parameters **O** for a given word sequence W using the training data **O**, and the corresponding word sequence W. We can solve this maximization problem by using Bayesian speech parameter generation algorithms.

Bayesian Approach: Bayesian learning is used to estimate posterior distribution of model parameters from prior distributions and training data [42, 41]. Bayesian learning techniques applied to HMM- based speech synthesis to determine the observation (hidden) O as following formula:

Bayes Rule

$$P\left(\frac{A}{B}\right) = \frac{P(B|A)\cdot P(A)}{P(B)} = \frac{P(B|A)\cdot P(A)}{\sum_i P(B/Ai)P(Ai)} \quad \text{.....(9)}$$

At synthesis at text to be synthesized is first converted to corresponding linguistic feature sequences is predicted from the trained acoustic model [41] as follows

$$\hat{O} = arg\,\max_{O} P(o|l,\ \hat{\lambda}) \quad \text{..... (10)}$$

, which are similar to maximum likelihood based speech parameter generation algorithms

Finally, a speech waveform is rendered from the predicted acoustic feature sequence using a vocoder.

2.8.2 Clustergen Synthesizer

Clustergen is one of the statistical parametric speech synthesis where its framework is integrated within festvox voice building [43]**.** It is used as back-end system at synthesis time by integrating with the festival speech synthesis [44]. In addition, clustergen synthesizer is well integrated in festvox voice building tool than HTS [44]. Unlike HTS that is state-based, it predicts frame-based MFCCs clustered using phonetic, metrical, and prosodic contexts.

A technique of clustering via Classification And Regression Trees (CART) uses to optimize the standard deviation of the frames in the cluster. Since, clustergen synthesis models are

stored as classification and regression trees of the phone state, each phone is realized as a left-to-right Markov chain of three states. The intermediate nodes of the tree are questions about phonetic and other high levels of contextual information. Figure 2.6 shows the typical diagram of clustergen synthesizer [18]. In addition, Classification used to provide to predict the discrete value (text) and regression is used to predict a continuous value (speech). The tree is learned from the labelled data, using supervised learning. Creating a CART model involves selecting input variables and split points on those variables until a suitable tree is constructed. Festival uses CART trees for predicting values from feature vectors. The machine learning algorithm used for training CART trees is a program named *wagon*. *Wagon* is a part of the publicly available Edinburgh Speech Toolkit, and is integrated into the clustergen component of festival used for the experiments.

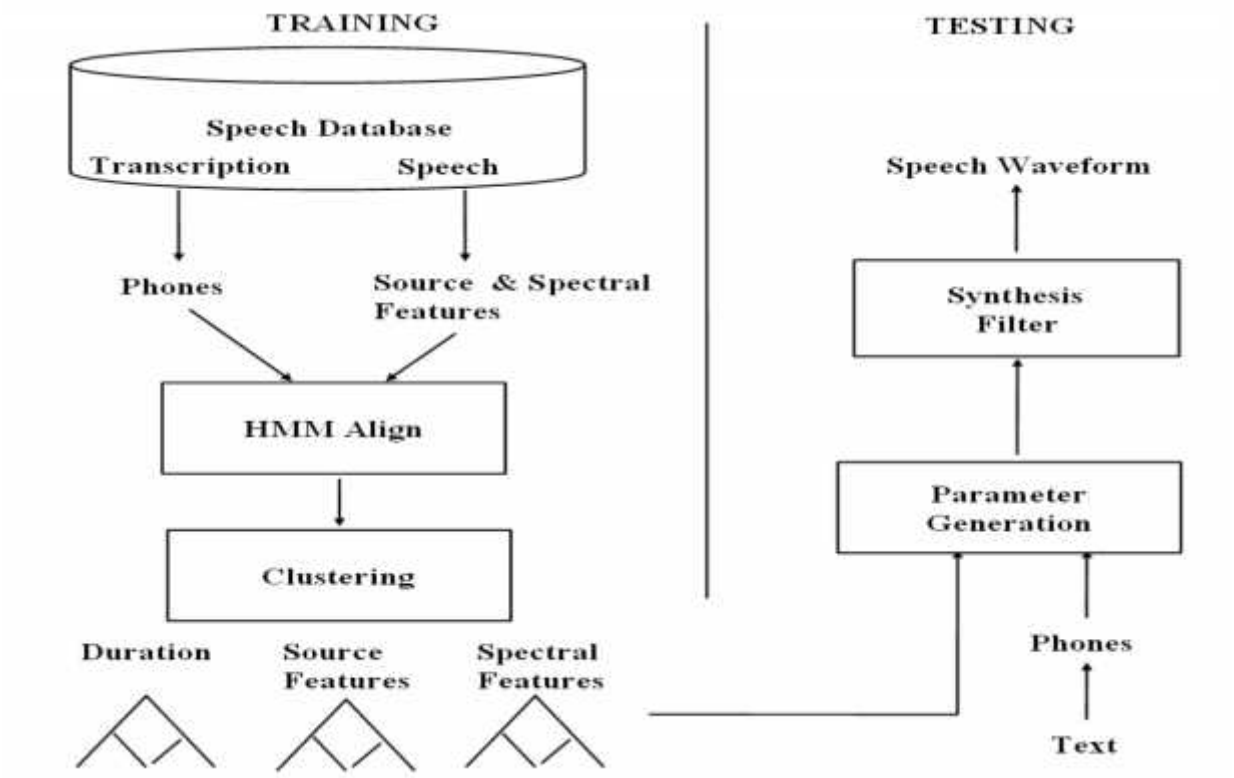


Figure 2. 7: *Typical Clustergen Synthesizer*

2.9 Synthesis Model or Source filter Model

2.9.1 Linear Prediction Code (LPC)

Linear predictive code is a tool used mostly in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form using the information of a linear predictive model [36].

It is one of the most powerful speech analysis techniques, and one of the most useful methods for encoding good quality speech at a low bit rate and provides extremely accurate estimates of speech parameters. There are various advantages for the use of LPC such as it produces better approximation coefficient spectrum, gives shorter and efficient calculation time for signal parameters and able to get important characteristics of the input signals [27]. LPC is a method used to represent and analyze human speech [40]. The idea of coding human speech is to change the representation of the speech. Representation when using LPC is defined with LPC coefficients and an error signal, instead of the original speech signal. The LPC coefficients are found by LPC estimation which describes the inverse transfer function of the human vocal tract [9]. Figure 2.7 shows the source of filter models or synthesis procedure [45].

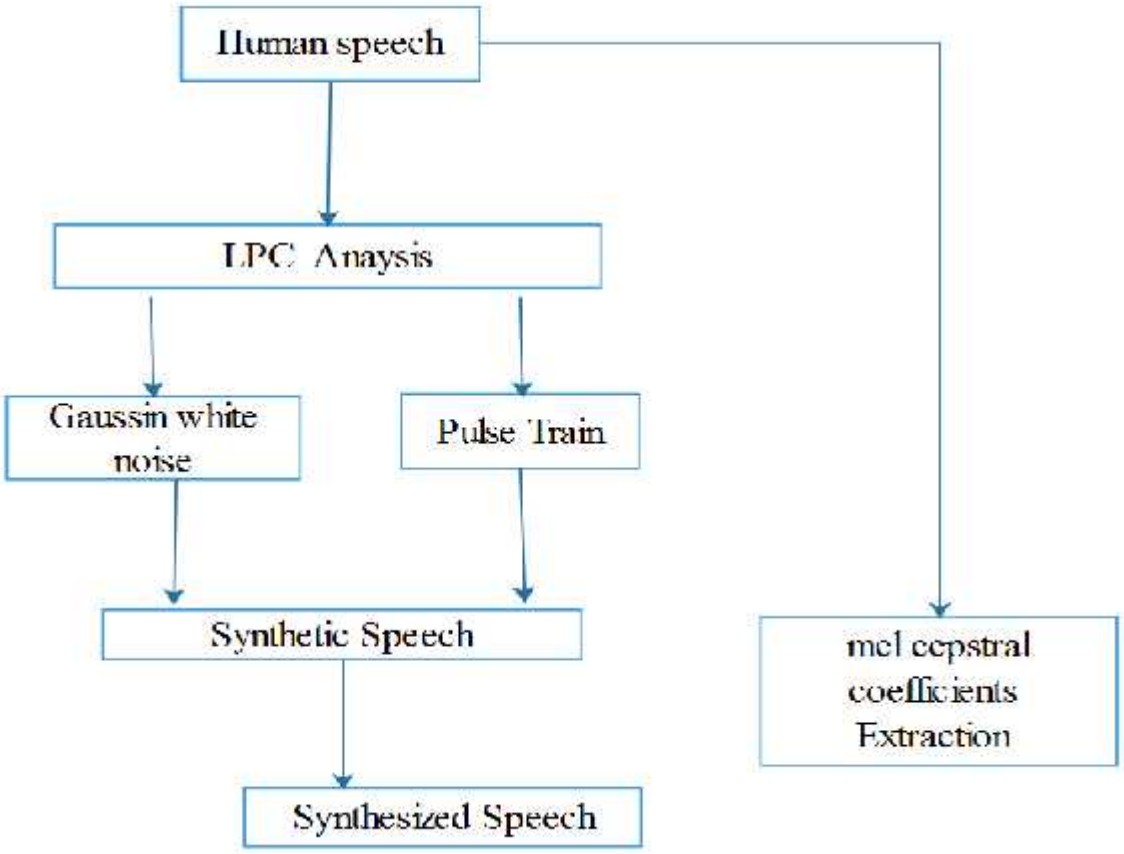


Figure 2. 8: *Source-filter model of speech.*

2.9.2 Excitation Model

The most basic excitation model uses periodic pulses with fundamental frequency *F0* voiced speech signal, and white noise to represent unvoiced signal. The voiced and unvoiced are differentiated by the short term energy frame to frame. The excitation model is improved with adding more features; one of the useful models is the mixed excitation model [45, 40], with voiced and unvoiced decision set according to different pass band.

Mixed Excitation

In order to limit the vocoded effect on synthesized speech, improvements are made in the Mixed- Excitation Linear Prediction (MELP) vocoder [45]. Mixed excitation model is based on MELP and requires more spectral parameters to easily incorporate into the HMM-based TTS system because its parameters are all trainable. In the analysis process, beside pitch periods and white noises, the mixed excitation model also includes the mel cepstral

coefficients presented previously as static features and its delta coefficients as dynamic features.

In the synthesizing stage, instead of directly applying the inverse synthesis filter to the sum of the white noise and pulse train, the periodic pulse train represented voiced signal and Gaussian white noise represented voiced signal are filtered with a band pass filter to determine the frequency band of voiced and unvoiced signal [45, 40].

The voiced and unvoiced decision of the pass band is determined by the voicing strength which is estimated with correlation function. The vocoder uses aperiodic pulses in the transition from voiced to unvoiced, and periodic pulses for elsewhere in voiced speech. The entire frequency band of the signal is evenly divided into four frequency pass bands from 0 to 8000 Hz [45]. The synthesized speech is obtained by applying inverse synthesis filter to the mix of filtered pulses and noise excitation.

2.9.3 Mel Cepstral Analysis

Cespstrum is defined as the inverse fourier transform of the logarithm of the spectrum, and excitation offers the advantages of low spectral distortion, low sensitivity to noise and efficiency in representing log spectral envelop [45]. In mel cepstral analysis, the log spectrum is non-uniform spaced in frequency scale [45]. Mel cepstral coefficients can be derived from LPC coefficients but with non-linear transfer function. Unlike the LPC model cares only filter poles, the mel cepstral model includes both poles and zeros.

Mel Scale Approximation

Mel cepstral analysis uses logarithmic spectrum on mel frequency scale to represent spectral envelopes and provide extra accuracy [40]. The mel frequency scale has a typical that it will expand the low frequency part and squeeze the high frequency part of the signal. Human ears have non-linear perception of frequency of sound, and are more sensitive to low frequency than to high frequency. Therefore, mel frequency scale is more effective than linear frequency scale.

Mel Cepstral Adaptive Method

The desired mel cepstral coefficients minimize the spectral criterion estimating log spectrum in the mean square sense. The quality of the synthesized signal is optimized by minimizing

the value of the unbiased estimator of log spectrum. Mel cepstrum is derived from LPC, and Mel cepstral coefficients could be calculated from LPC coefficients with recursive method. The system is best represented with a mel log spectrum approximation (MLSA) filter because of its low sensitive to noise and fine quantization characteristics [39]. MLSA filter has an all-pass transfer function as follows: Because of the nonlinearity of the transfer function, the system cannot be directly implemented with a regular filter. Estimation of log spectrum involves non-linearity which could be solved by the iterative algorithm. The best results are achieved with the MLSA filter, whose coefficients are obtained by a simple linear transform from the mel spectrum. This filter has two main advantage; it is a very low coefficient sensitivities and a good coefficient quantization characteristics.

2.10 Afaan Oromoo

Afaan Oromoo is a member of the Cushitic branch of the Afro-Asiatic language family which was the third most widely spoken language in Africa, after Hausa and Arabic [46]. Presently, it is an official language of Oromia Regional State which is the biggest region among the current federal states in Ethiopia.

2.10.1 Afaan Oromoo Writing System

Afaan Oromoo uses Latin based script called **Qubee** which has been adopted and became the official script of Afaan Oromoo since 1991 [47]. According to [46], Afaan Oromoo is a phonetic language, which means that it is spoken in the way it is written. Unlike English or other Latin based languages, there are no skipped or unpronounced sounds or alphabets in the language. Even though Afaan Oromoo is Latin based script language, its structures of writing and grammar like normative, case, genitive, dative, locative, and definiteness are different from other Latin languages like English, French, German ,Turkish and etc. The grammatical system of Afaan Oromoo language is quite difficult and exhibits many structures common to other languages.

2.10.2 Afaan Oromoo Alphabets: Vowels and Consonants

The "***Qubee Afaan Oromoo***" writing system has a total of 33 letters that consist of all the 26 English letters with an addition of 7 combined consonant letters which are known as "***Qubee Dachaa"*** [48]: ***ch, dh, sh, ny, ts, ph*** and ***zy***. All the vowels in English are also vowels in Afaan Oromoo"***Qubee***". The vowels have two natures in the language and they can result in different

meanings. A vowel is said to be short if it has one vowel and if it has two, which is the maximum, then it is called long vowel. For example, consider the words ***lafa*** (ground), ***laafaa*** (soft). In a word where a consonant is doubled, the sounds are more emphasized. For example, ***dammee*** (sweety), dame (branch).

2.10.3 **Classification of speech sounds**

Speech sounds can be classified into voiced and unvoiced categories based on components of the speech production mechanism [49]. The simplest classification divides speech units into two groups: vowels and consonants. Vowels are voiced sounds that are produced by unlimited airflow in the vocal tract. Consonants can be either voiced or unvoiced produced in the vocal tract [49]. It can be further divided to the place and manner of articulation or voicing [1]. Table 2.1 shows phonetic representation and Afaan Oromoo consonants and its International Phonetic Alphabets (IPA) equivalences [9].

Table 2. 2: *The phonetic representation of Afaan Oromoo consonants and its IPA equivalences.*

Manner		Bilabial	Labiodental	Alveolar	Palatal	Velar	Glottal
Stops	Voiced Voiceless Ejectives Implosive	B (p) Ph/p/		D T x/t/ dh / /		G K q/k /	 ‘ / /
Fricatives	Voiced Voiceless	(v)	 F	(z) S	(zy) Sh/s’/		 H
Affricative	Voiced Voiceless Ejective				J Ch(c’) C		
Nasals		M		N		Ny/ /	
Lateral				L			
Rhotic				R			
Semi-vowel		W				Y	

2.10.4 Afaan Oromoo Word Structure and boundaries

Afaan Oromoo and English are different in sentence structuring. The unique word order in Afaan Oromoo sentences is **Subject – Object – Verb** [46]**.** Modifiers, pronouns, articles, and case markers follow the nouns they modify. For instance, in the Afaan Oromoo sentence "*Dinqan bilisa bahe*", "*Dinqan* "is a subject, "*bilisa*" is an object and "*bahe*" is a verb. The translation of the sentence in English is "*Dinqa* has got freedom" which has SVO structure. There is also a difference in the formation of adjectives in Afaan Oromoo and English [50].

In Afaan Oromoo, adjectives follow a noun or pronoun and their normal position is close to the noun they modify while in English, adjectives usually precede the noun. For instance, *namicha gaarii* (good man), *gaarii* (adj.) follows *namicha* (noun). Afaan Oromoo also uses white space to separate words from each other. Moreover, parenthesis, brackets, quotes, etc., are being used to show a word boundary. Furthermore, sentence boundaries and punctuations are almost similar to English language i.e., a sentence may end with a period (.), a question mark (?), or an exclamation mark **(!)** [51]**.**

2.10.5 Afaan Oromoo Punctuation marks

Punctuation is placed in a text to make meaning clear and reading easier. Punctuation marks used in both Afan Oromoo and English languages are the same and used for the same purpose with the exception of the apostrophe [50]. Apostrophe mark (') in English shows possession, but in Afan Oromoo it is used in writing to represent a glitch (called *hudhaa*) sound. It plays an important role in Afan Oromoo reading and writing system. For example, it is used to write the word in which most of the time two vowels appeared together like "*ba'e*" to mean ("get out") with the exception of some words like "*ja'a*" to mean "six" which is identified from the sound created. Sometimes apostrophe mark (') in Afaan Oromoo is interchangeable with the spelling "h". For instance, "*ba'e*", "*ja'a*" can be interchanged by the letter "h" like "*bahe*", "*jaha*" respectively without changing the senses of the word. Like English language, the following are some of the most usually used punctuation marks in Afaan Oromoo [52] .

"***Tuqaa***", Full stop (.): is used at the end of a sentence and also in abbreviations.

"***Mallattoo Gaafii***", Question mark (?): is used in interrogative or at the end of a direct question.

*"**Rajeffannoo"***, Exclamation mark (!): is used at the end of command and exclamatory sentences.

*"**Qooduu**"*, Comma (,): is used to separate listing in a sentence or to separate the elements in a series.

*"**Tuqlamee**"*, Colon (:): is used to separate and introduce lists, clauses, and quotations, along with several conventional uses, and etc.

2.10.6 Word Segmentation

There are different methods for separating words from each other. The word, in Afan Oromoo "*jecha*" is the smallest unit of a language. This method might vary from one language to another. In some languages, the written or textual script does not have whitespace characters between the words. However, in most Latin languages a word is separated from other words, by white space characters " " [46]. Afaan Oromoo is one of Cushitic families that uses Latin script for textual purpose and it uses white space character to separate words from each other's. For example, "*Bilisummaan Finfinnee deeme*". In this sentence the word "*Bilisummaan*", "*Finfinnee*" and "*deeme*" are separated from each other by white space character. Therefore, the task of taking an input sentence and inserting legitimate word boundaries, called word segmentation, is performed using the white space characters.

Chapter Three: Related Work

In this chapter, different works related to speech synthesis technique for any local and foreign languages would be discussed.

3.1 HMM based speech synthesis systems

Tokuda et al. [53] developed a HMM based speech synthesis system for the English language. The resulting run time engine of HTS is very small even less than one megabytes. The authors, have used HMM method using festival speech tools for text analysis and feature extraction. However, the authors did not put a quantitative analysis of the result and simply concluded that the provided result of this synthesized speech is good as compared to any other rule based speech synthesizers like, a formant and concatenative methods.

Ntsako et al. [27] developed a highly intelligible and acceptably natural sounding speech synthesis system for Republic of South Africa Xitsonga language using a hidden Markov model (HMM) speech synthesis method. The mean opinion scores used in the evaluation process ranged from one (worst) to five best. The evaluation results indicate that the overall system was found to be good on average. From twelve languages, only 25% of the subjects said that he 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. Therefore, the system received a 92.3% acceptability rate. On the contrary, the authors didn't describe the voice conversion mechanism and algorithms used during the speech generations.

Bereket Kasseye [54]developed a text to speech synthesizer for Amharic language using hidden Markov model technique. The results from the mean opinion scale (MOS) were found to be 4.12 and 3.6 for intelligibility and naturalness respectively. However, the study did not consider the factor of intonation in developing the system.

3.2 Other method based speech synthesis system

Tewodros Abebe [55] developed a diphone based concatinative text to speech synthesizer for Wolaytta language. The author found the overall performance of the system to be 78%. The naturalness and the intelligence of the system levels were 2.77 and 3.17 respectively showing a poor result. In contrast, the author did not consider the nonstandard words, and also couldn't work for large units of speech like sentence, syllables and words.

Alula Tafare [16] developed a concatinative based a text to speech synthesizer for Amharic language. The author was done the possibility of developing a generalized approach in combining standard words and nonstandard for Amharic language. The overall performance found is 73.35%. According to mean opinion score metrics 2.83 and 3 out of 5 were the naturalness and the intelligibility of the system respectively. However, the author recommendation better to shift from rule based approach to statistical parametric based approach having advantage of limited storage and small run time computations.

Agazi kiflu [4] developed a unit selection based text to speech synthesizer for Tigrinya language. The author categorized the results based on the questions, 38.8% considered the voice is very good, 58.3% said of voice is good and 2.7 % considered the voice unnatural. The overall performance is more than 97.1% which is acceptable rate. Nevertheless, a unit selection technique requires a large database often in size of gigabyte (GB), single speaker's dependents and limited domain vocabulary.

Lemlem H and Million M [21]developed concatinative based text to speech synthesizer for Tigrinya language. They developed the prototype using festival speech synthesis framework that resulted 89.76% using 15 test case utterances (user based evaluation). Their synthesizer was evaluated via user based evaluation which is prone to high error rate. All these mentioned synthesis types are concatinative speech synthesis systems and their advantages of these techniques is high quality speech but it is disadvantageous due to enormous costs and time for constructing corpora and is not straightforward to synthesize diverse speakers such as emotions, styles, database dependent and only works for limited domain.

Samson Tadesse [9] developed a concatenative based text to speech synthesizer for Afaan Oromoo language. During the process, a limited rule based diphone database entries were constructed. The author showed that 75% and 54% of words in the data set are correctly pronounced as to the diphone and triphone speech units respectively, i.e., the concatenation of large units degraded the performance of the system. Based on mean opinion score measure, the author achieved the intelligibility results, 3.03 and 2.2 rate for the diphone and triphones respectively and the naturalness of the system was 2.65 and 2.02 for each speech units respectively. However, due to the use of a rule based approach, the overall

performance of the system is became poor. Although, the work of the author has a lot of limitations such as it require huge memory, database dependent and also needs a larger data base to include all possible utterance in the language.

3.3 Summary

In this chapter, we presented the review of number of speech synthesizer developed focusing on different techniques. The techniques presented and discussed are rule based and machine learning based approaches. Hidden Markov model based learning has shown an interesting performance. Hidden Markov model based learning has the capability to capture the speech synthesis problems. For instance, the works of Ntsako [27] and Bereket [54] approve the impact of HMMs.

As described in above, there were attempts made to developing speech synthesizer for Afaan Oromoo. However, their work are based on the traditional and hand crafted approach. The work made to develop speech synthesizer for Afaan Oromoo was limited to database dependent. In contrast, statistical parametric synthesis techniques generates average speech units which are smooth and stable, stores statistics rather than waveforms and need little memory for storage. In addition, HMM techniques provides an advantages such as a parametric model, can easly modify voice characteristics, small run time, easy for speakers adaptation and integrated with small handheld devices. Even if the related works have achieved a lot of contribution in the area, speech synthesis on Afaan Oromoo language has done with HMM. The current work tried to model speech synthesizer that handles the rule based limitation. Therefore, developing a statistical parameter speech synthesis based on HMMs approach is important for the language

CHAPTER FOUR: Design of Afaan Oromoo Text to Speech Synthesizer

4.1 Introduction

This chapter deals with the Afaan Oromoo text to speech synthesis system architecture. It explains the whole process of the design, the algorithms used, their relation and interaction, the representation and description of components. The proposed architecture of statistical parametric speech synthesis based on HMM text to speech system for the Afaan Oromoo language is illustrated in Figure 4.1.

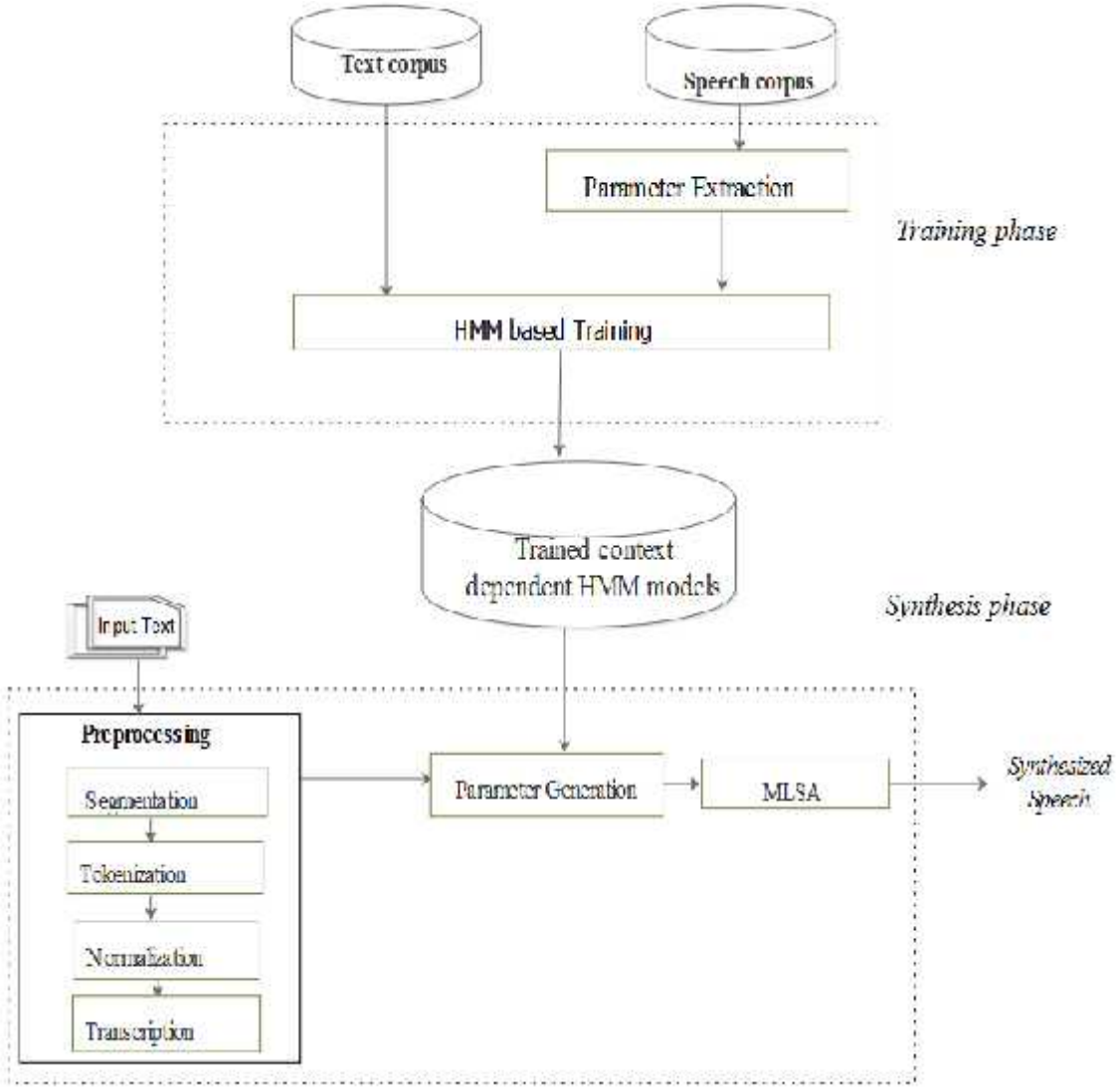


Figure 4. 1: *Proposed Architecture of HMM based text to speech for Afaan Oromoo.*

4.2 Training Phase

In the training phase, the speech and text corpus are collected and labeled according language structure further processing. The speech parameters including spectral (e.g., mel cepstral coefficients and their dynamic features) and excitation parameters (e.g., log F0 and its dynamic features) are automatically extracted from the manually labeled speech corpus using speech tools SPTK. After that the extracted speech parameters are automatically aligned with the manually labeled text to extract the utterance structure using ergodic hidden Markov model (EHMM) labeler. Next, from the labeled text and extracted parameters, the context dependent HMM-based training module are extracted and trained using the decision tree based context clustering methods. Finally, using expectation maximization (EM) algorithm, a trained voice of context-dependent HMMs and duration models are generated.

4.2.1 Labelling text and speech corpus

Labelling is the process of giving a label for each speech signal in the utterance or generates the labelled utterances. The manually labeled text and extracted speech corpus are align using text labeler EHMM to generate the utterance for each sentence. In this research, the labeled utterance are generated by speech tools using the festival to understand Afaan Oromoo letter to sound transliteration tools using automatic EHMM labeler. For example, from the given texts "sent _00002: *baajanni bara kuma lamaa fi afurii baajata bara kuma lamaa fi sadii ni caala.*" and generate the utterance structure parameters in the form of (utt) (see in Appendix A).

A phone set is a set of symbols which may be further defined in terms of features extraction such as vowel, place of articulation, manner of articulation for constants and type of vowel in which the set of features and their values must be defined. In Afaan Oromoo, there are around 39 phone sets and each has its own standard phone features including manner of articulation, height of vowel, vowel length, lip rounding, consonant sound and vowel frontless. Therefore, the concept of phone sets is very crucial to a number of different subsystems within festival.

The festival also supports multiple phone sets simultaneously and allows mapping between sets when necessary. As a result, the Afaan Oromoo letter to sound transliteration, waveform

synthesized, all require the definition of the phone set must declared before they will operate (see Appendix B)**.**

Transliteration is the mapping process from one system of writing into another word by word, or phone by phone. It is also the process of transferring a word from the alphabet of one language to another. In this study, we have written *Afaan Oromoo letter to sound (AOLTS*) transliteration rule that contains the grammar and structure of the language. The Afaan Oromoo pronunciation lexicon is mainly used for determining the pronunciations of these words. Finally, using festvox and festival provides relevant phonemes for building utterance structures for each speech unit and the *Afaan Oromoo letter to sound (AOLTS*) transliteration rule (see Appendix C).

4.2.2 Parameterization

In this work, mainly we have two parameters feature vectors: the spectrum and excitation parameters.

In order to produce natural sounding speech at low bit rate, the parameters representing speech information effectively need to be extracted from source files. The excitation signal usually requests fundamental frequency F0, spectral envelopes information and voiced and unvoiced decision parameters. The spectrum parameters consist of mel-frequency cepstrum coefficients (MFCC) and is used as spectrum parameters and the fundamental frequency is used as a source of speech.

In this thesis work, before going to extract the parameter features from the raw speech corpus, first we have to prune the given data to erase noise and generate the labelled speech from the raw speech database manually. The extracted parameters: the spectrum and excitation consists of two streams. The first stream contains the spectrum part along with the delta and delta of delta values, and the second stream includes the logarithm of the fundamental frequency (logF0) along with their delta, and delta of delta values respectively. Each stream of information is modeled separately and delta and delta of delta values are used to model the dynamic nature of the speech. Finally, using the speech signal parametric toolkits (SPTK) tools parameters features vectors are extracted.

4.2.3 Trained Parameters Models

Defining the structure and overall form of a set of HMMs is the first step towards building a synthesizer. The second step is to estimate the parameters of the HMMs from the data sequences that they are intended to model. This process of parameter estimation is usually called training. 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 (HTK). Algorithm 4.1 depicts the process.

```
Input: Text corpus and speech corpus

Output: Model trained context dependent HMM and duration

Begin

Input prepared or defined text and speech corpus

Extract the speech parameters from speech database

Align the analyzed text with its parameters

Trained and Cluster based on F0, spectrum and duration

Generate trained context dependent parameters

End if
```

Algorithm 4.1: *Algorithms for trained the HMM based speech.*

4.2.4 Constructing context dependent HMM parameters

In continuous speech, parameter sequences of particular speech unit (e.g. phoneme) can vary according to contextual factors. Hence, constructing a model for each phoneme separately is not sufficient to generate an intelligible and natural sounding speech. There are many contextual factors such as phone identity factors, stress-related factors, locational factors that affect spectrum, pitch and duration. When we construct context dependent models taking into account many combinations of the contextual factors, we expect to be able to obtain appropriate models. However, as combination of the text increase their contextual factors also increase rapidly. Therefore, it is difficult to prepare speech database which includes all

combinations of contextual factors. To address this problem, we apply a decision-tree based classification and regression technique to distribute for spectrum, excitation and state duration are clustered independently. In addition, it is used to provide to predict the discrete value (text) and regression is used to predict a continuous value (speech). It predicts the features by classifying a phoneme as a vowel or constant with its manner and place of articulation position. Classification and regression trees is a nonlinear and modelling algorithms that automatically find the features of the text based on its labelled features and utterance structure according to the duration, vocal source and its spectral parameters.

Decision trees are an important type of algorithm for predictive modeling machine learning. In addition, classification used to provide to predict the discrete value (text) and regression is used to predict a continuous value (speech). Creating a classification and regression tree model involves: selecting input variables and split points on those variables until a suitable tree is constructed.

4.3 Synthesis phase

First, an arbitrarily given text to be synthesized is converted to a context-dependent label sequence, and then an utterance HMM is constructed by concatenating the context-dependent HMMs according to the label sequence. Second, state durations of the utterance HMM are determined based on the state duration. Third, the speech parameter generation algorithm generates, the sequence of spectral and excitation parameters that maximize their output probabilities. Finally, a speech waveform is synthesized directly from the generated spectral and excitation parameters using the corresponding speech synthesis mel log spectrum approximation (MLSA) filter for mel-cepstral coefficients.

4.3.1 Preprocessing

Preprocessing is one of the most powerful and complex task in natural language processing module. In this work, the raw text is preprocessed into a context dependent label sequence by a text analyzer. These procedures of raw text preprocessing include four main tasks such as sentence segmentations, tokenization, and normalization and text transcription (G2P) conversation.

Tokenization

Tokenization is the process of segmenting and running texts into words, sentences and phonemes. Tokenization is the process of breaking sequences of sentences into its constituent words. During tokenization, the white space delimiter and special characters called hudha " ' " (diacritical) are the main focus areas where, whenever there exists the mentioned delimiters between characters in a sentence, sequences of characters are broken to produce a meaningful word for a given specific language. The festival and festvox speech tools can understand Afaan Oromoo letter to sound transcription, phone sets, and tokenizes any amount of standard words into single word by looking for the white space. For instance, the sentence 'Tolosan barata cimadha' which means *'Tolosa is a clever student', is tokenized into three tokens:* Tolosan, barata and cimadha (see Appendix A).

Text Normalization

Text normalization is the process of generating normalized word from the text containing nonstandard words. Afaan Oromoo text includes nonstandard words (NSW), which cannot be directly transcribed to phonemes, e.g. numbers, dates, abbreviations, currency, etc. These activities include, the conversion of nonstandard words into their orthographic representation by expansion into their full spoken words.

Text transcription (grapheme to phoneme)

One of the most difficult and challenging jobs of HMM approach speech synthesis is choosing the appropriate phone from multiple phones of HMM parametric generations. This is mostly determined by the grapheme to phoneme letter to sound rule conversion methods. In this thesis, the automatic labeling is one of the significant tasks to build an annotated corpus for TTS systems. To do so, after normalizing the text, the processed texts are converted into phonemic sequences using text analyzer or festival tools by integrating Afaan Oromoo phone sets and letter to sound transliteration tools (AOLTS) conversion module. Grapheme to phoneme (G2P) conversion is used for the reason that it is supported in festival system and Afaan Oromoo being a phonetic language. The process is depicted in algorithm 4.2.

```
Input: input text

Output: transcription of the text

Begin

Input the text

Define phone set and AOLTS module

Preprocessing the text

Generate transcribed text (grapheme to phoneme)

End if
```

Algorithms 4.2: *Algorithms for transcription text.*

In this study, we used the HMM based maximum likelihood parameter generation algorithms for extracting, the spectrum and excitation parameters from trained model. It is also used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. In addition, it finds the unknown parameters of a HMM and the most likely sequence of hidden states, string of text given the acoustic signal.

4.3 2 Mel Log Spectrum Approximation Filter

The synthesis filter for this thesis model has the structure of the MLSA filter. The main work of MLSA is to filter the synthesize speech from mixed excitation parameters and mel frequency cepstral coefficients extractions. Therefore, MLSA is a simple, low sensitive to noise, small spectral distortions and it has good statistical features.

Chapter Five: Experiment Result and Evaluation

In this chapter, we will discuss the developmental environment and tools, method of data collection, designing questionnaires, experimental results and system evaluations methods

5.1 The Development Environment and Tools

We used the following tools to develop the system.

- Laptop computer with Linux Ubuntu operating system, Intel core i5 with 2.5 GHz processor speed, 4.00 GB RAM and 465 GB hard disk capacity
- C++ programming language for coding purpose
- Microsoft office 2013 for documentation
- PRAAT- freely available software for sound manipulation, phonetic and acoustics analysis.
- Festival 2.5.0 current- freely available multilingual framework for building speech synthesis system
- Festvox 2.7.3 current- open source software for voice building process includes program tool such as phone set, lexicon, phrasing and etc.
- Speech tools 2.5.0-is free software for manipulating the sorts of objects used in speech processing.
- SPTK 3.6- freely available software for speech signal processing and extracting speech parameters and to re-synthesize speech from the parameters
- HTS- The toolkit is used for implementing HMM-based speech synthesis.
- Microphone for recording speech data.
- Adobe audition CS6 – freely available software for displaying and labelling the speech waveform.
- Edraw max 8.4 – a software for drawing different designing activities

5.2 Data Collection

To build Afaan Oromoo speech database, first phonetically balanced sentences are collected. These sentences are collected from news, blogs, stories, political essays, literary works, sports sections, magazines, holy books, proverbs, and newspapers. In Afaan Oromoo, there are 26

pure phonemes and 7 borrowed phonemes are available. For the evaluation, ten sentences are selected randomly out of the trained dataset.

The recording process took place in an office environment with minimal noise. The speech is recorded at noise free studio using a PC at ***Oromia Broad Casting Networks (OBN) Biiro at Adama*** with female journalists. The speech sample was recorded at 44.1 KHz stereo and the files are stored in waveform format (.wav). The waveform files are normalized and changed to conform to 16 KHz, 16 bit, RIFF format as required by the festvox system and to make it easier to create raw files of small sizes. Praat is the software that is used to record the speech corpus. A regular microphone and a normal office computer made up the hardware equipment used to record the speech corpus.

5.3 Preparing Questionnaire

Before going to the evaluation, a questionnaire is prepared that comprises questions that focus on the intelligibility and naturalness of the synthesized speech. The first question is targeting in measuring the intelligibility of the synthesized speech and the second is aimed at measuring whether the synthesized speech is human like or not. For both testing methods the same questions are used (see Appendix D).

5.4 Evaluation Results and Analysis

The synthetic speech is assessed depending on its naturalness and intelligibility using subjective (MOS) and mel cepstral distortion (MCD) as objective terms.

Subjective Evaluation Metrics

The mean opinion score (MOS) test is chosen for this evaluation, which allowed us to score and compare the global quality of TTS systems with respect to naturalness and intelligibility.

MOS is a technique that indicate their assessments on a scale of bad (1) to excellent (5). Four hundred sentences are used for training and ten sentences for testing. Appendix E shows testing sentences. Five men and five females' native speakers of the language are randomly selected to evaluate the system.

The participants are allowed to listen to the recorded voice samples before they check the developed text to speech synthesizer. Subsequently, each participant plays the sample voice to check the quality of the voice. However, in order to make the test effort easy and understandable by the participants, a questionnaire is delivered beforehand to familiarize with it. Then, after listening the synthetic speech from the system, the participants are requested to fill marks on the questionnaire properly. Finally, according to mean opinion score, the mean and standard deviation cumulative results are calculated as per the respondents' responses. Table 5.1 shows the result.

Table 5. 1: T*he Average MOS result of Afaan Oromoo Speech Synthesizer*

Evaluators	**Intelligibility**	**Naturalness**
Females average score result	4.6	3.9
Males average score result	4.2	4.3
Average	4.3	4.1

The analysis result shows good quality in terms of intelligibility and naturalness.

Objective Evaluation Metrics

Subjective evaluation is expensive and time consuming. An objective evaluation would offer an alternative for assessing synthetic speech. This is known as mel cepstral distortion (MCD). The system correlates the effective characteristics of the natural sound with the synthetic sound. Our system works on tenfold cross validation (9/10) rule which is widely used and an effective training method for the system. Accordingly, using mel cepstral distortion (MCD) the result obtained is 6.8, which is encouraging.

Chapter Six: Conclusion and Future Works

6.1 Conclusion

Text-to-Speech (TTS) synthesis can convert an arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person. It can be used as message readers, teaching assistants, tools to aid in communication and learning for the handicapped and visually challenged people. During developing Afaan Oromoo speech synthesis, the system involved collecting text, preprocessing the text, preparing phonetically balanced sentences, recording the sentences, preparing annotated speech database, and design a prototype.

An architecture contain training and synthesis phase. In training phase, first the text and speech corpus are manually prepared for processing. Mel-cepstral coefficients (excitation and spectrum) parameters are obtained from speech database by mel-cepstral analysis. Then using automatic EHMM labeler, the text corpus and speech parameters are align to generate utterance. Based on decision tree clustering method or classification and regression tree, the f0, spectrum and duration are trained to build the context dependent HMM parameters model.

In synthesis phase, first an arbitrarily given text to be preprocessed is converted to a context-based label sequence. Then, these features aligned with trained speech parameters (spectral and excitation) from built context dependent HMMs. lastly, a synthetic speech is obtained using mel log spectrum approximation (MLSA) speech parameters algorithms

In this work, a first attempt is made to develop a speech synthesizer for Afaan Oromoo language using statistical parameter speech synthesis based on HMM method. However, every feature of Afaan Oromoo language was not considered because it needs a lot of time and deep linguistic knowledge. Hence, only the characteristics and way of creation of Afaan Oromoo phonemes are considered. Mean opinion score evaluation technique was used to test the performance of the system. For this study, four hundred sentences are used for training and out of the trained sentences ten arbitrary sentence used for testing. According to training and testing our system, we obtained the result 4.1 and 4.3 out of 5 score in terms of naturalness and intelligibility respectively. A tenfold threshold method is used for training and testing of the prototype.

Furthermore, the performance of the system in generating intelligible speech is also good as per the result of MOS test. Therefore, from these evaluation results it indicates that the synthesized speech is almost resemble to natural human speech. The major contributions of this research are:

- Development Afaan Oromoo letter to sound transliteration tool and incorporation with speech tools.
- Define and integrate of Afaan Oromoo phone sets according language grammar structure and integrate with festvox.
- We proposed statistical parametric synthesis architecture for Afaan Oromoo with new approach.
- We collected and prepared a corpus according to the language grammatical structure.

6.2 Future Works

In this study, Afaan Oromoo statistical parametric based on HMM synthesizer is developed. Even if the developed systems performance is good, there are language dependent features that are not being incorporated in this thesis work. Therefore, to have a speech synthesizer that considers all speech features, the following points are recommended as a future work either to extend the work or to increase the quality of the synthesized speech.

- In the future, it is better to apply the deep learning methods or deep neural network methods (DNN) which will lead to hybrid techniques that incorporate both the benefits of statistical parametric speech based on HMM and unit selection synthesis method.
- Another task that needs future work is affective prosody is not considered in developing the current prototype.
- The texts that are provided to the SPS-FAO do not consider all abbreviated words, numbers, currency and punctuation marks to provide a high quality of speech for language.
- In this thesis, germination handling which is one of the linguistic features of the language was not handled.
- In this study, a voice conversion technique such as speaker interpolation, adaptation and average voice of different prosodic effects that affect the sound of the speech are not considered.

References

[1] R. Tuomo, "Voice source modelling techniques for statistical parametric speech synthesis," Unpublished Doctoral Dissertatations, Helsinki, school of Electrical Engineering,department of Signal processing and Acoustics,Aalto University, March 5,2015.

[2] M. Takashi, "HMM-Based Speech Synthesis and Its Applications," Unpublished Masters thesis, Japan, Tokyo, November, 2002.

[3] C. Manning and J. H. Christopher, "Advanced in natural language processing," in *in proceeding of the 52nd annual meeting of the association for computational linguistics*, Stanford, 2014.

[4] A. Kiflu, "Unit Selection Based Text-to-Speech Synthesizer for Tigrinya Language, School of Information Science, Addis Ababa university," Unpublished Masters thesis, addis ababa, 2004.

[5] D. J. a. J. H. Martin, "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," in *Speech and Language Processing*, S. Edition, Ed., Standford, Speech Communication and language, 2001, pp. 1-15.

[6] A. J. a. P. Mythili, "Developing a Child Friendly Text-to-Speech System," *Hindawi Publishing Corporation*, vol. 2008, p. 6, August 2008.

[7] T. Raitio, "Voice source modelling techniques for statistical parametric speech synthesis," Unpublished Doctoral Dissertations, Aalto University, School of Electrical Engineering,Department of Signal Processing and AcousticsIceland, June 2015.

[8] A. W. Black, "CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling," in *Language Technologies Institute,Carnegie Mellon University*, , Pittsburgh, PA, USA, 2006.

[9] S. Tadesse, "Concatenative Text-To-Speech System for Afaan Oromo Language," Unpublished Masters Thesis,, Addis ababa,Ethiopia, May,2011.

[10] T. Girma, "Human language Technologies and Afaan Oromo," *International Journal of Advanced Research in Engineering and Applied Sciences,*, vol. Vol.3, no. 5, pp. 8-10, May May 2014.

[11] T. Raitio, "Hidden Markov Model Based Finnish Text-to-Speech System Utilizing Glottal Inverse Filtering," Unpublished Masters Thesis, Espoo, May 30,2008.

[12] Heiga Zen, Recent development of the HMM-based speech synthesis system (HTS), Nagoya Institute of Technology, Nagoya Japan: Department of Computer Science,, 2015, pp. Department of Computer Science, Nagoya Institute of Technology, Nagoya 466–8555, Japan.

[13] T. T. T. NGUYEN, "HMM-based Vietnamese Text-To-Speech ,Prosodic Phrasing Modeling, Corpus Design.System Design, and Evaluation," Unpublished Doctrorial Dissertation , Paris, 24 September 2015.

[14] A. K. N. a. S. P. S. P. Panda, "Text to Speech Synthesis with an Indian Language perspective," *International Journal of Grid and Utility Computing, Inderscience, Vol. 6, No. 3/4,* vol. 6, pp. 170-178, 2015.

[15] H. H. a. A. B. David Suendermann, Challenges in Speech Synthesis, New York: Speech Technology, 2010.

[16] A. T. ZEGEYE, "A GENERALIZED APPROACH TO AMHARIC TEXT-TO-SPEECH (TTS) SYNTHESIS SYSTEM," Unpublished Masters Thesis in Information Science , Addis ababa ,Ethiopia, July, 2010.

[17] T. Dutoit, A Short Introduction to Text to Speech synthesis, Boston,: Kluwer Academic Publishers, December,1999.

[18] A. W. B. Gopala Krishna Anumanchipalli, "ADAPTATION TECHNIQUES FOR SPEECH SYNTHESIS IN UNDER-RESOURCED LANGUAGES," in *ICAASP*, Pittsburgh,USA, 2010.

[19] A. K. Tokuda, "Simultaneous Modeling of Phonetic and Prosodic Parameters, and Characteristic Conversion," 2002..

[20] T. Paul, Text to Speech Synthesis, Edinburgh: University’s Centre for Speech Technology Research (CSTR) ., November 19, 2007.

[21] L. H. and M. M. , "Developing Tigrigna Text to Speech Synthesizer," Unpublished Masters Thesis, Addis Ababa ,Ethiopia, Sene 22,2007.

[22] S. R. Mache, "Review on Text-To-Speech Synthesizer," *International Journal of Advanced Research in Computer and Communication Engineering* , vol. 4, no. 8, pp. 54-55, August 2015.

[23] A. Anand , S. Tanuja and W. B. Alan, "Text Processing for Text-to-Speech Systems in Indian Languages," International Institute of Information Technology, Hyderabad, India., edinburgh, 2012.

[24] A. A. Diro, "Automatic Morphological Synthesizer for Afaan Oromoo," Unpublished Masters Thesis, Addis ababa,Ethiopia, 2012.

[25] S. lemmetty, "Review of Speech Synthesis Technology,," Unpublished Masters Thesis, Espoo, March 30, 1999.

[26] D. Munkhtuya and K. Kuldip, "Diphone-Based Concatenative Speech Synthesis for Mongolian," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, March 2008.

[27] B. Ntsako, "Text- To-Speech Synthesis System for Xitsonga using Hidden Markov Models," Unpublished Masters Thesis, UNIVERSITY OF LIMPOPO, June, 2012.

[28] B. Divya, G. Ankita and J. Khushneet, "Punjabi Speech Synthesis System Ussing HTK," *International Journal of Information Sciences and Techniques (IJIST)*, vol. Vol.2, no. No.4,, p. 2, July 2012.

[29] H. A. Dr, " Statistical Parameter Speech based on HMM speech synthesis (HTS)," Unpublished Masters Thesis, Gujarati, India, 2006.

[30] J. Yamagishi, An Introduction to HMM-Based Speech Synthesis, October 2006.

[31] A. W. &. L. K. A. Black, Building synthetic voices., Pittsburgh, PA, USA: Carnegie Mellon University, Language Technologies Institute, 2014.

[32] K. T. a. A. W. B. Heiga Zen, "Statistical parametric speech synthesis," *Speech Communication,.Available online at www.sciencedirect.com,* pp. 1039-1064, 8 april 2009.

[33] H. Zen, "ACOUSTIC MODELING IN STATISTICAL PARAMETRIC SPEECH SYNTHESIS:FROM HMM TO LSTM-RNN," in *Speech Communication* , pittsburg, 2015.

[34] A. Andreyevich, History and Theoretical Basics of Hidden Markov Models, Germany: Speech Communication, July 20, 1922.

[35] A. W Black, H. Zen and T. Keiichi, "STATISTICAL PARAMETRIC SPEECH SYNTHESIS," Nagoya, JAPAN, 2007.

[36] S. K. D. M. Sankar Mukherjee, "A BENGALI HMM BASED SPEECH SYNTHESIS SYSTEM," Unpublished Masters Thesis, Bengali, 2012.

[37] T. T, "Speech Parameters generations Algorithms for HMM based speech synthesis," in *proc. ICASSP -2000*, Japan , june 2000.

[38] T. Yoshimura, "SIMULTANEOUS MODELING OF SPECTRUM, PITCH AND DURATION," in *Proc. of ICSLP*, Japan, 1999.

[39] S. K, "Mel Log Spectrum approximation (MLSA) Filter for speech synthesis," *IECE*, vol. I, no. 2, pp. 122-129, 1983.

[40] R. H. a. A. Vibhute, "Feature Extraction Techniques in Speech Processing: A Survey," *International Journal of Computer Applications,* vol. 107, December 2014.

[41] N. K. a. T. T. Tokuda K, "Speech Synthesis Based on Hidden Markov Models," in *Edinburg Research Explorer*, Japan, januart 2013.

[42] S. K, "HMMs as Generative Models of Speech," *Workshop on Text-to-Speech (TTS) Synthesis,* 16-18 June 2014.

[43] A. W. B. a. J. Kominek, "OPTIMIZING SEGMENT LABEL BOUNDARIES FOR STATISTICAL SPEECH SYNTHESIS," *IEEE*, vol. 9, no. 5, pp. 3785-3788, 2009.

[44] W. B. ALan and P. Taylor, The festival Speech synthesis System, Edinburgh, 2000.

[45] W. Lu and H. Mark , "Speech Synthesis using Mel-Cepstral Coeffiecent Feature," Unpublished Senior thesisin Electrical Engineering University of Illinois at Urbana-Champaign , Urbana, May 2018.

[46] W. T. a. D. TAMIRAT, "Investigating Afan Oromo Language Structure," *American Journal of Computer Science and Engineering Survey*, vol. 1, no. 1, pp. 1-8, 2017.

[47] F. B. Kebede, "Dissimilation in Oromo Phonology," *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT*, vol. 3, no. 13, pp. 187-195, December, 2014 .

[48] G. G. Eggi, "Afaan Oromo Text Retrieval System," Unpublished Masters Thesis, Addis ababa, Ethiopia, 2012.

[49] H. Xuedong, A. Alex and H. Hsiao-Wuen, Spoken Language processing: A guide theory, algorithm and system development, carnigie Mellon University: in United state of america, 2001.

[50] O. Gadisa and M. DR Dida, "Design and Implementation of Afaan oromo Spell Checker," Unpublished Masters thesis, Addis ababa, june 2013.

[51] I. Nagi, Caassefama Afan oromo, Addis Ababa, Ethiopia:, 2006.

[52] M. L. Kejela, "Named Entity Recognition for Afan Oromo," Unpublished masters Thesis, Addis Ababa,Ethiopia, 2010.

[53] K. Tokuda, H. Zen and A. W. Black, "An HMM Based Speech Synthesis System Applied To English," Unpublished Masters Thesis, Department of Computer Science, Carnegie Mellon University, 2002.

[54] B. Kasaye, "DEVELOPING A SPEECH SYNTHESIZER FOR AMHARIC LANGUAGE," Unpublished Masters Thesis, Addis ababa,Ethiopia, October, 2008.

[55] T. A. Gebreselassie, "TEXT-TO-SPEECH SYNTHESIZER FOR WOLAYTTA LANGUAGE," Unpublished Masters Thesis, Addis Ababa,Ethiopia, October,2009.

Appendices

Addis Ababa University

Department of Computer Science

Users' Evaluation of HMM-based speech synthesis for Afaan Oromoo Language.

The aim of this questionnaire is to evaluate the performance of the hidden Markov model based speech synthesis for Afaan Oromoo (SPS-FAO). So, we kindly request you to consider each question critically and give the rank honestly. I would like to thank you for your help and participation.

Appendix A: Utterance Structure

(EST_File utterance
DataType ascii
version 2
EST_Header_End
Features max_id 135 ; type Text ; iform "\"maddeen baajata biyya keessaa fi alaa irraa akka ta'e himameera.\"" ; filename prompt-utt/uniph_00002.utt ; fileid uniph_00002 ;
Stream_Items
1 id _1 ; name maddeen ; whitespace "" ; prepunctuation "" ;
2 id _2 ; name baajata ; whitespace " " ; prepunctuation "" ;
3 id _3 ; name biyya ; whitespace " " ; prepunctuation "" ;
4 id _4 ; name keessaa ; whitespace " " ; prepunctuation "" ;
5 id _5 ; name fi ; whitespace " " ; prepunctuation "" ;
6 id _6 ; name alaa ; whitespace " " ; prepunctuation "" ;
7 id _7 ; name irraa ; whitespace " " ; prepunctuation "" ;
8 id _8 ; name akka ; whitespace " " ; prepunctuation "" ;
9 id _9 ; name ta'e ; whitespace " " ; prepunctuation "" ;
10 id _10 ; name himameera ; punc . ; whitespace " " ; prepunctuation "" ;
11 id _20 ; name himameera ; pbreak B ; pos nil ;
12 id _21 ; name . ; pbreak B ; pos punc ;
13 id _19 ; name ta'e ; pbreak NB ; pos nil ;
14 id _18 ; name akka ; pbreak NB ; pos nil ;

15 id _17 ; name irraa ; pbreak NB ; pos nil ;
16 id _16 ; name alaa ; pbreak NB ; pos nil ;
17 id _15 ; name fi ; pbreak NB ; pos nil ;
18 id _14 ; name keessaa ; pbreak NB ; pos nil ;
19 id _13 ; name biyya ; pbreak NB ; pos nil ;
20 id _12 ; name baajata ; pbreak NB ; pos nil ;
21 id _11 ; name maddeen ; pbreak NB ; pos nil ;
22 id _23 ; name syl ; stress 1 ;
23 id _25 ; name syl ; stress 0 ;
24 id _28 ; name syl ; stress 0 ;
25 id _30 ; name syl ; stress 1 ;
26 id _33 ; name syl ; stress 0 ;
27 id _36 ; name syl ; stress 1 ;
28 id _38 ; name syl ; stress 0 ;
29 id _41 ; name syl ; stress 1 ;
30 id _44 ; name syl ; stress 0 ;
31 id _46 ; name syl ; stress 0 ;
32 id _49 ; name syl ; stress 0 ;
33 id _51 ; name syl ; stress 0 ;
34 id _54 ; name syl ; stress 0 ;
35 id _57 ; name syl ; stress 1 ;
36 id _59 ; name syl ; stress 0 ;
37 id _61 ; name syl ; stress 0 ;
38 id _64 ; name syl ; stress 1 ;
39 id _67 ; name syl ; stress 0 ;
40 id _70 ; name pau ; dur_factor 1 ; end 0.035 ;
41 id _24 ; name m ; dur_factor 1 ; end 0.295 ;
42 id _26 ; name d ; dur_factor 1 ; end 0.45 ;
43 id _27 ; name ee ; dur_factor 1 ; end 0.605 ;
44 id _29 ; name n ; dur_factor 1 ; end 0.795 ;
45 id _31 ; name b ; dur_factor 1 ; end 1.41 ;
46 id _32 ; name aa ; dur_factor 1 ; end 1.49 ;
47 id _34 ; name j ; dur_factor 1 ; end 1.505 ;
48 id _35 ; name a ; dur_factor 1 ; end 1.66 ;
49 id _37 ; name b ; dur_factor 1 ; end 1.905 ;
50 id _39 ; name y ; dur_factor 1 ; end 2.27 ;
51 id _40 ; name a ; dur_factor 1 ; end 2.365 ;
52 id _42 ; name k ; dur_factor 1 ; end 2.695 ;
53 id _43 ; name ee ; dur_factor 1 ; end 3.075 ;
54 id _45 ; name aa ; dur_factor 1 ; end 3.485 ;

55 id _47 ; name f ; dur_factor 1 ; end 4.01 ;
56 id _48 ; name i ; dur_factor 1 ; end 4.13 ;
57 id _50 ; name aa ; dur_factor 1 ; end 4.64 ;
58 id _52 ; name r ; dur_factor 1 ; end 5.11 ;
59 id _53 ; name aa ; dur_factor 1 ; end 5.335 ;
60 id _55 ; name k ; dur_factor 1 ; end 5.78 ;
61 id _56 ; name a ; dur_factor 1 ; end 5.87 ;
62 id _58 ; name t ; dur_factor 1 ; end 6.01 ;
63 id _60 ; name ; dur_factor 1 ; end 6.105 ;
64 id _62 ; name ·; dur_factor 1 ; end 6.3 ;
65 id _63 ; name e ; dur_factor 1 ; end 6.755 ;
66 id _65 ; name h ; dur_factor 1 ; end 6.94 ;
67 id _66 ; name ee ; dur_factor 1 ; end 6.955 ;
68 id _68 ; name r ; dur_factor 1.5 ; end 7.02 ;
69 id _69 ; name a ; dur_factor 1.5 ; end 7.04 ;
70 id _71 ; name pau ; dur_factor 1 ; end 7.34 ;
71 id _72 ; name Accented ;
72 id _73 ; name Accented ;
73 id _74 ; name Accented ;
74 id _75 ; name Accented ;
75 id _76 ; name Accented ;
76 id _77 ; name Accented ;
77 id _102 ; f0 110 ; pos 3.41 ;
78 id _100 ; f0 121.75 ; pos 3.07 ;
79 id _101 ; f0 112.5 ; pos 3.08 ;
80 id _99 ; f0 123.75 ; pos 2.87 ;
81 id _98 ; f0 113.75 ; pos 2.86 ;
82 id _95 ; f0 126.25 ; pos 2.43 ;
83 id _96 ; f0 124.25 ; pos 2.52 ;
84 id _97 ; f0 115.625 ; pos 2.53 ;
85 id _94 ; f0 116.25 ; pos 2.42 ;
86 id _92 ; f0 130.5 ; pos 1.53 ;
87 id _93 ; f0 121.25 ; pos 1.54 ;
88 id _91 ; f0 132.5 ; pos 1.33 ;
89 id _90 ; f0 122.5 ; pos 1.32 ;
90 id _87 ; f0 134.375 ; pos 1 ;
91 id _88 ; f0 132.375 ; pos 1.09 ;
92 id _89 ; f0 123.75 ; pos 1.1 ;
93 id _86 ; f0 124.375 ; pos 0.99 ;
94 id _84 ; f0 134.875 ; pos 0.76 ;

95 id _85 ; f0 125.625 ; pos 0.77 ;
96 id _83 ; f0 136.875 ; pos 0.56 ;
97 id _82 ; f0 126.875 ; pos 0.55 ;
98 id _79 ; f0 139.375 ; pos 0.12 ;
99 id _80 ; f0 137.375 ; pos 0.21 ;
100 id _81 ; f0 128.75 ; pos 0.22 ;
101 id _78 ; f0 129.375 ; pos 0.11 ;
102 id _135 ; name BB ;
End_of_Stream_Items
Relations
Relation Token ; "(" ")" ;
11 21 1 0 0 0
1 1 0 11 2 0
12 20 2 0 0 0
2 2 0 12 3 1
13 19 3 0 0 0
3 3 0 13 4 2
14 18 4 0 0 0
4 4 0 14 5 3
15 17 5 0 0 0
5 5 0 15 6 4
16 16 6 0 0 0
6 6 0 16 7 5
17 15 7 0 0 0
7 7 0 17 8 6
18 14 8 0 0 0
8 8 0 18 9 7
19 13 9 0 0 0
9 9 0 19 10 8
20 11 10 0 21 0
21 12 0 0 0 20
10 10 0 20 0 9
End_of_Relation
Relation Word ; "(" ")" ;
1 21 0 0 2 0
2 20 0 0 3 1
3 19 0 0 4 2
4 18 0 0 5 3
5 17 0 0 6 4
6 16 0 0 7 5

7 15 0 0 8 6
8 14 0 0 9 7
9 13 0 0 10 8
10 11 0 0 0 9
End_of_Relation
Relation Syllable ; "(" ")" ;
1 22 0 0 2 0
2 23 0 0 3 1
3 24 0 0 4 2
4 25 0 0 5 3
5 26 0 0 6 4
6 27 0 0 7 5
7 28 0 0 8 6
8 29 0 0 9 7
9 30 0 0 10 8
10 31 0 0 11 9
11 32 0 0 12 10
12 33 0 0 13 11
13 34 0 0 14 12
14 35 0 0 15 13
15 36 0 0 16 14
16 37 0 0 17 15
17 38 0 0 18 16
18 39 0 0 0 17
End_of_Relation
Relation Segment ; "(" ")" ;
1 40 0 0 2 0
2 41 0 0 3 1
3 42 0 0 4 2
4 43 0 0 5 3
5 44 0 0 6 4
6 45 0 0 7 5
7 46 0 0 8 6
8 47 0 0 9 7
9 48 0 0 10 8
10 49 0 0 11 9
11 50 0 0 12 10
12 51 0 0 13 11
13 52 0 0 14 12
14 53 0 0 15 13

15 54 0 0 16 14
16 55 0 0 17 15
17 56 0 0 18 16
18 57 0 0 19 17
19 58 0 0 20 18
20 59 0 0 21 19
21 60 0 0 22 20
22 61 0 0 23 21
23 62 0 0 24 22
24 63 0 0 25 23
25 64 0 0 26 24
26 65 0 0 27 25
27 66 0 0 28 26
28 67 0 0 29 27
29 68 0 0 30 28
30 69 0 0 31 29
31 70 0 0 0 30
End_of_Relation
Relation SylStructure ; "(" ")" ;
15 41 12 0 0 0
12 22 1 15 13 0
16 42 13 0 17 0
17 43 0 0 0 16
13 23 0 16 14 12
18 44 14 0 0 0
14 24 0 18 0 13
1 21 0 12 2 0
21 45 19 0 22 0
22 46 0 0 0 21
19 25 2 21 20 0
23 47 20 0 24 0
24 48 0 0 0 23
20 26 0 23 0 19
2 20 0 19 3 1
27 49 25 0 0 0
25 27 3 27 26 0
28 50 26 0 29 0
29 51 0 0 0 28
26 28 0 28 0 25
3 19 0 25 4 2

32 52 30 0 33 0
33 53 0 0 0 32
30 29 4 32 31 0
34 54 31 0 0 0
31 30 0 34 0 30
4 18 0 30 5 3
36 55 35 0 37 0
37 56 0 0 0 36
35 31 5 36 0 0
5 17 0 35 6 4
39 57 38 0 0 0
38 32 6 39 0 0
6 16 0 38 7 5
41 58 40 0 42 0
42 59 0 0 0 41
40 33 7 41 0 0
7 15 0 40 8 6
44 60 43 0 45 0
45 61 0 0 0 44
43 34 8 44 0 0
8 14 0 43 9 7
49 62 46 0 0 0
46 35 9 49 47 0
50 63 47 0 0 0
47 36 0 50 48 46
51 64 48 0 52 0
52 65 0 0 0 51
48 37 0 51 0 47
9 13 0 46 10 8
55 66 53 0 56 0
56 67 0 0 0 55
53 38 10 55 54 0
57 68 54 0 58 0
58 69 0 0 0 57
54 39 0 57 0 53
10 11 0 53 11 9
11 12 0 0 0 10
End_of_Relation
Relation IntEvent ; "(" ")" ;
1 71 0 0 2 0

2 72 0 0 3 1
3 73 0 0 4 2
4 74 0 0 5 3
5 75 0 0 6 4
6 76 0 0 0 5
End_of_Relation
Relation Intonation ; "(" ")" ;
7 71 1 0 0 0
1 22 0 7 2 0
8 72 2 0 0 0
2 25 0 8 3 1
9 73 3 0 0 0
3 27 0 9 4 2
10 74 4 0 0 0
4 29 0 10 5 3
11 75 5 0 0 0
5 35 0 11 6 4
12 76 6 0 0 0
6 38 0 12 0 5
End_of_Relation
Relation Target ; "(" ")" ;
17 101 1 0 0 0
1 40 0 17 2 0
18 98 2 0 19 0
19 99 0 0 20 18
20 100 0 0 0 19
2 41 0 18 3 1
21 97 3 0 0 0
3 44 0 21 4 2
22 96 4 0 0 0
4 45 0 22 5 3
23 94 5 0 24 0
24 95 0 0 0 23
5 46 0 23 6 4
25 93 6 0 0 0
6 48 0 25 7 5
35 83 0 0 36 34
36 84 0 0 0 35
12 62 0 34 13 11
37 81 13 0 0 0

13 65 0 37 14 12
38 80 14 0 0 0
14 66 0 38 15 13
39 78 15 0 40 0
40 79 0 0 0 39
15 67 0 39 16 14
41 77 16 0 0 0
16 69 0 41 0 15
End_of_Relation
Relation Phrase ; ()
2 21 1 0 3 0
3 20 0 0 4 2
4 19 0 0 5 3
5 18 0 0 6 4
6 17 0 0 7 5
7 16 0 0 8 6
8 15 0 0 9 7
9 14 0 0 10 8
10 13 0 0 11 9
11 11 0 0 0 10
1 102 0 2 0 0
End_of_Relation
End_of_Relations
End_of_Utterance

Appendix B: Afaan Oromoo Phonesets

(defPhoneSet aau _ao

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 affricative nasal liquid

(ctype s f a n l r 0)

place of articulation: labial alveolar palatal labio-dental

dental velar

(cplace l a p b d v g 0)

consonant voicing

(cvox + - 0)

)

(

(pau - 0 - - - 0 0 -) ;; slience ...

(SIL - 0 0 0 0 0 0 -);;silence….

(a + s 3 3 - 0 0 0)

(e + s 2 1 - 0 0 0)

(i + s 1 1 - 0 0 0)

(o + s 2 3 + 0 0 0)

(u + s 1 3 + 0 0 0)

(aa + l 3 3 - 0 0 0)

(ee + l 2 1 - 0 0 0)

(ii + l 1 1 - 0 0 0)

(oo + l 2 3 + 0 0 0)

(uu + l 1 3 + 0 0 0)

(b - 0 0 0 0 s l +)

(c - 0 0 0 0 a p +)

(d - 0 0 0 0 s a +)

(f - 0 0 0 0 f b -)

(g - 0 0 0 0 s v +)

(h - 0 0 0 0 f g -)

(j - 0 0 0 0 a p +)

(k - 0 0 0 0 s v -)

(l - 0 0 0 0 l a +)

(m - 0 0 0 0 n l +)

(n - 0 0 0 0 n a +)

(p - 0 0 0 0 s l -)

(q - 0 0 0 0 s v +)

(r - 0 0 0 0 r a +)

(s - 0 0 0 0 f a +)

(t - 0 0 0 0 s a -)

(v - 0 0 0 0 f b +)

(w - 0 0 0 0 r l +)

(x - 0 0 0 0 s a +)

(y - 0 0 0 0 r p +)

(z - 0 0 0 0 f a +)

(ch - 0 0 0 0 a p -)

(dh - 0 0 0 0 s a +)

(ny - 0 0 0 0 n p +)

(ph - 0 0 0 + s l +)

(sh - 0 0 0 0 f p -)

(ts - 0 0 0 0 f a -)

(zy - 0 0 0 0 f p -)

(' + 0 0 0 0 s 0 +)))

Appendix C: Sample of Afaan Oromoo letter to sound rule transliteration

```
#include<iostream>
#include<fstream>
using namespace std;
int isVowel(char ch);
void word_to_pronunciation(string word, const char *output_filename);
ofstream outf;
ifstream inpf;
string normalize(string input);
string de_normalize(char input);
int main()
{
	string word;
	/*if(argc != 3)
	{
		cout<<"Error: Usuage "<<argv[0]<<" word_input_filename
word_feature_output_filename"<<endl;
		return 1;
	}*/
	inpf.open("/home/muhidin/tts/adb_ao_z/bin/word");
	if(inpf.fail())
	{
		cout<<"unable to open file"<<endl;
		return 1;
	}
	inpf>>word;
	word_to_pronunciation(word, "/home/muhidin/tts/adb_ao_z/wordpronunciation");
	inpf.close();
	return 0;
}
void word_to_pronunciation(string word, const char *output_filename)
{
	outf.open(output_filename);
	if(outf.fail())
```

```
	{
		cout<<"unable to open file"<<output_filename<<endl;
		return;
	}
	string normalized_word = normalize(word);
	cout<<normalized_word<<endl;
	int len = normalized_word.length();

	outf<<"(set! wordstruct '( ( (";

	int index = 0;
	while(index < len-2)
	{
		if(isVowel(normalized_word[index+1]))
		{
			if(index == 0) outf<<de_normalize(normalized_word[index])<<"
"<<de_normalize(normalized_word[index+1])<<" ) 1 ) ( ( ";
			else outf<<de_normalize(normalized_word[index])<<"
"<<de_normalize(normalized_word[index+1])<<" ) 0 ) ( ( ";
			index = index + 2;
		}
		else
		{
			if(index == 0) outf<<de_normalize(normalized_word[index])<<" ) 1 )
( ( ";
			else outf<<de_normalize(normalized_word[index])<<" ) 0 ) ( ( ";
			index++;
		}
	}
	if(index + 1 == len)
		outf<<de_normalize(normalized_word[index])<<" ) 0) ))"<<endl;
	else
		outf<<de_normalize(normalized_word[index])<<"
"<<de_normalize(normalized_word[index+1])<<" ) 0) ))"<<endl;
	outf.close();
}

string de_normalize(char input)
{
	string output;
	string temp = "x";

	switch(input){
		case 'A': {output = "aa";break;}
		case 'E': {output = "ee";break;}
		case 'I': {output = "ii";break;}
```

```
                case 'O': {output = "oo";break;}
                case 'U': {output = "uu";break;}
                case 'C': {output = "ch";break;}
                case 'D': {output = "dh";break;}
                case 'P': {output = "ph";break;}
                case 'S': {output = "sh";break;}
                case 'N': {output = "ny";break;}
                case 'Z': {output = "zy";break;}
                case 'T': {output = "ts";break;}
                case 'J': {output = "";break;}
                default: {temp[0] = input; output = temp;}
        }
        return output;

}

string normalize(string input)
{
      string output;
      int i = 0;
      int len = input.length();
      while (i < len-1)
      {
         if(input[i] == 'a')
          {
           switch(input[i+1]){
                    case 'a': {output =output + "A";break; }
             //default: {cout<<"a: error word "<<input<<" processing at index "<<i<<endl;}
                              }
                i = i + 2;
          }


           else if(input[i] == 'e')
                {
                 switch(input[i+1]){
                 case 'e': {output =output + "E";break; }
          //   default: {cout<<"e: error word "<<input<<" processing at index "<<i<<endl;}
                               }
                i = i + 2;
           }


          else if(input[i] == 'i')
          {
           switch(input[i+1]){
```

```
            case 'i': {output =output + "I";break; }
//      default: {cout<<"i: error word "<<input<<" processing at index "<<i<<endl;}
                    }
          i = i + 2;
     }


     else if(input[i] == 'o')
     {
      switch(input[i+1]){
            case 'o': {output =output + "O";break; }
//        default: {cout<<"o: error word "<<input<<" processing at index "<<i<<endl;}
                    }
          i = i + 2;
     }

     else if(input[i] == 'u')
     {
      switch(input[i+1]){
            case 'u': {output =output + "U";break; }
//      default: {cout<<"u: error word "<<input<<" processing at index "<<i<<endl;}
                    }
          i = i + 2;
     }

     else if(input[i+1] == 'h' && len > i+ 2 && input[i+2] != 'h')
      {
       switch(input[i]){
         case 'c': {output =output + "C";break; }
         case 'd': {output =output + "D";break; }
         case 'p': {output =output + "P";break; }
         case 's': {output =output + "S";break; }
//        default: {cout<<"h: error word "<<input<<" processing at index "<<i<<endl;}
                    }
          i = i + 2;
      }


     else if(input[i+1] == 'y' && len > i+ 2 && input[i+2] != 'y')
      {
       switch(input[i]){
         case 'n': {output =output + "N";break; }
         case 'z': {output =output + "Z";break; }
//       default: {cout<<"y: error word "<<input<<" processing at index "<<i<<endl;}
                    }
          i = i + 2;
```

```
            }

            else if(input[i+1] == 's' && len > i+ 2 && input[i+2] != 's')
            {
             switch(input[i]){
                case 't': {output =output + "T";break; }
        //       default: {cout<<"s: error word "<<input<<" processing at index "<<i<<endl;}
                    }
                    i = i + 2;
            }
          //  else if(input[i+1] != '')
         //    {
           //    switch(input[i]){
             //    case '': {output =output + "J";break; }
         //      default: {cout<<"y: error word "<<input<<" processing at index "<<i<<endl;}
                  //              }
                 //       i = i + 2;
                //}
          else
            {
                    output = output + input[i];
                    i++;
            }
        }

        if (i == len -1) output = output + input[i];
        return output;
}

int isVowel(char ch)
{
          if(ch == 'a' || ch == 'e' || ch == 'i' ||ch == 'o' ||ch == 'u' ||ch == 'A' ||ch == 'E' ||ch == 'I' ||
ch == 'O' || ch == 'U') return 1;
          return 0; }
```

***Appendix D*:** Questionnaire

Part 1:

1. Are you familiar with Oromiffaa?

    A. Yes

    B. No

Part 2:

Testing the Afaan Oromoo TTS Synthesizer

Listen to the following audio files and tick (√) what you consider is the right value for the synthesizer. Please give a grade from a scale 1 to 5 in each synthesized Afaan Oromoo sentences. 1=bad, 2=fair, 3=good, 4=very good and 5= excellent. **Naturalness:** measures to what extent that synthesized speech looks like human sounds.

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

Sentence 1. _____

Sentence 2. _____

Sentence 3. _____

Sentence 4. _____

Sentence 5. _____

Sentence 6. _____

Sentence 7. _____

Sentence 8. _____

Sentence 9. _____

Sentence 10. _____

Intelligibility: measures the understandability of the synthesized speech.

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

Sentence 1. _____

Sentence 2. _____

Sentence 3. _____

Sentence 4. _____

Sentence 5. _____

Sentence 6. _____

Sentence 7. _____

Sentence 8. _____

Sentence 9. _____

Sentence 10. _____

Appendix E: Sentences Used to Test the Afaan Oromoo TTS

(uniph_0001 "iji waaqayyoo iddoo hundumaa jira, isa hamaa fi isa gaarii ni arga.")

(uniph_0002 "dubartoonni ogeeyyiin hundinuu mana isaanii ni ijaaru.")

(uniph_0003 " dhabatni Gamtoman gargarsa qarshiikennera)

(uniph_0004 "namni yaada qajeelaadhaan jiraatu waaqayyoon ni sodaata.")

(uniph_0005 "namni jireenya isaatti micciiramaa ta’e garuu waaqayyoon hin sodaatu.")

(uniph_0006 "namni gowwaan dubbii afaan isaa keessaa ba’uun of gurguddisa.")

(uniph_0007 "ilmi nama kenna waaqayoti ”)

(uniph_0008 "Ayyani irrecha magaala finfinneti kabajamera.")

(uniph_0009 "namni of eeggachaa hin dubbanne garuu, waaqayyoon ni gaddisiisa.")

(uniph_0010 "namni gaariin waaqayyo biraa ayyaana ni argata.")

Declaration

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

Declared by:

Muhidin Kedir Wosho Date: June, 2020.

_____

Signature:

Confirmed by advisor:

Dida Midekso (PhD) Date June, 2020.

_____

Signature:

Place and date of submission: Addis Ababa, June, 2020.