ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

A SPEAKER INDEPENDENT TEXT-TO-SPEECH SYNTHESIS (TTS) FOR AMHARIC LANGUAGE USING HIDDEN MARKOV MODEL

BY:

HABTAMU ABATE DEMESSIE

A THESIS SUBMITED TO
THE SCHOOL OF INFORMATION SCIENCE OF ADDIS ABABA UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION SCIENCE

January 2018

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

A SPEAKER INDEPENDENT TEXT-TO-SPEECH SYNTHESIS (TTS) FOR AMHARIC LANGUAGE USING HIDDEN MARKOV MODEL

BY:

HABTAMU ABATE DEMESSIE

APPROVED BY

Examining board:	Signature
1. Wondowssen Mulugeta (PhD), Advisor	_____
2. Solomon Tefera (PhD)	_____
3. Tibebe Beshah (PhD)	_____

Acknowledgements

First and for most I would like to thank the almighty GOD. Next I would like to acknowledge my advisor Dr. Wondowssen Mulugeta for his constructive comment and advice through this study. I gratefully would like to thanks Eteyi and Abiwu for those invaluable help through my life.

Abréviations

CD-HMM	Context-Dependent HMM
CI-HMM	Context-Independent HMM
MLLR	Maximum Likelihood Linear Regression
SD	Speaker Dependent
SI	Speaker-Independent
NLP	Natural Language Processing
G2P	Grapheme To Phoneme
DSP	Digital Signal Processing
F0	Fundamental Frequency
FD-PSOLA	Frequency-Domain Pitch Synchronous Overlap Add
GV	Global Variance
HMM	Hidden Markov Model
HTK	Hidden Markov Model Tool Kit
HTS	Hidden Markov Based speech synthesis tool kit
IPA	International Phonetic Alphabet
LPC	Linear Predictive Coding
MCD	Mel Capstral Distortion
MLSA	Mel Log Spectral Approximation
MOS	Mean Opinion Score
MSD	HMM_Multi-Space Probability Distribution HMM
NLP	Natural Language Processing
PDF	Probability Density Function
SAT	Speaker Adaptive Training
TD-PSOLA	Time-Domain Pitch Synchronous Overlap Add
TTS	Text To Speech
ATTS	Amharic Text To Speech

Contents

Abstract

In this paper we present Hidden Markov Model (HMM)-based speaker independent Amharic Text-to-Speech system. Amharic, the common spoken language in Ethiopia and Federal Working Language, speaker-independent modeling methods were employed using HMM-based Text-to-Speech technique on a read speech database of 726 sentences uttered by 3 female and 3 male speaker with various speaking styles. Speech signals were sampled at a rate of 16 kHz and windowed by using a 25-ms Blackman window with a 5-ms shift, then 25 Mel-cepstral coefficients including the zeroth coefficient, the logarithm of the fundamental frequency, and its dynamic values (delta and delta-delta) coefficients obtained using Mel-cepstral analysis technique to model context-dependent phoneme HMMs. The speech information is first modeled by context Dependent HMMs, including: (1) spectral envelop and gain; (2) voiced/unvoiced and fundamental frequency; and (3) duration. The corresponding 3 state left to right HMMs was automatically trained by construct speaker independent model as initial model ,then speaker Adaptive model is estimated by using speaker independent model using one male speech data. A decision-based clustering technique was applied in isolation to the distributions of Mel-cepstral, log F0, and state durations of context-dependent phoneme HMMs. Finally, to improve the voice quality, trajectory HMM and mixed excitation model was included by applying parameter generation algorithm based on ML using dynamic features to the Gaussian Mixture Model (GMM). Objective evaluation was conducted to evaluate the speaker Independent or Speaker Adaptation training demo (SAT) using spectral analysis and preference score, it treats the training data which consists of several speakers' speech as that of one speaker and makes no distinctions among the training speakers of the average voice model. in addition the voice conversion technique evaluate using subjective evaluation. In a test of subjective evaluation more than 60% of the speech generated from the voice conversion models using the first 30 sentences is judged to almost the same score of SI models.

Finally subjective mean opinion score (MOS) evaluation was conducted to evaluate the overall performance of the adapted models and developed system. The developed Amharic Speech Synthesizer attains 74% intelligibility and 70 % naturalness MOS result from fifty subjects' first language Amharic speakers. Besides the intelligibility test, we have performed a unit test on the text normalizer. The performance of text normalizer is 85% for Amharic numbers, punctuation marks and abbreviation.

Chapter 1

Introduction

1.1. Background of the study

Speech is ultimately a communication tool; it gives us the possibility to interact with the world through the transmission of a message. Speech technology can improve communication by means of enhancing speech, making it more compact for transmission or storage and more robust to adverse conditions *(Valentin, 2013).* In the field of speech technology, A text-to-speech (TTS) system is one of the human-machine interfaces using speech. In recent years, TTS system is developed as an output device of human-machine interfaces, and it is used in many application such as a car navigation system, information retrieval over the telephone, voice mail, a speech-to-speech translation system , services over telephone, e-document reading, and speaking system for handicapped people, dialogue notification and reading applications as well as personalized voices for people that have lost the use of their own and so on (Klatt,1987;Zen et al,2007). To maintain communication success, humans change the way they speak and hear according to many factors, like the age, gender, native language and social relationship between talker and listener. Other factors are dictated by how communication takes place, such as environmental factors like an active competing speaker or limitations on the communication channel *(Valentin, 2013).* To communicate with in natural (human--human) interaction and human-machine interfaces we use natural or synthetic voices that can adapt to different listening scenarios and keep the level of naturalness and intelligibility to be high.

The ultimate goal of text-to-speech synthesis (TTS) is to produce synthetic voices that should sound as natural, expressive and intelligible as possible and if necessary be similar to a particular speaker. It has been one of a core research area for the last many decades for different languages aiming at accurate modeling of different voice characteristics as well as prosodic features of speech (*Zen et al, 2009).*

The design of TTS systems faces three main challenges *(Pouget ,2000)*:

1. Incremental Natural Language Processing (NLP): extracting linguistic information needed for the generation of the audio waveform (e.g. grammatical/syntactic structure, Phoenician) from an 'incomplete' sentence.

2. .Incremental waveform generation: the generation of the speech waveform from a set of potentially incomplete or inaccurate linguistic features.

3. Time management: when do we deliver the synthetic voice chunks to maintain fluency and naturalness?

One of the biggest challenges in this field is the production of naturally sounding synthetic voices. This means that is not enough that the synthetic voices have high quality; they must also be able to capture the natural expressiveness that the human speech has *(Pouget,2000).* Nowadays, speech research area introduces corpus based statistical parametric speech synthesis known as hidden Markov models (HMM) based speech synthesis system. HTS was emerged this last decade as a promising technique for the automatic generation of speech from text (*Zen et al, 2009).* HMM based speech synthesis system can basically be either speaker-dependent or speaker-independent. A speaker-dependent system is intended to be used by a single speaker and is therefore trained to understand one particular speech pattern. A speaker-independent system is intended for use by any speaker and is naturally more difficult to achieve. The HTS techniques are promoted from better flexibility on speaker-independent and speaker adaptability which give special emphasis on voice characteristics such as speaker individualities, speaking styles, and emotions *(Yamagishi et al, 2004).*

In this work, we address this problem in the framework of HMM-based speech synthesis, on exploring the possibility to develop speaker independent TTS using HTS system that enhance the intelligibility and naturalness of synthetic speech with a special focus on including NLP module, to extracting linguistic information needed for the generation of the audio waveform, estimation of the prosodic features of speech and speaker adaptation technique. The statistical parametric nature of HMM-based speech synthesis offers a high degree of control over the generated speech by modifying the models or extracted parameters (*Zen et al, 2009).* So we are able to control the acoustic characteristics of the generated speech without the need for new data.

1.2 Statement of the problem

Speech is one of the vital forms of communication for human being and it is the core activity in our day to day life *(Valentin, 2013).* Currently, many speech synthesis systems have been made possible and successful results were obtained in various application areas for 'major' languages such as English, Japanese etc. There is no single official definition on what makes a language low resource, and the meaning usually shifts depending on the target application. For applications such as text to speech systems, this requires acoustic-phonetic characterization sound patterns, perceptual psychology, mathematical modeling of speech production, structured programming, and computer hardware design (Klatt, 1987). However, thousands of the world's languages with over millions of speakers can be considered low resource too in a time critical application such as text to speech system. From an TTS perspective, a low resource language can be considered as a language that is not advance or lacking in linguistic theory, acoustic-phonetic characterization sound patterns, perceptual psychology, mathematical modeling of speech production, structured programming, and computer hardware design (Allen et al, 1987). In addition to such technologies, with the time constraint, it is often hard to collect the sufficient amount of resources and lack of researches in the area. Recently many localization projects are being undergoing for many languages it is quite inadequate and the localization process is not an easy task mainly because of the lack of linguistic resources and absence of similar works in the area. Therefore, there is a strong demand for the development of a speech synthesizer for under resource language (URL) including Amharic.

Speech can be generated from text in a variety of ways. The first TTS methods proposed were constructed by rules on how speech sounds are produced. Instead of following production rules, the next generation of TTS systems creates speech from the concatenation of natural speech components. Concatinative systems were first proposed in the shape of fixed component units, diphone synthesizers *(Moulines et al., 1990; Charpentier et al., 1986).* A diphone is a segment defined from the middle of one phone to the middle of the subsequent phoneme. As more storage and computing power became available, the second generation of Concatinative systems appeared: unit selection systems *(Black, 2000).* Currently the most popular speech synthesis technique is unit selection, where appro-

priate sub-word units are selected from large speech databases. Over the last decade, this technique has been shown to synthesize high quality speech and is used for many applications. Although it is very hard to surpass the quality of the best examples of unit selection, it does have a limitation that the synthesized speech will strongly resemble the style of the speech recorded in the database. As we require speech which is more varied in voice characteristics, speaking styles, and emotions, we need to record larger and larger databases with these variations to achieve the synthesis we desire without degrading the quality (Yamagishi et al., 2004). However, recording such a large database is very difficult and costly .One possible way of decreasing this gap is to use the statistical parametric TTS systems (in our case HMM) by applying average voice model and different speaker interpolation technique, on the available multiple speakers speech data (Yamagishi et al, 2007). The statistical parametric TTS systems examined the trade-offs between amounts of data and voice quality and using these methods successful results were obtained for different languages (Yoshimura et al., 1999). Some researchers have been done on speech synthesis for Amharic language using the rule based, concatenation (diphone and unit selection) and statistical parameter based speech synthesis techniques (Laine,1998; Sebsibe et al., 2005; Nadew, 2008; Alula, 2010; Eyob, 2011;Mulat, 2012 and Bereket,2008). Inclusion of non-standard words, pronunciation dictionary and prosody are significantly positively affecting the quality of the synthesized speech.

Nadew (2010) used the formant synthesis technique only for Amharic vowels, and consonants are not considered in his work. (Bereket, 2008, Bahiru, 2017) do not consider abbreviated words, numbers, and punctuation marks. Since every written text has these tokens, it would be better if they are handled by NLP module. in particular (Alula, 2010) uses the rule-based mapping process to convert non-standard words to their equivalent standard words. He used non-standard words (NSWs) and standard words (common words and proper names) to build the system. The inconsistency in usage of non-standard words and an existence of ambiguities in non-standard words also another challenge to generate rule-based mapping schema. Hence, to consider all NSWs and to solve existence of ambiguities in NSW,(Alula, 2010) recommend to use statistical techniques such as, n-grams, Markov model, neural networks or classification and regression tree (CART).

Both (Bereket, 2008 and Bahiru, 2017) used HMM based speech synthesis technique for Amharic. But the following NLP and DSP module are not considered and recommended by both researchers. such as Factor of prosody, voice conversion, non-standard words (NSWs) and development of pro-

nunciation dictionary and syllabification rules because using only Amharic phonemes doesn't generate synthetic speech from the given Amharic text instead it does only resynthesizes the trained speech. To develop a full TTS system for Amharic it is far from prototype design we need to have both the back-end module such as Tokenization, pronunciation dictionary and the grapheme-to-phoneme rules and the front end module which consider contextual factors (using HTS embedded training). Both of the researchers cannot consider Intelligibility features such as intonation, prosody, voice characteristics, and speaking styles, or emotions, in developing the system. Intelligibility features are mainly affected by the acoustic modeling techniques and as well as the recorded speech (Tokuda et al, 1994). Due to unavailability of speech corpus (Bereket ,2008) was prepare Amharic speech corpus only for speech synthesis research the corpus holds only one speaker speech pattern and categorized as limited domain speaker dependent speech synthesizer. In contrast (Bahiru , 2017) use three male and three female speech data collected from ASR and it is only different from (Berket, 2008) by its nature of its dataset (Multi-speaker speech data).

As (Bahiru, 2017) recommend noise free speech is very important in speech synthesis contrary to speech recognition which sometimes might require speech with noise. Unlike speech recognition, noise-free-speech in speech synthesis is a priority unless the TTS synthesis is to be trained using data collected for ASR or be used with an ASR, like the same with in this study and his study. However, The corpora containing speech in a noisy environment that are designed for automatic speech recognition (ASR) have also been explored for building HMM- based TTS voices for Germany language (Valentin, 2013) by using mixed excitation method to enhance the intelligibility of synthetic speech in noise. The corpora designed for automatic speech recognition (ASR) have also been explored for building HMM- based TTS voices for English, Japanese, Indian language; in particular, (Zen et al., 2007; Ling et al., 2006; Black, 2006) built TTS voices on various ASR corpora containing cleanly-recorded read speech, as well as some corpora containing speech in a noisy environment with the goal of being able to create “thousands of voices” from the many speakers in each corpus. The sub part of the corpora,which have over 100 speakers speech data and over 20,000 utterance, designed by (Martha et al, 2014) for automatic speech recognition (ASR) for Amharic have also been used in this work for building HMM- based Amharic TTS voices. So in this work we use the mixed excitation method to enhance the intelligibility of synthetic speech in noise.

As (Bereket, 2008 ,Alula, 2010, and Bahiru, 2017) recommend creation of pronunciation dictionary with full context can also significantly positively affect the quality of synthesized speech. The creation of such a dictionary will be of great benefit to the language itself as it is difficult to find a pronunciation dictionary for Amharic. This will in turn further play a role in adding to the pool of essential components or resources of Ethiopian indigenous spoken language systems and the language itself. Incorporating pronunciation dictionary into the system will also undoubtedly better the quality of synthesized speech as it has been proven in several research articles. So in this work we incorporate pronunciation dictionary for the most frequently words in the training set and much of the work is done by adding letter -to-sound for the system.

In this work, we address inclusion of NLP module (non-standard words (NSW) such as numbers, abbreviation and punctuation marks and development of lexicon and letter to sound rules, to find pronunciation of words) and DSP module (voice conversion technique, to enhance the naturalness of the synthetic speech and mixed excitation method, to enhance the intelligibility of synthetic speech in noise. This problem will be address in the framework of HMM-based speech synthesis, by exploring the possibility to develop speaker independent TTS using HTS

1.3. Research questions

1. What is the overall performance of statistical parametric (HTS) technique for Amharic language on multiple speakers' speech data?

2. To what extent mixed excitation modeling technique enhance the intelligibility of the synthetic speech in noise?

3. To what extent Amharic Non-standard words are synthesized from the HMM-based Amharic speech Synthesizer?

1.4. General objectives

- ✓ The general objective of the study is to develop speaker independent Amharic speech synthesizer using HTS technique.

1.5. Specific objective

The specific objectives of this research work are:

1. To investigate various speech synthesis techniques
2. To assess the nature of Amharic language writing style and phonology features
3. To investigate the performance of HTS technique for Amharic language
4. To evaluate the performance of the proposed Amharic speech synthesizer

1.6. Significance of the study

The potential applications of high quality TTS Systems are indeed numerous. This work has contributed in some way to different applications that needs Amharic TTS. According to (Dutoit, 1993; Taylor, 2007; Zen et al, 2007) here are some applications of speech synthesis in general:

➔**Language education:** TTS synthesis can be coupled with a Computer Aided Learning system, and provide a helpful tool to learn language.

➔**Aid to handicapped persons:** Blind people also largely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to Amharic written information.

➔**Multimedia, Man-Machine Communication:** the development of TTS systems is a necessary step towards a means of communication between men and computers. TTS is applicable for entertainment, talking characters, proof reading, and productivity tools, on line talking assistance. Synthesized speech has been used for decades in all kind of telephone inquiry systems and may also be used to speak out short text messages (SMS) in mobile phones.

➔**Fundamental research on speech synthesis:** repeated experiences and researches on TTS for Amharic or other language provide identical results and consequently, they allow investigating the efficiency of TTS methods and models.

➔**From general purpose to specialized information retrieval systems:** TTS allows the user to query with his own voice (with the help of a speech recognizer).

1.7. Original contributions

This thesis describes techniques used to develop HMM based Amharic speech synthesizer which synthesize speech with natural human voice and with various voice characteristics such as speaker individuality and emotion. The major original contributions are as follows:

- Development of phone set for Amharic it is suitable for HMM based speech synthesis method
- Incorporating both standard and non-standard word for the Amharic HMM-based TTS system.
- Development of lexicon and letter-to-sound rules for Amharic language.
- Development of the Amharic TTS system using HTS embedded training.
- Improvement of the quality of the synthesized speech by incorporating the mixed excitation model and post filter for the Amharic HMM-based TTS system.
- Improvement of the naturalness of the synthesized speech by incorporating Voice conversion technique for the synthesizer.

1.8. Research Methodology and Tools

Before starting the actual work, deep experiment is made in the speech synthesis tools and systems that are developed for different languages. In addition deep study is made in the literature that are written on this area to have a clear picture about the work. Papers written on Amharic language are reviewed, to understand the nature of the language and how to apply to the new system. The different speech synthesis techniques have been studied to identify their differences. During the course of the thesis, we have also reviewed different speech synthesis techniques that are applied for Amharic and other languages.

The main activities that are conducted to achieve the objective are:

Data collection: To begin our data collection we selected a portion of the corpora designed by (Martha et al, 2014) for automatic speech recognition (ASR) for Amharic. The majority of existing databases have been prepared primarily with automatic speech recognition in mind. Prominent examples include TIDIGITS (isolated word recognition), SWITCHBOARD and CALL-HOME (spontaneous phone conversations), and Aurora (noisy speech) (Kominek et.al, 2003). Databases that are designed for training and testing of ASR systems require large amounts of speech collected under realistic and noisy conditions, by multiple speakers with broadly varying accents. These characteristics are not well suited for constructing synthetic voices. The HMM - based speech synthesis needs of 500 sentences to build a voice (Black ,2003). So a total of 726 prompt list or sentences ware selected and automatically divided using the festvox script *./bin/traintest etc/txt.done.data* into 654 training set and 72 test data set. Then the first 30 list of utterances are selected for voice conversion experiment using the festvox script *$FESTVOXDIR/src/vc/build_transform def_us.*

Development Methodology

Existing HMM based speech synthesis system is chosen for development of Amharic speech synthesis system. This is because; the existing system is already tested and verified by many researchers and avoids reinventing the wheel. The tool kit that is used for HMM based speech synthesis system is called HTS. HTS requires Linux operating system and a PC with high processing speed during training the model. This research is conducted on a personal computer having 500GB of hard disk, 4GB of RAM, Intel Pentium IV CPU with 2.0 GHZ processing speed. Moreover, HTS require HTK to be in place. Hence, it is also installed on the above machine. In addition, the following tools are used to develop the system.

HTS Toolkit (HTS 2.0): for building the state-of-the- art speaker-dependent and speaker-adaptive synthesizers and some voices for the Festival Speech Synthesis System.

Festival Speech Synthesis System

Many TTS systems consist of two components: front end (text normalization, grapheme-to-phoneme conversion) and back end (waveform generation). The HMM-based speech synthesis performs the back-end part only. Therefore, it is essential to combine the HMM-based speech synthesis module with a front end module in other software packages to build a complete TTS system. The Festival Speech Synthesis System is a widely used general multilingual speech synthesis system in C and Scheme (Black et al 2000). Flite (festival-lite) is a small and fast run-time synthesis engine designed for embedded devices in C (Black et al, 2001).

Speech signal processing toolkit

The Speech Signal Processing Toolkit (SPTK) is a suite of speech signal processing tools in C and shell scripts. It includes (frequency-warped) linear predictive (LP) and cepstral analysis/synthesis, vector quantization, and other extended versions of them. We can use it to extract spectral and excitation parameters and reconstruct a speech waveform from the spectral and excitation parameters generated from HMMs.

Edinburgh speech tools

The Edinburgh Speech Tools is a collection of APIs and programs for speech processing including waveform manipulation, feature extraction, and conversion. It also includes a pitch tracker, a labeling system, a classification and regression tree (CART), and support for linguistic type objects. This tool is used in the Festival Speech Synthesis System. ESPS software is a suite of signal processing programs that can be used for the analysis, manipulation, and labeling of speech. It also includes the stand-alone version of get_f0.

Transcription editor

EHMM Labeler is segmentation editor used in the Festival Speech Synthesis system. It can read and write HTK style transcription files and provides an boot strap to manually edit segmentation labels. The use of manually corrected segmentation labels sometimes improves the quality of synthesized speech.

Evaluation Techniques: Both objective Evaluation is made to compare the model and Subjective Evaluation using Measure perceptual test (MOS) was conduct to evaluate the proposed Amharic speech synthesizer overall performance.

1.9. Thesis overview

The remainder of this thesis is organized as follows:

- Chapter 2 reviews concepts related to TTS. It also provides human speech production system, overview of speech synthesis, different techniques of speech synthesis, review works done to design TTS system for Amharic language and descriptions about the techniques and metrics used in the thesis.
- Chapter 3 goes over the details on both Amharic language phonologies and writing system.
- Chapter 4 describes a basic HMM based text to speech synthesis techniques based on prior works which inspires the work in this thesis.
- Chapter 5 describes the process and experiment conducted to prototype design of TTS for Amharic language. And how data selection and different techniques plays a significant role in performance. Finally the experimentation and testing result of the proposed Amharic speech synthesizer is discussed.
- Chapter 6 summarizes the key concepts of the thesis and provides directions for future work.

Chapter 2

Literature review

In this chapter, the theoretical basis of speech synthesis, models of speech production and how these are used in text-to-speech conversion are reviewed. In the first section of the chapter the introduction and overview of text to speech components is present. In the second part of the chapter the history and development of synthesized speech and speech production theory and speech synthesis process is discussed. The third part presents a theoretical model or approaches of speech production, which are the basis of most synthesizers and their relative merits and shortcomings are considered. The chapter ends with a detail look at techniques that are being used for Amharic and other local languages.

2.1. Introduction

Speech is the primary means of communication between people. Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware *(Klatt, 1987).* A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. The ultimate goal of speech synthesis or text-to-speech synthesis is to convert ordinary orthographic text into an acoustic signal that is indistinguishable from human speech *(Breen 1992; Flanagan, et al, 2008).*

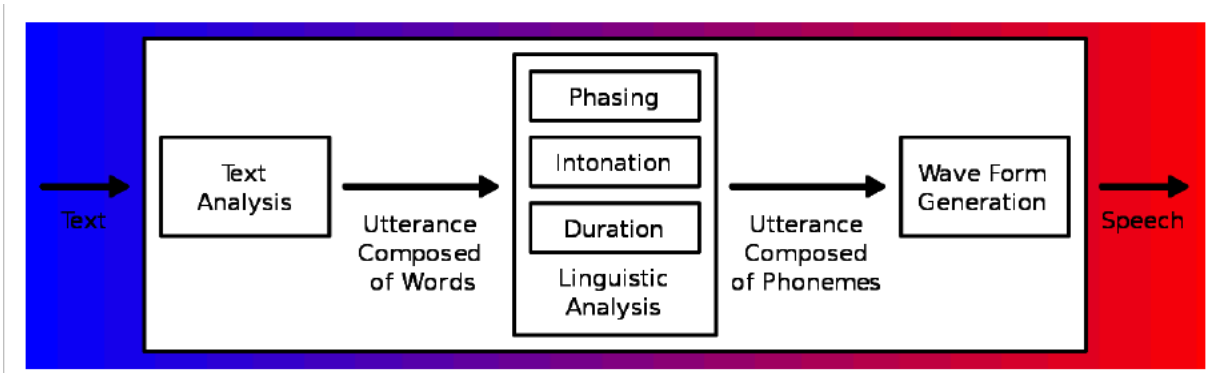


Figure 2-1 Text-to-speech synthesis conversion (Santen et al, 1997)

Figure 2.1 and 2.2 illustrate the block diagram of Sentence to speech conversion module it converts the symbolic linguistic or the phonetic transcription representation into a speech

waveform containing appropriate values for acoustic parameters such as pitch, amplitude, duration, and spectral characteristics (*Santen et al, 1997).* The first one handles problems in text analysis and higher level linguistic features. This process is often called text normalization, preprocessing, or tokenization; and the second one handles problems in phonetics, acoustics and signal processing; these two phases are usually called as front end and back end or synthesis *(Klatt, 1987).*

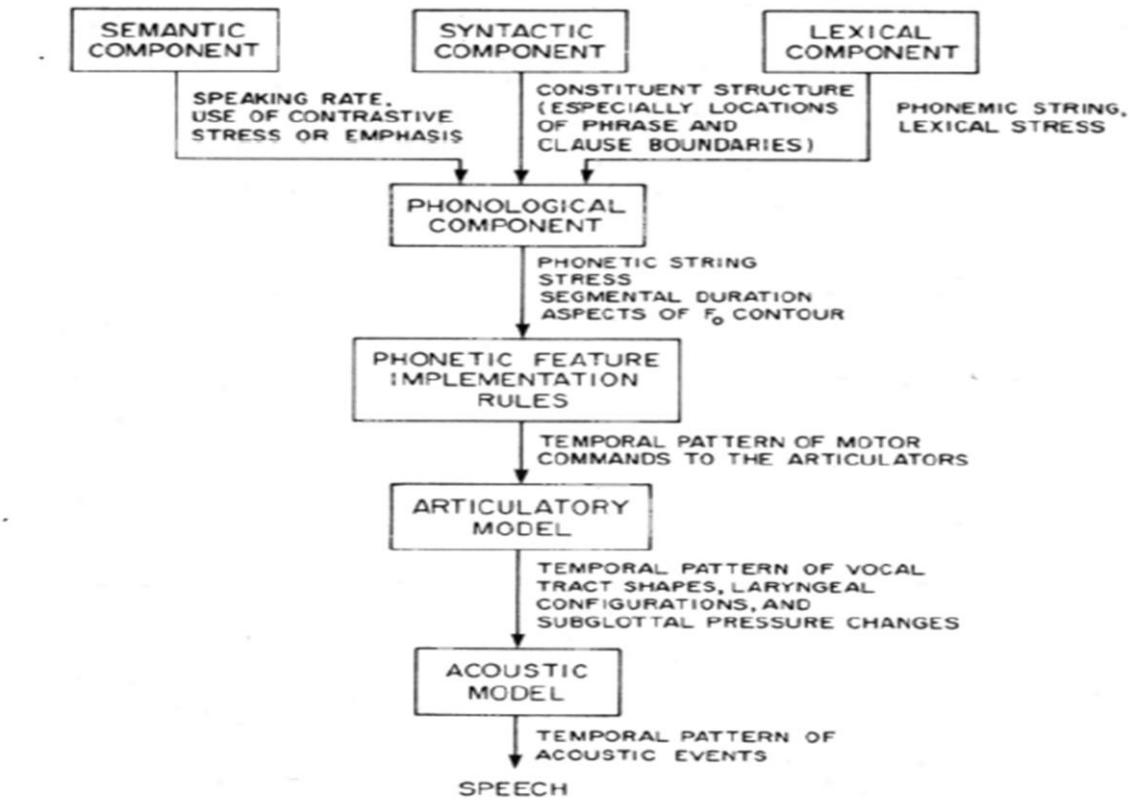


Figure 2-2 Simplified block diagram of the back end module DSP module (klatt, 1987)

2.2. History and Development of Speech Synthesis

To understand how the present systems work and how they have developed to their present form, a historical review may be useful. In this section, the history of synthesized speech from the first mechanical efforts to systems that form the basis for today's high-quality synthesizers is discussed. Some separate milestones in synthesis-related methods and techniques will also be discussed in detail.

2.2.1. From Mechanical to Electrical Synthesis

Attempts at making mechanical speaking machines date back to two hundred years ago. One of the most successful devices of that period was produced by Russian Professor Christian Frankenstein in St. Petersburg 1779 *(Lemmetty, 1999)*. Christian Frankenstein contracted acoustic resonators similar to the human vocal tract and activated the resonators with vibrating reeds like in music instruments based on physiological differences between five long vowels (/a/, /e/, /i/, /o/, and /u/) and made apparatus to produce them artificially. A few years later, in Vienna 1791, Wolfgang Von Kempelen introduced his "Acoustic- Mechanical Speech Machine", which was able to produce single sounds and some sound combinations. Briefly, it consisted of a bellows which fed a reed connected to a leather cylinder which acted as a vocal tract. Air was expelled from the bellows using the left hand and the shape of the leather vocal tract was modified by the right hand (Breen, 1992). The essential parts of the machine were a pressure chamber for the lungs, a vibrating reed to act as vocal cords, and a leather tube for the vocal tract action. By manipulating the shape of the leather tube he could produce different vowel sounds. Consonants were simulated by four separate constricted passages and controlled by the fingers. For plosive sounds he also employed a model of a vocal tract that included a hinged tongue and movable lips. His studies led to the theory that the vocal tract, a cavity between the vocal cords and the lips, is the main site of acoustic articulation. Before Von Kempelen's demonstrations the larynx was generally considered as a center of speech production. Kempelen received also some negative publicity. While working with his speaking machine he demonstrated a speaking chess-playing machine. Unfortunately, the main mechanism of the machine was concealed, legless chess-player expert. Therefore his real speaking machine was not taken so seriously as it should have (Lemmetty,1999). In about mid 1800's Charles Wheatstone constructed his famous version of Von Kempelen's speaking machine.

In late 1800's Alexander Graham Bell with his father, inspired by Wheatstone's speaking machine, constructed same kind of speaking machine. Bell made also some questionable experiments with his terrier. He put his dog between his legs and made it growl, then he modified vocal tract by hands to produce speech-like sounds. The research and experiments with mechanical and semi-electrical analogs of vocal system were made until 1960's, but with no remarkable success (Donovan, 1996). It was a bit more complicated and was capable to produce vowels with vibrating reed and all passages were closed and most of the consonant sounds produce with turbulent flow through a suitable passage with reed-off. Resonances were affected

by deforming the leather resonator like in Von Kempelen's machine

2.2.2 Development of electrical synthesizers

The first full electrical synthesis device was introduced by Stewart in 1922. The synthesizer had a buzzer as excitation and two resonant circuits to model the acoustic resonances of the vocal tract. The machine was able to generate single static vowel sounds with two lowest formants, but not any consonants or connected utterances. Same kind of synthesizer was made by Wagner. The device consisted of four electrical resonators connected in parallel and it was excited by a buzz like source. The outputs of the four resonators were combined in the proper amplitudes to produce vowel spectra. In 1932 Japanese researchers Obata and Teshima discovered the third formant in vowels *(Breen, 1992).*

Mechanical speaking machines were replaced with the advent of electrical technology by electrical devices, the First device considered as a speech synthesizer and one notable early version of which was the VODER (Voice Operating DEmonstratoR), first demonstrated at the world fair of 1939 in New York by Homer Dudley. The VODER worked by exciting a set of fixed filters, which acted as resonators, which chosen to produce a particular sound were controlled by a human operator. The VODER consisted of wrist bar for selecting a voicing or noise source and a foot pedal to control the fundamental frequency. The source signal was routed through ten band pass filters whose output levels were controlled by fingers. It took considerable skill to play a sentence on the device. It was finally shown that intelligible speech can be produced artificially. Actually, the basic structure and idea of VODER is very similar to present systems which are based on source-filter-model of speech. However, the speech quality and intelligibility were far from good but the potential for producing artificial speech were well demonstrated and it was only with the formulation of a mode1 of speech production that speech synthesizers as they are today appeared *(Breen, 1992)*. About a decade later, in 1951, Franklin Cooper and his associates developed a Pattern Playback synthesizer at the Haskins Laboratories. The four first formants are generally considered to be enough for intelligible synthetic speech. The first formant synthesizer, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953. First articulatory synthesizer was DAVO (Dynamic Analog of the Vocal tract), introduced in 1958 by George Rosen at the Massachusetts Institute of Technology, M.I.T. *(Allen et al, 1987).* The first full text-to-speech system for English was developed in the Electro

technical Laboratory, Japan 1968 by Nariko Umeda and his companions Klatt. It was based on an articulatory model and included a syntactic analysis module with sophisticated heuristics. The speech was quite intelligible but monotonous and far away from the quality of present systems. In 1979 Allen, Hunnicutt, and Klatt demonstrated the Mi Talk laboratory text-to-speech system developed at M.I.T *(Allen et al, 1987).*

Modern speech synthesis technologies involve quite complicated and sophisticated methods and algorithms. One of the methods applied recently in speech synthesis is hidden Markov models (HMM). HMMs have been applied to speech recognition from late 1970's. For speech synthesis systems it has been used for about two decades (*Zen et al, 2007).* HMM Hidden Markov model (HMM) based speech synthesis has been widely used in recent years. In this method, the frequency spectrum, pitch and duration of speech are modeled simultaneously within a unified framework during the training procedure. At synthesis stage, speech waveforms are reconstructed using acoustic features predicted from trained HMMs by maximum likelihood parameter generation (MLPG) algorithm (*Zen et al, 2009; Tamura et al, 2001).*

Neural networks have been applied in speech synthesis for about ten years and the latest results have been quite promising. However, the potential of using neural networks have not been sufficiently explored. Like hidden Markov models, neural networks are also used successfully with speech recognition *(Valentin, 2013).*

2.3. Speech production system

Speech processing and language technology contains lots of special concepts and terminology. To understand how different speech synthesis and analysis methods work we must have some knowledge of speech production, articulatory phonetics, and some other related terminology. The basic theory of these topics is discussed in this section.

2.3.1. Human speech production

Articulatory phonetics attempts to describe the production of the linguistically important sounds of a language in terms of the vocal organs. Most speech sounds are normally produced on a pulmonary regressive air stream, in other words air is expelled from the lungs through muscular action. Human speech is produced by vocal organs presented in figure 2.3. The main energy source is the lungs with the diaphragm. Air leaving the lungs passes through a body of

interlocking cartilage called the larynx .The two major components of the larynx are the thyroid cartilage and the colloid cartilage. In men the thyroid cartilage is set at a slight angle, the front of which is commonly called the 'Adam's apple'. Within the thyroid and corridor cartilages are the vocal folds (figure 2.3), which consists of two fleshy membranes attached by the vocalist muscle to the inside wall of the thyroid cartilage. The tension of the vocal folds is mainly governed by the vocalist muscles which form the body of the vocal folds. The front of the vocal folds are brought together and attached to the thyroid cartilage, while at the back they are attached to a pair of small cartilages called the adenoids (*Tamura et al, 2001).*

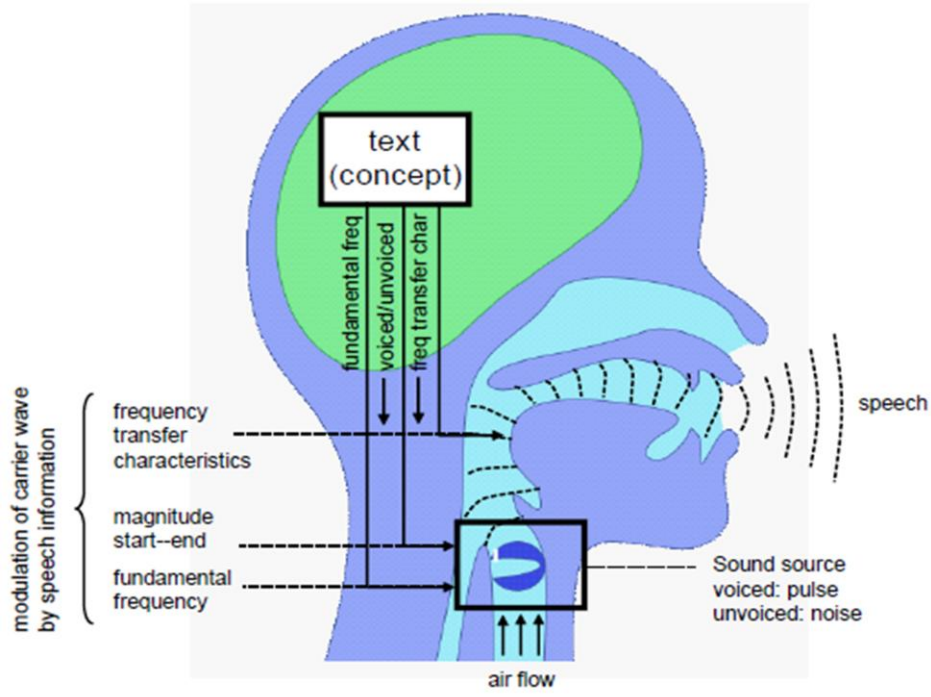


Figure 2-3 Outline of speech production process (Zen.2007)

During normal breathing the vocal folds are abducted (held apart) allowing air to pass freely through the gap between the two folds (termed the glottis). During voiced speech (Phoenician) the vocal folds are repeatedly brought together and forced apart. The tension in the folds is adjusted by tilting the thyroid cartilage, which is used to control the fundamental period of vocal fold oscillation. A fundamental period can be divided simplistically into two portions, an open-glottis cycle (open phase) and a closed-glottis cycle. During the closed portion of the glottal cycle a pressure difference builds up between the pressure in the lungs and trachea and the external atmospheric pressure. The sub-glottal pressure on the folds forces them to move apart.

Once they start to move apart air passes through the glottis. The particle velocity of the air through the glottis is high and a Bernoulli force is induced which, in conjunction with the muscular tension in the folds, tends to draw the folds hack together, eventually closing the glottis. This procedure is repeated over and over again. The vocal cords may act in several different ways during speech. The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which vowels and voiced consonants are produced (*Santen et al, 1997)*. The theory is called the myelitides/aerodynamic theory of Phoenician. The time between each closure of the vocal folds is called the fundamental period *To*, the reciprocal of which is the fundamental frequency F0. Pitch and fundamental frequency are not synonymous but are often used interchangeably owing to their close correspondence. Fundamental pitch is a tonal sensation as perceived by a human listener whereas fundamental frequency is a property of the physical system. In contrast, voiceless sounds are produced when the vocal folds are sufficiently abducted to allow air to pass relatively unimpeded through the glottis.

The sound pressure wave above the larynx is modified by the vocal tract either by modifying the spectral larynx, at some point of constriction within the vocal tract such as a voiceless fricative distribution of the energy in the sound wave, or by generating sound within the vocal tract**,** Voiced sounds are produced at the larynx: voiceless sounds however are normally produced above the larynx, at some point of constriction within the vocal tract. sound such as Is1 in ‘six’ is produced by turbulent air flow past a constriction made between the blade of the tongue and the roof of the mouth at the alveolar ridge .In articulatory phonetics the consonant sounds of a language are described using three variables: voice unvoiced and consonant. With stop consonants the vocal cords may act suddenly from a completely closed position, in which they cut the air flow completely, to totally open position producing a light cough or a glottal stop (DSP). On the other hand, with unvoiced consonants, such as /s/ or /f/, they may be completely open (Klatt,1987; Breen,1992). From technical point of view, the vocal system may be considered as a single acoustic tube between the glottis and mouth. Glottal excited vocal tract may be then approximated as a straight pipe closed at the vocal cords where the acoustical. All speech synthesizers assume an underlying model of speech production. Such models can be best appreciated if described in parallel with a brief description of the human mechanism of speech production and the characteristics of the speech signal (*Santen et al, 1997)*. Speech production and phonetics for Amharic language will be discussed in chapter 3.

2.4. Methods, techniques, and algorithms

All speech synthesizers assume an underlying model of speech production. Such models can be best appreciated if described with a detailed description of the human mechanism of speech production and the characteristics of the speech signal. Speech can be generated from text in a variety of ways. All in general, are classified into three categories: rule-based, data-driven, and model-based methods (Valentin , 2013).

2.4.1. Rule-based speech synthesis

The first TTS methods proposed were constructed by rules on how speech sounds are produced. Rule-based TTS can be implemented in two different ways. The first way generated Speech by following rules on the realization of acoustic components like the formants and the fundamental frequency (formant synthesizers), Rule-based formant synthesis, which is based on the source-filter-model of speech production, uses the rules to modify the pitch, formant frequencies, and other parameters from one sound to another while maintaining continuity present in physical systems like the human production system. The method models only the source of the sound and the formant frequencies, not any physical characteristics of the vocal tract. The second Rule based method is articulatory synthesizers, which is implemented based on any physical characteristics of the vocal tract like the position of the articulators. Rule-based TTS systems are created from prototypical rules of speech production that can create intelligible but very unnatural voices (Valentin, 2013). The parameterization of production enables controllability, however devising rules for formant and articulator placement manually requires a great deal of expert knowledge. formant requires small linguistic resources and able to generate various speaking styles. However, this method produced less natural-sounding synthesized speech and the complex rules required to model the prosody is a big problem *(Zen et al, 2007;Valentin, 2013).*

2.4.2. Data driven based speech synthesis

Instead of following production rules, the next generation of TTS systems uses Data Driven approach, it create speech from the concatenation of natural speech components, are derived

during the training of the system, from a large database of several hours of speech. Concatenation based speech synthesis technique uses various length of pr-recorded voices derived from natural speech. By connecting pr-recorded natural utterances, the intelligible and natural sounding synthetic speech will be produced. Concatenation systems were first proposed in the shape of fixed component units, Di-phone synthesizers *(Valentin, 2013).* A di-phone is a segment defined from the middle of one phone to the middle of the subsequent phone. These segments were represented by linear predictive analysis components extracted during training more storage and computing power became available, the second generation of concatenation systems appeared: unit selection systems. In unit selection, the segments units of concatenation are variable in size. Unit-selection is the dominant method in speech synthesis**.** Due to performance advantages such as high quality, and naturalness of synthetic speech. However unit-selection systems are highly dependent on the database and the quality of the recorded database. Due to this quality dependency, voice modification at the selected units cannot be carried out, and voice conversion/ adaptation is a difficult task by the time being, for unit selection systems *(Yoshimura et al, 1999).* Furthermore, databases where perfect recording conditions are not possible to achieve cannot be used. Additionally big storage memory is necessary, which is prohibitory in specific applications and particularly, for languages with limited linguistic resources, it is more difficult. Because of these limitations much research has moved to the third Generation called statistical parametric speech synthesis and mainly to Hidden Markov Models (HMM)-based systems (*Tokuda, et al., 1999).*

2.4.3. Statistical parametric based speech synthesis

Proposed at the end of the 1990s, another paradigm for creating speech from text appeared based on units derived from statistical models, the statistical parametric TTS systems *(Yamagishi, J ,et al., 2009: Zen et al, 2007).* At synthesis time, the models are used to generate a low dimension parametric representation of speech. Instead of storing a large database of units this system represents units of speech by model parameters of lower dimensionality. The most widely used statistical model for statistical parametric TTS is the hidden Markov model (HMM), creating what is referred to the HMM-based speech synthesis systems. HMMs are used in other areas of speech technology like speech enhancement, conversion and quite extensively in the field of automatic speech recognition. Advances in this field led to many different methods and criteria for training, clustering and adapting HMMs, alongside freely available toolboxes such as HTK.

Using statistical models as a choice for acoustic modeling has also influenced research in parametric representations of speech that can offer good interpolation and compression properties.

Text-to-speech can be viewed as a sequence-to-sequence mapping problem; from a sequence of discrete symbols (text) to a real valued time series (waveform). Typical TTS systems consist of text analysis and speech synthesis parts. The text analysis part includes a number of natural language processing (NLP) steps, such as word segmentation, text normalization, part-of-speech (POS) tagging, and grapheme-to-phoneme (G2P) conversion. This part performs a mapping from a sequence of discrete symbols to another sequence of discrete symbols (e.g sequence of characters to sequence of words). The speech synthesis part performs mapping from a sequence of discrete symbols to real-valued time series. It includes prosody prediction and speech waveform generation. The former and latter parts are often called "front-end" and "back-end" in TTS, respectively. Although both of them are important to achieve high-quality TTS systems, this section focuses on the latter one. Statistical parametric speech synthesis (SPSS), is one of the major approaches in the back-end part (*Yoshimura,2002; Zen et al, 2007).*

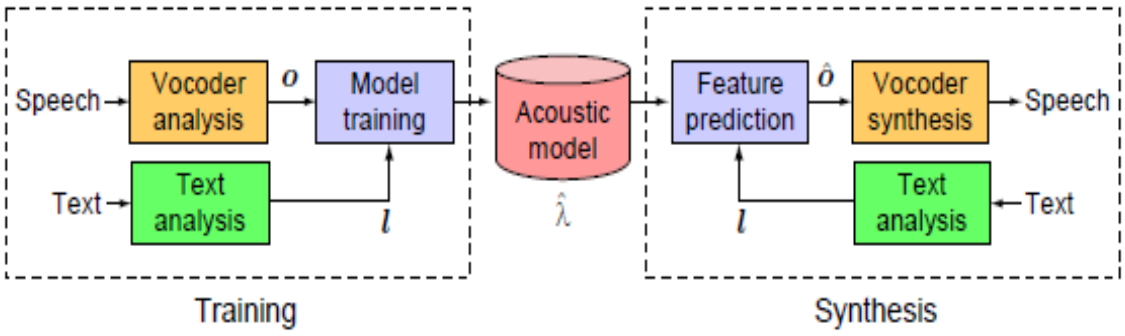


Figure 2-4Typical Statistical parametric TTS (Zen et al. 2009)

Statistical parametric based speech synthesis involves generating speech from a linear sequence of models, in which each model corresponds to a particular linguistic unit type. Therefore, it is necessary to flatten the structured linguistic specification into a linear sequence of labels. This is achieved by attaching all the other linguistic information (about syllable structure, prosody, etc) to the phoneme tier in the linguistic specification – the result is a linear sequence of context-dependent phonemes. Given these full context labels, the corresponding sequence of HMMs can be found, from which speech can be generated. According to (*Valentin, 2013),* **S**tatistical

parametric-based speech synthesis presents many advantages over the other TTS paradigms:

- ✓ Generalization: wider coverage of the acoustic space.
- ✓ smaller footprint: storage of the statistics of acoustic models rather than waveform templates;
- ✓ versatility: new voices of different speakers and speaking styles can be easily obtained by transforming model parameters through well-established model adaptation techniques that require small amounts of additional recordings;
- ✓ robustness: the quality of generated speech is more robust to variability in recording conditions and speaking quality as reported in Yamagishi et al. [Yamagishi, J. 2006];
- ✓ unified learning: text and acoustic analysis can be jointly performed in an unified statistical approach;
- ✓ controllability: parameterization and context dependency allow for localized control strategies;
- ✓ Multilingual support: a large recording database of a particular language is not required to build a voice with good quality.

2.5. Related works on speech synthesis for local language

There are few research works based on formant ,unit selection and statistical parameter methods for Ethiopian languages in general and some of them are in detail discussed below.

The first text to speech synthesis for amharic language was done by (*Laine,1998)* in *1998* using diphone based concatenate speech synthesis technique. Di-phones were used as the basic concat-enation units and he used the linear predictive coding method to produce the synthesized speech and he used interpolation scheme to minimize the discontinuous nature of the synthesized speech, tools used were pascal and mat-lab, and the evaluation reported as good. Furthermore, *Laine (2003)* noted that prosodic information was not considered in his work.

Henok (2003) did the next attempt in 2003 using a concatenation speech synthesis approach for Amharic language. Tools and techniques includes TD-PSOLA technique for smoothing, Pratt for spectrograph analysis; and Delphi and MATLAB for prototype development. He included prosodic information as recommended by *Laine (2003).* Evaluation used included Open Rhyme Test (ORT),

and Mean Opinion Score (MOS), and result was reported promising.

Sebsibe et al. (2005), discuss the development of unit selection voice for Amharic using Festvox they defined a transliteration scheme to work with Amharic scripts and incorporated Amharic phone set, syllabification rules, letter to sound rules into Festvox. However, the unit selection techniques do not permit for adjustment of the TTS system to diverse speaking styles and speaker characteristics and requires databases of broad sizes. To evaluate the quality of Amharic synthesizer, they conducted perceptual tests on 11 college students (2 females and 9 males) 20 to 30 years old native speakers of the Amharic language and the average score of the proposed Amharic synthesizer was obtained 2.9 (which is categorized as good). However, the unit selection techniques do not permit for adjustment of the TTS system to diverse speaking styles and speaker characteristics and requires databases of broad sizes.

The first formant based speech synthesis for Amharic vowels was done by Nadew (2008) in 2008 using MATLAB. The focus was on vowels since vowels play a big role in change of pronunciation of a word in different contexts. The developed system was generate vowels from a given context, by best selection of parameters from the decision tree using the CART machine, it selects best parameters from the database that exactly or closely matches to the contexts of the input vowel. Nadew (2008) recommend the refinement of the work to including consonant consideration, and preparation of appropriate speech corpus the same time the acoustic parameters of the file will be passed to the formant synthesizer to synthesize vowels from the given context of a word. Result indicated intelligibility of 88.85% for isolated vowels using MOS.

Alula (2010) attempt to develop a generalized Amharic Text-To-Speech (TTS) synthesis based on diphone unit concatenation synthesis to handle both Amharic standard and non standard word. he used Residual Excited Linear Predictive (RELP) coding method. he constructed Diphone database used as a base store for Amharic data, such as phone-sets, text utterances and recorded diphone sounds. The performance of the system shows on the average an accuracy level of 73.75% for Amharic text containing both non standard word and standard word. In addition, the system performance was evaluated by adopting the Mean Score Opinion (MOS) and achieved 3 and 2.8 MOS score for intelligibility and naturalness respectively. The experiment shows a promising result to design an applicable system that synthesis both non standard word and standard word for unrestricted text of a language. He finally recommend to use statistical

techniques such as, n-grams, Markov model, neural networks or classification and regression tree (CART) to consider all non standard word during text normalization, building Grapheme-to-Phoneme (G2P) rule to handle unknown words and prosody feature led to better quality Speech synthesizer, which take in to account speaker specific intonations and speaker specific duration and with prosody analysis building part of speech is crucial step.

Tadesse et al (2010), presents the development of Amharic Text-to-Speech system (Amharic TTS) based on a spectral method, combination of a parametric and rule-based Approach. They use a source filter model for speech production and a Log Magnitude Approximation (LMA) filter as the vocal tract filter. Tadesse (2010) describes an architecture, a preprocessing morphological analyzer integrated into an Amharic text to speech system, to convert Amharic Unicode text into phonemic specification of pronunciation. The study mainly focused on disambiguating gemination and vowel epenthesis which are the significant problems in developing Amharic TTS system. They presented preliminary results on a method for automatic assignment of geminates and epenthetic vowel in GTP conversion for Amharic TTS system. A preliminary evaluation of the proposed automatic geminate assignment method was made by analyzing 666 words and they found 100% correct assignment/restoration of gemination. for more accurate GTP conversion. They recommend parts-of-speech (POS) tagger and phrase break predictor needs to be implemented or addressed.

The same work was done by Eyob (2011) in 2011 he used concatenation speech synthesis for Amharic using unit selection method , in his project work he tried to address epenthesis and germination in having as many allophones as possible and identifying contexts that determine allophonic variations. In his project, At least two allophonic variations for each phoneme in the Amharic alphabet are built; and epenthesis, gemination and interrogative prosody modeling rules to make appropriate selection of those variations from context are also used. The performance of the system was evaluated by using the MOS techniques and generally the system achieved cumulative result of the naturalness of the system was 3.63 and its intelligibility was 3.53

Mulat (2012) describes the design of a syllable based concatenation speech waveform synthesizer for Amharic language using TD-PSOLA algorithm he applies syllable based concatenation speech synthesis approach to design TTS for Amharic language. He used Time-Domain Pitch Synchronous Overlap and Add (TD-PSOLA) algorithm for the prosodic

modification and speech waveform analysis/synthesis purpose and Di phones and syllables were used as the basic concatenation units to synthesize sample. The system was obtained 73.75% and 89.58% using ORT test result for intelligibility for transcribed and syllabified texts and mean average score of 3.45 using MOS test result for the naturalness for transcribed and syllabified texts. Finally He recommend The integration of the input module (automatic syllabification and prosody generation) with the rest of the synthesis system (synthesis module) , Spectral continuity measures, to predict the audible discontinuities of the Amharic syllable boundaries and marks ,in the case of waveforms generation phase and including prosodic Features such as Speech emotion development for different type emotions and correct assignment of stress to Amharic words was recommended by the researcher.

The first attempt using Hidden Markov Model was done by Bereket (2008) to develop a speech synthesizer for Amharic language. The utterance structure generated by festival and festvox together with the parameters extracted from the raw wave data were used for training the model. The speech parameters used for training the model are Mel-spectrum coefficients and fundamental frequencies. In this research work the text that is going to be synthesized was assumed to be normalized, that is, all the preprocessing activities are done before it is given to the synthesizer. Finally, the synthesized speech is generated from the trained model based on the input text. Technique was used to test the performance of the system: namely Mean Opinion Score. In this technique, respondents were given speech synthesized by HTS-FA and data driven approach (concatenation method) and then they gave a rank for each sentence for different criterion. Based on the value of the MOS, HTS-FA performs better than that of data driven approach for both naturalness and intelligibility criteria. The performance of the system in generating intelligible speech is also good as per the result of MOS test. Recent progress in speech synthesis has produced synthesizers with very high intelligibility but the sound quality and naturalness still remain a major problem.

Tesfay (2004) attempted the first text to speech for Tigrigna language using diphone based concatenation approach with MATLAB. The performance of the synthesizer measured using MOS was 3.05. Inclusions of acronym converter to the text processing module and prosody control are some of the things that the researcher recommends for further work.

Agazi et al (2012) describe the development of the first unit selection based Text-to-Speech (TTS) system for Tigrinya using the Festival framework and practical applications of it. They describe the implementation and evaluation of a G2P conversion model for a Tigrigna TTS system. The major tasks they have been performed, via development of concatenation Unit selection voice using phone as basic unit. they used a speech corpus having a size of 4 hour, 38 minutes and 29 seconds, labeled at phoneme level. The system was evaluated using MOS and The system achieves on average, 97.1% of the sentences are correctly recognized by the listeners. The naturalness of the synthesized speech demonstrates the appropriateness of the proposed approach. Finally they recommend inclusion of Syllabification of words, Automatic gemination and epenthesis handling algorithm, Deep studies on syllabification and final consonant cluster, Stress assignment algorithm , Tigrinya morphological analyzer, Duration modeling of consonants and vowels and proper identification of Tigrinya stress point to increase the quality of the synthesized speech.

Lemlem et al (2014) develop text to speech synthesizer for Tigrigna language . Tools and techniques include using the festival speech synthesis for prototype development. they used Mean Opinion Score (MOS) Evaluation technique, the intelligibility of the proposed synthesizer was achieve promising result in intelligibility and the accuracy of correctly pronounced syllables was 89.76%.

Bahiru (2017) used ASR corpus to develop a syllable based speech synthesis system for Amharic language using Hidden Markov Model. He used the text and speech corpus with the size of 600 and he split it to 550 (90%) for training and the rest 50 (10%) for testing data sets. He considered only the characteristics and way of creation of Amharic phonemes. He used Mean Opinion Score (MOS) Evaluation technique, the intelligibility of the proposed synthesizer was achieve on the overall performance of 75.56% for syllable based and 77.78% for phone based system.

As a result both Bereket, (2008) and Bahiru (2017) do not consider abbreviated words, numbers, and punctuation marks. in particular (Alula, 2010) uses the rule-based mapping process to convert non-standard words to their equivalent standard words. he used non-standard words (NSWs) and standard words (common words and proper names) to build the system. But, the inconsistency of usage of non-standard words and existence of ambiguities in non-standard words also another challenge to generate rule-based mapping schema. Hence, to consider all NSWs and to

solve existence of ambiguities in NSW, he recommend to use statistical techniques such as, n-grams, Markov model, neural networks or classification and regression tree (CART).

Both (Bereket, 2008 and Bahiru, 2017) used HMM based speech synthesis technique for Amharic. But the following NLP and DSP module are not considered and recommended by both researchers. such as Factor of prosody, voice conversion, non-standard words (NSWs) and development of pronunciation dictionary and syllabification rules because using only Amharic phonemes doesn't generate synthetic speech from the given amharic text instead it does only resynthesizes the trained speech.

To develop a full TTS system for Amharic it is far from prototype design we need to have both the back-end module such as Tokenization, pronunciation dictionary and the grapheme-to-phoneme rules and the front end module which consider contextual factors (using HTS embedded training). In addition both of the researchers cannot consider Intelligibility features such as intonation, prosody, voice characteristics, and speaking styles, or emotions, in developing the system. Intelligibility features are mainly affected by the acoustic modeling techniques *and* as well as the recorded speech *(Tokuda et al, 1994)*. Due to unavailability of speech corpus (Bereket ,2008) was prepare Amharic speech corpus only for speech synthesis research the corpus holds only one speaker speech pattern and categorized as limited domain speaker dependent speech synthesizer. In contrast (Bahiru ,2017) use three male and three female speech data collected from ASR.

As (Bahiru, 2017) recommend noise free speech is very important in speech synthesis contrary to speech recognition which sometimes might require speech with noise. Unlike speech recognition, noise-free-speech in speech synthesis is a priority unless the TTS synthesis is to be trained using data collected for ASR or be used with an ASR, like the same with in this study and his study. However, The corpora containing speech in a noisy environment that are designed for automatic speech recognition (ASR) have also been explored for building HMM- based TTS voices for Germany language (Valentin, 2013) by using mixed excitation method to enhance the intelligibility of synthetic speech in noise. The corpora designed for automatic speech recognition (ASR) have also been explored for building HMM- based TTS voices for English, Japanese, In-

dian language; in particular, (Zen et al., 2007; Ling et al., 2006; Black, 2006) built TTS voices on various ASR corpora containing cleanly-recorded read speech, as well as some corpora containing speech in a noisy environment with the goal of being able to create “thousands of voices” from the many speakers in each corpus.

Chapter 3

Amharic language phonology

This chapter presents the nature of Amharic language linguistic features and how to transcribe to its phoneme sound representation. The first part introduces Amharic language and discusses the general sound production system with special reference to Amharic language. The second and Third sub sections deals about phonetics and Amharic writing system that are related to the development of text to speech synthesis.

3.1. Overview of Amharic language

Amharic (አማርኛ) is an Afro-Asiatic language of the Semitic branch. The language serves as the official working language of Ethiopia, and is also the official or working language of several of the states within the federal system. Amharic is the second-most widely spoken Semitic language in the world after Arabic. Amharic is spoken by 22 million native speakers in Ethiopia (Ethnologue,2016). Additionally 3 million emigrant outside of Ethiopia speak the language . It is written (left-to-right) using Ethiopic Fidel, ፊደል, which grew out of the Ge'ez abugida —called, in Ethiopian Semitic languages , ፊደል fidel ("writing system", "letter", or "character") and አቡ ጊዳ abugida (from the first four Ethiopia letters, which gave rise to the modern linguistic term abugida). The scripts are more or less orthographic representation of the phonemes in the language (Baye,1997; Getahun ,2001)

3.2. Phonetics

In most languages the written text does not correspond to its pronunciation so that in order to describe correct pronunciation some kind of symbolic presentation is needed. Every language has a different phonetic alphabet and a different set of possible phonemes and their combinations. The number of phonetic symbols is between 20 and 60 in each language. A set of phonemes can be defined as the minimum number of symbols needed to describe every possible word in a language. In English there are about 40 phonemes. Due to complexity and different kind of definitions, the number of phonemes in English and most of the other languages can not be defined exactly *(Donovon, 1996).*

Phonemes are abstract units and their pronunciation depends on contextual effects, speaker's characteristics, and emotions. During continuous speech, the articulatory movements depend on

the preceding and the following phonemes. The articulators are in different position depending on the preceding one and they are preparing to the following phoneme in advance. This causes some variations on how the individual phoneme is pronounced. These variations are called allophones which are the subset of phonemes and the effect is known as co-articulation. the same phoneme but different allophones and have different vocal tract configurations. Another reason why the phonetic representation is not perfect is that the speech signal is always continuous and phonetic notation is always discrete and different emotions and speaker characteristics are also impossible to describe with phonemes so the unit called phone is usually defined as an acoustic realization of a phoneme *(Taylor, 2007).*

3.3. Amharic articulatory phonetics

The phonetic alphabet is usually divided in two main categories, vowels and consonants. Vowels are always voiced sounds and they are produced with the vocal cords in vibration, while consonants may be either voiced or unvoiced. Vowels have considerably higher amplitude than consonants and they are also more stable and easier to analyze and describe acoustically. Because consonants involve very rapid changes they are more difficult to synthesize properly. Overview of each of these major categories of Amharic phonemes is given in the following section.

3.3.1. Amharic vowel

Amharic has a set of 39 phones, seven vowels and thirty-two consonants, makes up the complete inventory of sounds for the Amharic language. There are seven vowels in Amharic. These vowels can be divided into different categories depending how they are formulated: Front/back position of tongue, wideness/roundness of the constriction position, place of the tongue (high or low), and how open or close the mouth is during articulation (Baye, 1997). Amharic vowels (IPA) and their categorization are shown in figure 3.1.

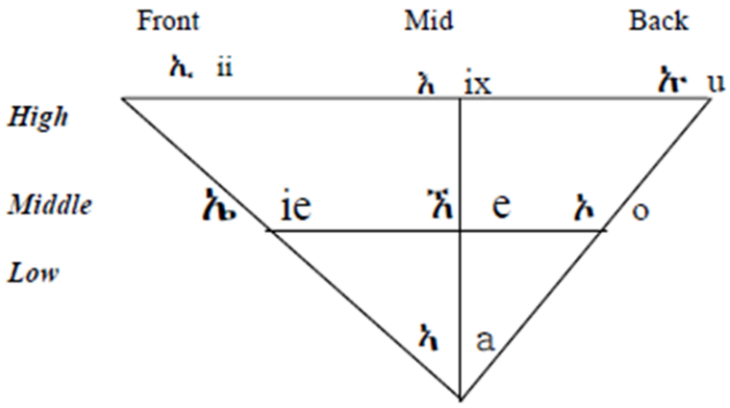


Figure 3-0-1Amharic Vowels (sebsebi .et.al 2004)

3.3.2. Amharic consonants

Amharic consonants are generally classified as stops, fricatives, nasals, liquids, and semi-vowels Further classification may be made based on the place of articulation as labials (lips), dentals (teeth), alveolar (gums), palatals (palate), velars (soft palate), glottal (glottis), and labiodental (lips and teeth) (*Getahun ,2001)*. Table3.1 shows the phonetic representation of the consonants of Amharic as to their manner of articulation, voicing, and place of articulation. Amharic consonants can be divided as stops, fricatives, nasals, liquids, and semi-vowels into the following categories depending on the place and the manner of articulation *(Baye,1997; Sebsebi et al,2005)*:

1. Plosive or stop consonants: /*k, p, t, g, b, d*/./ ከ ፐ ት ብ ድ ግ ጓ ጵ/ The vocal tract is closed causing stop or attenuated sound. When the tract reopens, it causes noise-like, impulse-like or burst sound.

2. Fricatives: /*f, h, s*/ / ፈ ሀ ሰ ሸ ዠ ዘ/. The vocal tract is constricted in some place so the turbulent air flow causes noise which is modified by the vocal tract resonances. Amharic fricatives are unvoiced.

3. Nasals: /*n, m, ÷*/. / ም ን ኝ/The vocal tract is closed but the velum opens a route to the nasal cavity. The generated voiced sound is affected by both vocal and nasal tract.

4. Laterals: /l/ /ር ል/. The top of the tongue closes the vocal tract leaving a side-route for the air flow.

		Labials		*Alveolar*		*Palatals*		*Velars*		*Labio-Velar*		*Glottals*	
Stops	Voiceless	p	ፕ	t	ት			k	ክ	kx	ኩ	ax	ዕ
	Voiced	b	ብ	d	ድ			g	ግ	gx	ጉ		
	Glottalized	px	ጵ	tx	ጥ			q	ቅ	qx	ቋ		
Fricatives	Voiceless	f	ፍ	s	ስ	sx	ሽ					h	ሀ
	Voiced	v	ቭ	z	ዝ	zx	ዥ						
	Glottalized			xx	ጽ							hx	ኍ
Africatives	Voiceless					c	ች						
	Voiced					j	ጅ						
	Glottalized					cx	ቈ						
Nasals	Voiced	m	ም	n	ን	nx	ኝ						
Liquids	Voiced			l	ር								
				r	ል								
Glides		w	ው			y	ይ						

Table 3.1: Amharic Consonants (sebsebi Z.et.al 2004)

When synthesizing consonants, better results may be achieved by synthesizing these six consonant groups with separate methods because of different acoustic characteristics they have.

3.4. Amharic writing system

The Amharic script is an abugida, and the grapheme of the Amharic writing system is called Fidel (Ethnology, 2016). Each character represents a consonant and vowel sequence, but the basic shape of each character is determined by the consonant, which is modified for the vowel. Some consonant phonemes are written by more than one series of characters: / /, /s /, /sʔ’ /, and /h / (the last one has four distinct letter forms). This is because these fidel originally represented distinct sounds, but phonological changes merged them (Getahun, 2001).The citation form for each series is the consonant+ ä form, i.e. the first column of the fidel. The Amharic script is included in Unicode, and glyphs are included in fonts available with major operating systems. , , , , present writing system of Amharic is taken from Ge‟ez. Ge‟ez in turn took its script from the South Arabian mainly attested in inscriptions in the Sabine dialect (Baye, 1997). Amharic did not discriminate in adopting the Ge‟ez fidel; it took all of the symbols and added some new ones that represent sounds not found in Ge‟ez. These added alphabetic characters are ቸ ጨ ጀ ኘ ቨ, ሸ, ኸ, and ዠ. Currently, the orthographic representation of the language is organized

into seven orders. The list of all Amharic letters is depicted in Appendix A. Amharic language‟s writing system contains 34 base characters each of which occurs in a basic form and six other forms known as orders . The seven orders represent syllable combinations consisting of a consonant following vowel. Out of the seven derivatives six of them are CV (Consonant Vowel) combinations while the sixth is the consonant itself *(Baye, 1997).*

Other symbols representing lateralization, numerals, and punctuation marks are also available. Therefore, having these orthographic variations for each of the 33 core letters, totally the language has more than 230 orthographic symbols; in case of Amharic character ʋ, it is represented by “ha” instead of using “he” unlike other Amharic character sets/orthographies since the sound is the same as its 4th order.

Chapter 4

Hidden Markov Model (HMM) based speech synthesis

4.1. Introduction

As described in chapter one the goal of TTS system is to synthesize speech with natural human voice characteristics and, furthermore, with various speaker individualities and emotions (e.g., anger, sadness, joy). The increasing availability of large speech databases makes it possible to construct TTS systems, which are referred to as data-driven or corpus-based approach, by applying statistical learning algorithms. These systems can be automatically trained, to generate natural and high quality synthetic speech. For constructing such a system, the use of Hidden Markov Models (HMMs) has arisen largely. HMMs have successfully been applied to modeling the sequence of speech spectra in speech recognition systems, and the performance of HMM-based speech recognition systems have been improved by techniques which utilize the flexibility of HMMs: context-dependent modeling, dynamic feature parameters, mixtures of Gaussian densities, tying mechanism, speaker and environment adaptation techniques (Toda et al, 2007). *Tokuda et al (1994)* categorized HMM-based approaches to speech synthesis as follows:

1. Transcription and segmentation of speech database
2. Construction of inventory of speech segments.
3. Run-time selection of multiple instances of speech segments.
4. Speech synthesis from HMMs themselves.

In approaches 1–3, by using a waveform concatenation algorithm, e.g., PSOLA algorithm, a high quality synthetic speech could be produced. However, to obtain various voice characteristics, large amounts of speech data are necessary, and it is difficult to collect, segment, and store the speech data. On the other hand, in approach 4, voice characteristics of synthetic speech can be changed by transforming HMM parameters appropriately (*Tokuda et al 1994).* From this point of view, *Tokuda et al (1994)* proposed parameter generation algorithms for HMM-based speech synthesis.

Figure 4.1 is an overview of HMM-based speech synthesis system. It consists of training and synthesis parts. The training part is similar to that used in speech recognition systems. The main difference is that both spectrum (Mel-cepstral coefficients and their dynamic features) and

excitation (logarithmic fundamental frequencies (log F0) and its dynamic features) parameters are extracted from a speech database by Mel-cepstral analysis and Mel-cepstral coefficients respectively and modeled by context-dependent HMMs (phonetic, linguistic, and prosodic contexts are taken into account). To model variable dimensional parameter sequence such as log F0 with unvoiced regions properly multi-space probability distributions (MSD) are used. Each HMM has state duration probability density functions (PDF) to capture the temporal structure of speech. As a result, the system models spectrum, excitation, and durations in a unified HMM framework *(Zen et al. 2009).*

The synthesis part does the inverse operation of speech recognition. 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 PDF. 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 filter (Zen, et, al.2009).

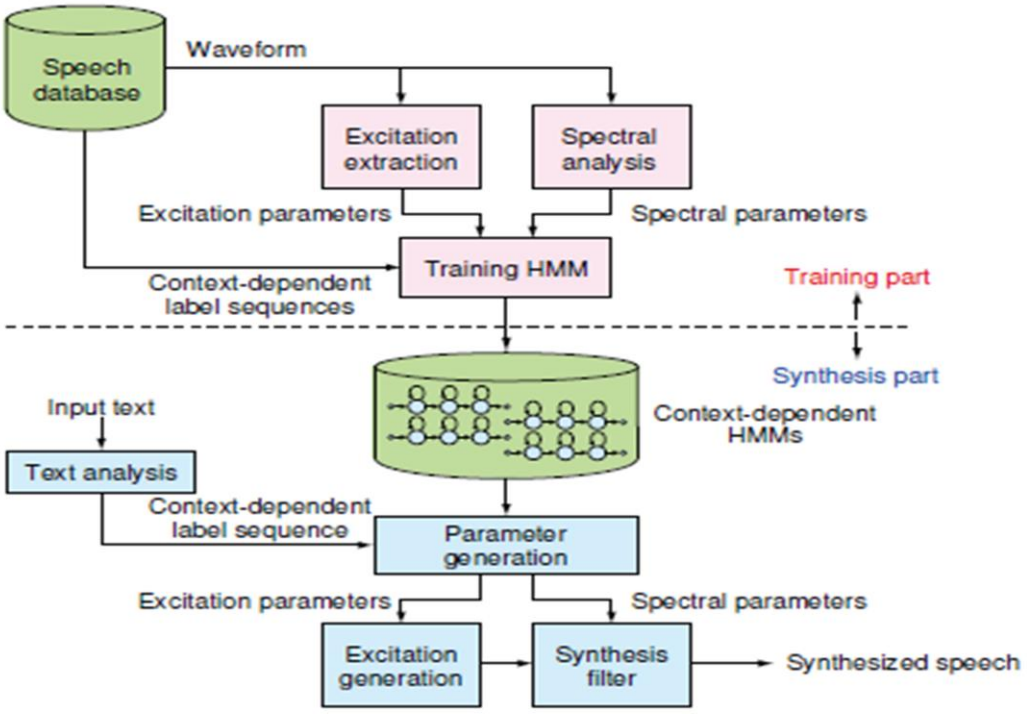


Figure 4.1: HMM-based speech synthesis system (Heiga et al, 2007)

The rest of this chapter is organized as follows: Section 4.2 describe HTS as source-filter model. Section 4.3 summarizes the speech analysis technique. Section 4.4 introduces Hidden Markov Model in general. Section 4.5 describes Speech parameter modeling technique based on HMM. Section 4.6 describes the HMM-Based speaker adaptive training (average voice model) for reducing the influence of speaker- and/or gender-dependent characteristics of spectral, F0 and phone duration. Section 4.8 summarizes the speech synthesis technique and Section 4.9 describes relevant details of HTS version 2.0 system and new features of the system.

4.2. Source-filter model

HMM based Speech synthesis, viewed as a vocoder. The input stage of the vocoder has become "training" and is performed just once for the entire speech corpus (training data). The output stage of the vocoder has become "synthesis", which is performed once for each novel sentence to be synthesized (Zen et al., 2007).To treat a speech waveform mathematically, a discrete-time model is generally used to represent sampled speech signals, as shown in Fig. 4.2. The transfer function H(z) models the structure of vocal tract. The excitation source is chosen by a switch which controls voiced/unvoiced characteristics of speech. The excitation signal is modeled as either a quasi-periodic train of pulses for voiced speech, or a random noise sequence for unvoiced sounds.

To produce speech signals x(n), the parameters of the model must change with time. For many speech sounds, it is reasonable to assume that the general properties of the vocal tract and excitation remain fixed for periods of 5–10 m sec. Under such an assumption, the excitation e(n) is filtered by a slowly time-varying linear system H(z) to generate speech signals x(n).

The speech x(n) can be computed from the excitation e(n) and the impulse response h(n) of the vocal tract using the convolution sum expression

$$x(n) = h(n) * e(n) \quad \text{(4.1)}$$

where the symbol $*$ stands for discrete convolution. Figure 4.2 shows the block diagram of source-filter model.

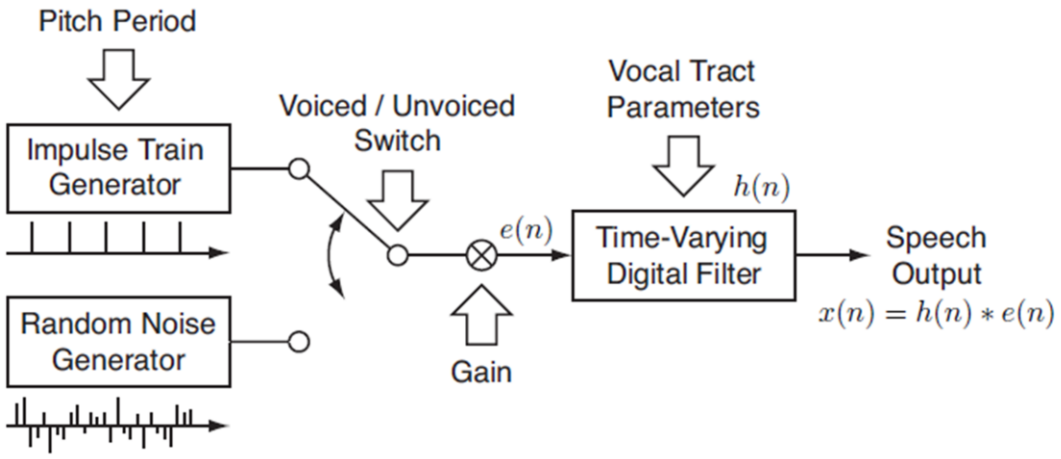


Figure 4-2Source filter Model (Tamura, M., et al, 2001)

4.3. Speech analysis

Speech analysis is the process of converting a speech signal to an alternative representation that in some way better represents important features or characteristics of speech signal that are discriminative and computationally efficient to differentiate words such as type of speech segment (voiced-unvoiced), the value of pitch period, energy and gain values of speech segment and the output of this process are vectors coefficients (Taylor, 2007).

All speech signals in the real world are continuous signals which describe the pattern of air pressure variation over time. These signals can be recorded with a variety of analogue means, but for computer analysis, we require our signals to be digitized such that the continuous signal is converted to a discrete signal. We need to perform analysis because waveforms do not usually directly give us the type of information we are interested in.

Speech analysis in general concerned with three main problems (Taylor, 2007):

- Eliminate phase; the magnitude DFT does this.
- Separate source and filter; this can be done with Capstral analysis or linear prediction
- Transform the representation into a space which has more desirable properties; log magnitude spectra follow the ear's dynamic range; Mel-scaled Capstral scales according to the frequency sensitivity to the ear. Log area ratios are amenable to simple interpolation and line spectral frequencies show the formant patterns robustly.

4.3.1. Discrete Fourier transform (DFT)

The first essential process in speech analysis is windowing, which divide the continuously varying signal into a sequence of stationary smaller signals called frames. Windowing affects the signal and so a perfect representation of the original waveform is not possible. Then the magnitude spectrum is find from a speech signal ,this should be in a discrete form that is easy to calculate and store in a computer, so the most common method is the discrete Fourier transform (DFT) as our principle algorithm and the spectral representation is achieved by applying a discrete Fourier transform (DFT) to the frame of speech. Successive application of DFT can be used to generate a spectrogram (Tokuda et al, 1994).

The second important problem of speech analysis is source-filter separation. In general, we wish to do this because the two components of the speech signal have quite different and independent linguistic functions. The source controls the pitch, which is the acoustic correlate of intonation, while the filter controls the spectral envelope and formant positions, which determine which phones are being produced. There are three popular techniques for performing source-filter separation (Taylor, 2007).

4.3.2. Spectrum Analysis

The spectrum is the inverse Fourier transform of the log magnitude spectrum. It separates variation in frequency across the range and so implicitly separates source and filter components in the spectrum (Acero 1999; Paul 2007).

Spectrum definition

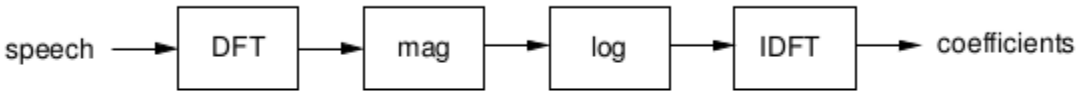
The spectrum is (most commonly) defined as the inverse DFT of the log magnitude of the DFT of a signal:

$$c[n] = \mathcal{F}^{-1}\{\ log|\ \mathcal{F}\ \{\ x[n]\ \}\ |\ \} \qquad \text{..... (4.2)}$$

Where F is the DFT and F^{-1} is the inverse DFT. For a windowed frame of speech y(n] the spectrum is therefore:

$$c[n] = \sum_{n=0}^{N-1} log\Big(\ \Big|\ \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}\ \Big|\ \Big)e^{j\frac{2\pi}{N}kn} \qquad \text{..... (4.3)}$$

The following figure shows the process steps of calculating the DFT, log, and inverse DFT on a single frame of speech. We will now look at how this operation performs source/filter separation.



(a) Steps involved in standard cepstral analysis

Figure 4.3: (b) Steps involved in complex cepstral analysis (Taylor, 2007)

The spectrum calculation differs in two important respects. First we are only using the magnitude of the spectrum - in effect we are throwing away the phase information. The inverse DFT of a magnitude spectrum is already very different from the inverse DFT of a normal (complex) spectrum. The log operation scales the harmonics which emphasizes the "periodicity" of the harmonics. It also ensures that the spectrum is the sum of the source and filter components and not their convolution. The spectrum is useful as it splits the signal into an envelope, given by the first few coefficients, and a source, given by the spike. Subsequent analysis usually throws away one of these parts: if we are interested in the vocal tract we use only the low coefficients, if we are interested in the pitch and behavior of the glottis we study the peak. We can demonstrate this by calculating a spectrum in which the spike is eliminated - this is done by simply setting a cutoff point K above which all the Capstral coefficients are set to zero. From this, we can create a log magnitude spectrum by applying a DFT on the modified spectrum.

4.3.3. Mel frequency capstral coefficients (MFCC)

The use of Mel frequency capstral coefficients can be considered as one of the standard method for feature extraction. The use of about 25 MFCC coefficients is common in TTS, although 10-12 coefficients are often considered to be sufficient for coding speech *(Tokuda et al ,1994; Zen et al, 2007).* According to (*Tamura et al, 2001)* a reasonably accurate spectral envelope can be generated from about 30 coefficients, but the number chosen really depends on the degree of precision required in the spectrum. For many cases where the spectral envelope is required, it is

advantageous to keep the low part of the cepstrum as it is, and not convert it back to a frequency domain spectrum. The most notable downside of using MFCC is its sensitivity to noise due to its dependence on the spectral form. Methods that utilize information in the periodicity of speech signals could be used to overcome this problem, although speech also contains aperiodic content *(Taylor, 2007).*

The low Capstral coefficients form a very compact representation of the envelope, and have the highly desirable statistical modeling property of being independent so that only their means and variances, and not their covariances, need be stored to provide an accurate statistical distribution. It is for this reason that Capstral are the feature representation of choice in most speech recognition systems. The higher part of the spectrum contains the pitch information, and is often used as the basis for pitch detection algorithms (*Tamura et al, 2001).*

The non-linear frequency scale used an approximation to the Mel-frequency scale which is approximately linear for frequencies below 1 kHz and logarithmic for frequencies above 1 kHz. This is motivated by the fact that the human auditory system becomes less frequency-selective as frequency increases above 1 kHz. The MFCC features correspond to the spectrum of the log filter-bank energies. To calculate them, the log energy is first computed from the filter-bank outputs as where X(n] is the DFT of the t th input speech frame, HM(n] is the frequency response of m th filter in the filter-bank, N is the window size of the transform and M is the total number of filters. Then, the discrete cosine transform (DCT) of the log energies is computed. Since the human auditory system is sensitive to time evolution of the spectral content of the signal, an effort is often made to include .In order to capture the changes in the coefficients over time, first and second difference coefficients are computed as respectively (*Yamagishi, J ,et al., 2009).* These dynamic coefficients are then concatenated with the static coefficients according to making up the final output of feature analysis representing the t th speech frame *(Yoshimura et al, 1999).*

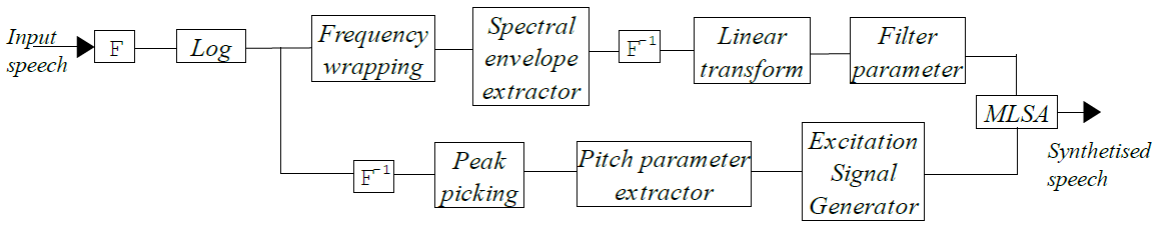


Figure 4-4 Mel Frequency Capstral Coefficients Analysis and Synthesis (Cecilia, 2008)

4.4. The Hidden Markov Model

The hidden Markov model (HMM) is one of statistical time series models widely used in various fields. Especially, text-to-speech synthesis systems to generate speech from input text information has also made substantial progress by using the excellent framework of the HMM. In this section, we briefly describe the basic theory of the HMM.

Definition

A hidden Markov model (HMM) is a finite state machine which generates a sequence of discrete time observations. At each time unit, the HMM changes states at Markov process in accordance with a state transition probability, and then generates observational data o in accordance with an output probability distribution of the current state.

An N-state HMM is defined by the state transition probability A= $\{a_{ij}\}_{i,j=1}^{N}$,the output probability distribution B = {b i (o)} i=1 , and initial state probability Π = {π i } N i=1 . For notational simplicity, the model parameters of the HMM as follow:

$$\lambda = (A, B, \Pi) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

Figure 4.6. shows examples of typical Triphone HMM structure. 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. The sequence of frames of speech, known as observations and denoted as O =< o 1 , o 2,..o T >. The frames are processed so as to removed phase and source information. Hence each observation o j is a vector of continuous values. For each phone three types of a probabilistic model which tells us the probability of observing a particular acoustic input see figure 4.6 and 4.7.

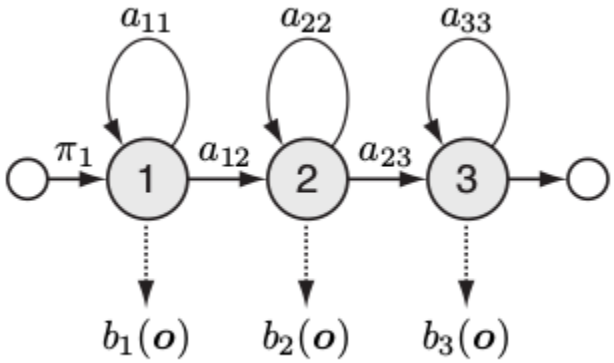


Figure 4-6 Triphone HMM structure (Yoshimura ,2002)

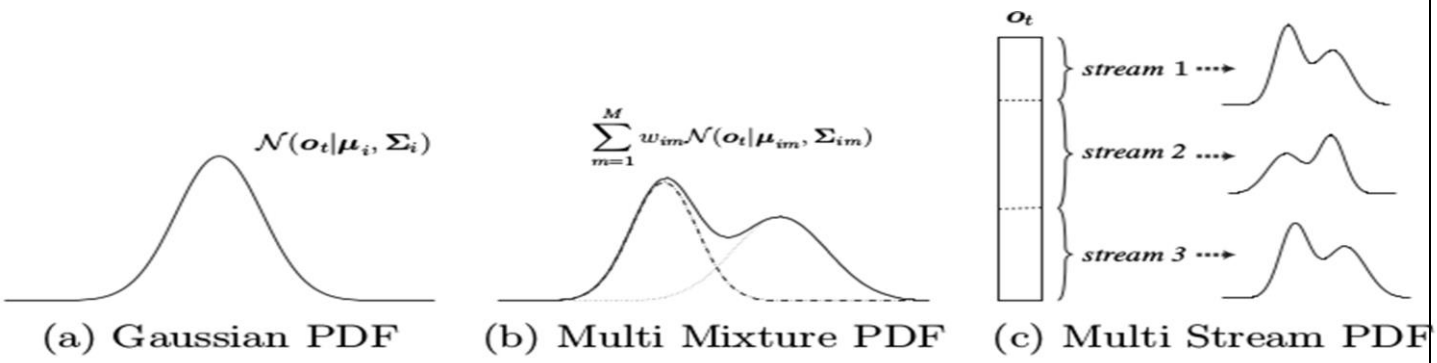


Figure 4-7 Output Distribution (Yoshimura ,2002)

The output probability distribution b i (o) of the observational data o of state i can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM) for the continuous observational data, the output probability distribution is usually modeled by a mixture of multivariate Gaussian distributions as follows:

$$b(o_t) = \sum_{m=1}^{M} c_m \mathcal{N}(o_t; \mu_m, \Sigma_m) \quad \text{……...……. (4.5)}$$

where M is the number of mixture components for the distribution, and w im , μ im and Σ im are a weight, a L-dimensional mean vector, and a L × L covariance matrix of mixture component m of state i, respectively. A Gaussian distribution N (o; μ ,Σ) of each component is defined by:

$$\mathcal{N}(o;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^N|\Sigma|}} e^{-\frac{1}{2}(o-\mu)'\Sigma^{-1}(o-\mu)} \quad \text{…….…….. (4.6)}$$

Where N is the dimensionality, μ is the vector of means and Σ is the covariance matrix where L is the dimensionality of the observation data o.

With this, we can therefore build a system in which we have a model for every phone, each described by its own multivariate Gaussian. For an unknown utterance, if we know the phone boundaries, we can therefore test each phone model in turn and find which model gives the highest probability to the observed frames of speech, and from this find the sequence of phones that are most likely to have given rise to the observations in the utterance in question. We on the

accuracy of these models can be improve in a number of ways. First, we note that the true density of a phone model is in fact rarely Gaussian. Rather than use other types of distribution, we adopt a general solution whereby we use a mixture of Gaussian s, shown in Figure 10.C and given by:

$$b_i(\boldsymbol{o}) = \prod_{s=1}^{S} b_{is}(\boldsymbol{o}_s) \qquad \text{..... (4.7)}$$

$$= \prod_{s=1}^{S} \left\{ \sum_{m=1}^{M_s} w_{ism} \mathcal{N}(\boldsymbol{o}_s; \boldsymbol{\mu}_{ism}, \boldsymbol{\Sigma}_{ism}) \right\} \qquad \text{..... (4.8)}$$

In this, we have M Gaussian, each with a mean vector μ m and covariance matrix Σ m . The parameters c m are known as the mixture weights and are used to determine the relative importance of each Gaussian. As with any pdf, the area under the curve should sum to 1. Here we can model a pdf of arbitrary complexity by summing weighted Gaussian, there we could build a periodic signal of arbitrary complexity by summing weighted sinusoids.

The three core problems associated with HMMs are:

- **Efficient evaluation of the marginal over all states**: how to compute the probability density of an observation sequence given an HMM, the so-called likelihood
- **How likely the data is to be generated by that model**. Method: the forward-backward algorithm.
- **Model parameter estimation**: how to estimate the parameters that define an HMM given a training dataset. Method: the expectation maximization algorithm.
- **Computation of the optimal state sequence:** given the observation sequence how to find the most probable state sequence. Method: Viterbi algorithm. In the next section we present how these problems are tackled during the train stage of HMMs.

4.4.1. Probability calculation

For calculation of P (O|λ), which is the probability of the observation sequence O =(o1,o2 ,·,o T) given the model λ, forward-backward algorithm is generally used. If we calculate P (O|λ) directly without this algorithm, it requires on the order of 2T N 2 calculation. On the other hand, calculation using forward-backward algorithm requires on the order of N 2 T calculations, and it

is computationally feasible. In the following part, forward-backward algorithm is described.

The forward algorithm

Consider the forward variable α t (i) defined as α t (i) = P (o 1 , o 2 , · · · , o t , q t = i|λ) that is, the probability of the partial observation sequence from 1 to t and state I at time t, given the model λ. We can solve for α t (i) inductively, as follows:

1. Initialization $\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

The backward algorithm

In the same way as forward algorithm, consider the backward variable β t (i) defined as

β t (i) = P (o t + 1, o t + 2, · · · , o T |q t = i, λ)

that is, the probability of the partial observation sequence from t to T , given state i at time t and the model λ. We can solve for β t (i) inductively, as follows:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N.$$

2. Induction

$$\beta_t(i) = \sum_{i=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad \begin{matrix} t = T-1, T-2, \cdots, 1 \\ 1 \leq i \leq N \end{matrix}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \beta_1(i).$$

The forward-backward probability calculation is based on the trellis structure shown in Fig. 3.2. In this figure, the x-axis and y-axis represent observation sequence and states of Markov model, respectively. On the trellis, all the possible state sequence will re-merge into these N nodes no matter how long the observation sequence. In the case of the forward algorithm, at times t = 1,

we need to calculate values of α 1 (i), $1 \leq i \leq N$. At times t = 2, 3, · · · , T , we need only calculate values of α t (j), $1 \leq j \leq N$, where each calculation involves only the N previous values of α t−1 (I) because each of the N grid points can be reached from only the N grid points at the previous time slot.

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N,$$
$$\psi_1(i) = 0, \qquad 1 \leq i \leq N.$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_t(i) a_{ij}\right] \boldsymbol{o}_t, \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array}$$

$$\psi_t(j) = \operatorname*{argmax}_i \left[\delta_t(i) a_{ij}\right], \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T. \end{array}$$

3. Termination

$$P(\boldsymbol{O}, \boldsymbol{q}^* | \lambda) = \max_i \left[\delta_T(i)\right],$$
$$q_T^* = \operatorname*{argmax}_i [\delta_T(i)].$$

4. Path backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*).$$

As the result, the forward-backward algorithm can reduce order of probability calculation.

All of the information needed to perform HMM parameter re-estimation using the Baum-welch algorithm is now in place. The steps in this algorithm may be summarized as follows

1. For every parameter vector/matrix requiring re-estimation, allocate storage for the numerator and denominator summations of the form. These storage locations are referred to as accumulators.
2. Calculate the forward and backward probabilities for all states j and times t.

3. For each state j and time t, use the probability L j (t) and the current observation vector o t to update the accumulators for that state.

4. Use the final accumulator values to calculate new parameter values.

5. If the value of P = P(O|M) for this iteration is not higher than the value at the previous iteration then stop, otherwise repeat the above steps using the new re-estimated parameter values. All of the above assumes that the parameters for a HMM are re-estimated from a single observation sequence, that is a single example of the spoken word.

4.5. Speech parameter modeling based on HMM

4.5.1. Acoustic modeling

Acoustic modeling is used to represent distinct sound that makes up a spoken word used in the language model. Each distinct sound verses phone. Acoustic modeling process takes four main inputs; pronunciation dictionary (can be canonical or phoneme label), training text corpus, feature vectors of the training speech corpus and language models (can be either letter sequences or phoneme sequence). The acoustic models are statistical models which capture the correspondence between a short sequence of acoustic vectors and letters or phonemes. It is created using audio recordings of speech and their text scripts and compiling them into a statistical representation of sounds which makeup words. The HMM-Based acoustic modeling system is also carried out in three modeling techniques, namely acoustic modeling without tied state, with tied state and tied state with various Gaussian mixture values. multi-space probability distribution (MSD with various Gaussian mixture These modeling are considered to get the most accurate results so as to increase the performance of speech Synthesizer and used in HMM TTS systems. multi-space probability distribution (MSD), is the same as the discrete distribution and the continuous distribution. The continuous distribution is represented by a G-mixture probability density function. Thus multi-space probability distribution is more general than either discrete or continuous distributions.

4.5.2. Acoustic representations and covariance

A final point regarding the probabilistic modeling of observations concerns issues with the covariance matrix. If we use say 13 acoustic coefficients (often a standard number made from 12 cepstral coefficients and one energy coefficient), and then use delta and acceleration coefficients, we have a total of 39

coefficients in each observation frame and so the mean vector has 39 values. The covariance matrix however has $39^2 = 1521$ values. It is often difficult to find enough data to accurately determine each of these values, and so a common solution is to ignore the covariance between coefficients, and simply model the variance of each coefficient alone. This is termed using a diagonal covariance, as only the diagonal values in the covariance matrix are calculated; the other values are set to zero. In general, using such a covariance matrix is not advised as the correlations (covariances) between coefficients often contain useful information. However, if we can show that the coefficients in the data vary more or less independently of one another, no modeling power is lost in this assumption. It is partly for this reason that Mel-frequency Capstral coefficients (MFCC) are used as the representation of choice in HMM based TTS systems (in addition, they are deemed to have good discrimination properties and are somewhat insensitive to differences between speakers). It is important to note however HMMs themselves are neutral to the type of observation used, and in principle we can use any of the signal processing derived representations.

4.5.3. States and transitions

As we noted above, the pattern of frames within a phone is not static. In addition to modeling the rate of change, it is also normal to split each phone model into a number of states, each of which represents a different part of the phone. In principle any number of states is allowable, but it is quite common to use three, which can informally be thought of as modeling the beginning, middle and end of each phone. Transition probabilities which give us the probability of moving from one state to the next (this "moving" process is most easily visualized if we think of the models as the generators of the acoustic data). In general, we can move from any state to any other state, and so for a model with P states, the transition probabilities can be stored in a $P \times P$ matrix. This matrix stores a set of set of discrete probabilities a i j which give the probability of moving from state i to state j.

4.5.6. F0 parameter modeling

The models are trained with parameters extracted from natural speech, to maximize the likelihood of the training data. The source can be represented by the fundamental frequency and the aperiodicity band energies and the spectral envelope by Mel generalized Capstral coefficients (Tokuda et al., 1994). The F0 pattern is composed of continuous values in the "voiced" region and a discrete symbol in the "unvoiced" region. Therefore, it is difficult to apply the discrete or

continuous HMMs to F0 pattern modeling. Several methods have been investigated for handling the unvoiced region:

A possible approach is just too randomly sample from each HMM. So starting in the first state of the first phone, simply use a random number generator to generate a value which it use with the state's Gaussian to generate an observation. It generates another random number and use this to decide which state will use next. This process is repeated until the entire utterance generated. This approach is valid in a statistical sense in that over a sufficiently large number of syntheses, the utterances generated will have the same statistical properties as the models. replacing each "unvoiced" symbol by a random vector generated from a probability density function (pdf) with a large variance and then modeling the random vectors explicitly in the continuous HMMs,This approach does not however produce natural speech; the main reason being that such an approach causes the spectra changes rapidly and randomly from one frame to the next. Real speech in contrast evolves with some level of continuity (Taylor, 2007).

A second approach is to use a maximum likelihood solution, in which instead of randomly sampling from the model in accordance with its mean and variance, we just generate the most likely sequence of observations from the sequence of models. (Unlike the first approach, retraining on this data will not give us the same models as the values for all states will be exactly the same and hence all the variances will be zero). It should be obvious that in all cases each state will generate its mean observation. This avoids the problem of the previous approach in which the observations were "jumping around" from one frame to the next, but now has the problem that the spectrum clearly "jumps" at each state boundary. Furthermore the spectra are the same during each state, meaning that, in one dimension, the generated speech is a flat line followed by a discontinuity, followed by a different flat line. This again does not look or sound like natural speech. In effect ignoring all variance information - if retrained on speech generated by this model we would always see the same observation for each state and hence would calculate the same mean but calculate a zero variance in all cases. Modeling the "unvoiced" symbols explicitly in the continuous HMMs by replacing "unvoiced" symbol by 0 and adding an extra pdf for the "unvoiced" symbol to each mixture.

The synthesis module of the vocoder requires parameters that describe both the spectral envelope and the excitation signal. To drive the generation of the excitation signal, F0 values need to be modeled. Unlike the spectral and aperiodicity parameters, F0 is not strictly continuous. For

voiced segments, F0 is continuously defined but for unvoiced segments it is undefined, however, this does not mean that it takes the value of zero. One of the ways of handling this is to consider F0 as a multi space variable (Tokuda et al., 2002), where one space assumes continuous values and the other space a discrete distribution. For each state, there is a label that indicates which space F0 is and which distribution is attributed to it. In order to maintain synchronization between the different parameters (spectral, aperiodicity and F0), the observation vector adopted in HTS is composed of multiple separate streams in a multi-stream HMM (Tokuda et al., 2002). Each stream contains static and dynamic representations of this data. Each stream refers here to sections of the observation vector that are considered to be statistically independent of each other. Multi-stream training keeps the synchronization of spectral and excitation models while still allowing them to be separately tied, as we will soon discuss.

4.5.7. Duration modeling

In a standard HMM, transition probabilities determine the durational characteristics of the model. If we take a standard topology where a state can loop back to itself (with say probability 0.8) or move to the next state (with probability 0.2), .(Yoshimura ,(2002) shows that the distribution of state occupancy has an exponential form, where it is most common to generate a single observation, with the next most common being 2 observations, then 3 and so on. In reality, (Yoshimura ,2002) shows that this does not follow the pattern of observed phone durations which are much more accurately modeled by a Gaussian, or a skewed Gaussian-like distribution which has low probability for both very short and very long sequences, and higher probability for sequences at or near the mean. Without explicit duration modeling, the state duration of an HMM would be given by the distribution of the transition probabilities which in turn give an exponential decaying distribution. As this is not a good model to generate natural sounding phone durations, explicit duration modeling in the form of the semi-Markov structure was proposed (Tokuda et al., 2002). In this model, the transition probabilities are replaced by an explicit Gaussian duration model. It is now known that modeling the duration accurately is known to be important for speech synthesis.

4.5.8. Context-Independent HMMs (CI-HMM)

HMM training is carried out first for each mono-phone in context-independent training, creating context-independent HMMs (CI-HMM). The CI-HMM are then tied together using a stream-

dependent tree-based state clustering: a different decision tree will be built for spectral, excitation and duration parameters. The streams of spectral, excitation and duration parameters are clustered independently with the assumption that their dependency on the linguistic context will be different: F0 and duration are more affected by supra-segmental linguistic specification while spectral parameters are affected by localized linguistic characteristics like the phone (*Yoshimura, 2002*). Each leaf of the decision tree refers to a context-dependent state (CD-HMM) . The linguistic specification determined by the questions leading to the leaf nodes indexes the !!br0ken!! The questions associated with the decision trees in practice define regions in the linguistic space (the multidimensional space covered by all possible linguistic specifications) so an unseen specification will be associated with the model of the region it comes from. In other words, any context will reach one of the leaf nodes, from the root node then selecting the next node depending on the answer about the current context. In the clustering technique, the size of the decision tree is automatically controlled based on the minimum description length criterion (Tokuda et al., 2002).

4.5.9. Context-Dependent modeling

In continuous speech, parameter sequences of particular speech unit (e.g., phoneme) can vary according to phonetic context. To manage the variations appropriately, context dependent models, such as Tri-phone models, are often employed (*Yoshimura, 2002*). One common HMM topology is to use three states, each its own observation probability. Each state has is linked to the next state and back to itself again. This last transition is known as the self transition probability and is basically the probability that will generate the next observation from the state are already in (*Yoshimura, 2002*). A phone's state transition probabilities govern the durational characteristics of the phone; if the self transition probabilities are high, it is more likely that more observations will be generated by that phone which means the overall phone length will be longer (Taylor 2007)**.**

Triphone models, which are not models made of three phones, but single phone models in the context or the preceding and following phone. Hence for a phoneme set of N phones, there will be just less than N 3 triphones (some combinations never occur). Unfortunately, we have insufficient examples to train a model (low occupancy), or no examples at all (zero occupancy). The solution to this is to cluster the data; in effect borrow parameters from well trained models for

use in the ones that suffer from data sparsity. There are a number of ways we could do this for the case of models with low occupancy, in that we can just find examples in acoustic space which are close to the known examples. The situation is more difficult for those with zero occupancy, as we don't have any acoustic examples at all. Tokuda et al., (2002) solve this by making use of some of the common properties of phones, and the most common way of doing this is to use the phones' distinctive features. In doing so, positing that phones which share the same place of articulation may have more similar acoustic realizations than ones which don't. The most common way of performing this feature based clustering is to use a decision tree Tokuda et al., (2002).

4.5.10. Decision-tree-based context clustering

In the HMM-based speech synthesis system, Yamagishi et al (2006) use more complicated speech units considering prosodic and linguistic context such as Mora, accentual phrase, part of speech, breath group, and sentence information to model suprasegmental features in prosodic feature appropriately. However, it is impossible to prepare training data which cover all possible context dependent units, and there is great variation in the frequency of appearance of each context dependent unit. To alleviate these problems, a number of techniques are proposed to cluster HMM states and share model parameters among states in each cluster (Young,1994). Here, HTS used a decision-tree-based state tying algorithm. This algorithm is often referred to as decision-tree-based context clustering algorithm. the method of the decision tree construction is using the minimum description length (MDL) criterion (Tokuda et al., 2002).

The key idea this technique is that while it use the features to suggest commonality between triphones, it use the actual data to determine how close any particular feature combination actually is. In other words, the features serve as constraints or structure on the clustering, but don't determine what should be clustered with what. The decision tree operates in a top down manner. It operates on binary features, so as a preprocessing step it convert the original features into binary ones (so for instance the feature consonant type which can take values stop, fricative etc gets converted into a new set of features which encode the same information but in a binary valued system: often this is done by simply using simple question features, such as "is this a stop?", "is this a fricative?"). Initially, all the data points for the state of one phone are grouped together in a

single cluster. This cluster is characterized by a measure of impurity; choices of which include variance, log likelihood or entropy (Young,1994).

The process of growing a tree is as follows Yamagishi (2006):

1. Create an initial cluster containing all the data points
2. For each feature: form two new clusters based on the value of the feature measure the combined variance of the two new clusters:
3. Find the feature which gives the biggest reduction in variance
4. Delete this feature from the list of features to be examined.
5. Form two new clusters based on this feature.
6. Repeat steps 1 to 4 on each new cluster until the stopping criteria have been met.

Stopping criteria usually involve specifying a minimum decrease in impurity, and that clusters should have a minimum occupancy (say 10 data points). The process of examining and splitting clusters is shown in figure 4.8. An important point about the decision tree grown in this way is that it provides a cluster for every feature combination, not just those encountered in the training data. So see this, consider the tree in one branch of this has the feature set.

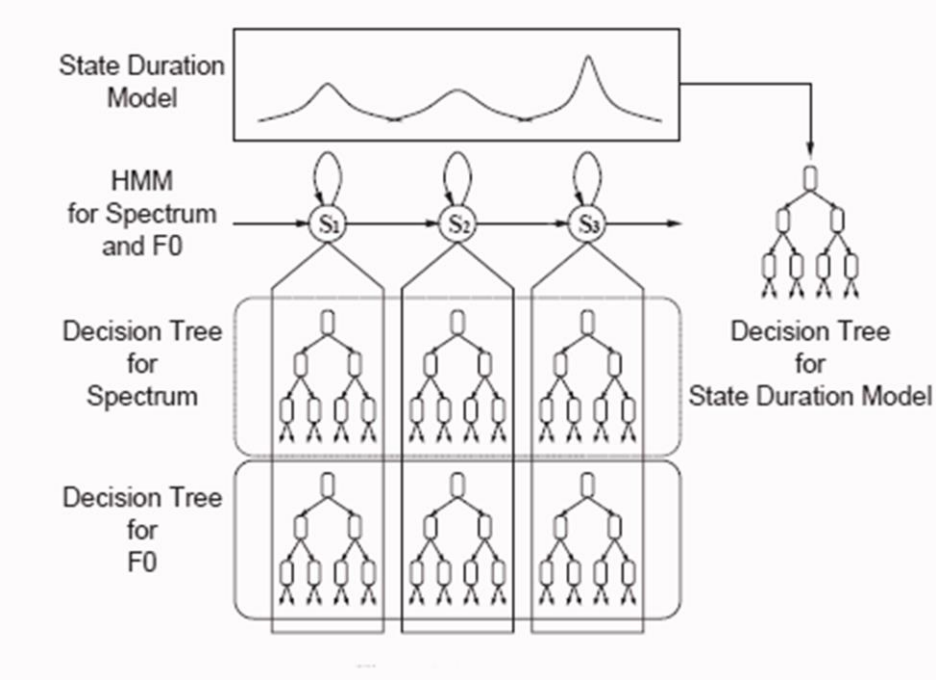


Figure 4-8 Decision Tree (cf :Tokunda 2009)

4.6. Speaker adaptation using average voice model

The statistical framework enables the use of model adaptation techniques that can adjust trained models in such a way that they better describe a new dataset. Adaptation methods for HMMs in speech technology were first proposed in the context of speech recognition in order to adapt a system for a particular speaker, channel condition or language *(Black.,et.al, 2007)* .

For TTS, the adaptation techniques are used, for instance, to create voices for a particular speaker from a small amount of speech data through the use of a model trained with a large amount of data from other speakers – the average voice model (Yamagishi,2006).Speaker adaptation can also be used to create voices with different speaking styles, which could be emotional states like happy and sad or production-related styles like hyper and hypo articulation, clear speech and noise-driven Lombard speech. In this context, it is possible to adapt a model trained with neutral speech data to one of these styles using a small amount of style-matched training data *(Yoshimura et al., 1999; Zen et al., 2007a; Ling et al., 2006; Black, 2006).*

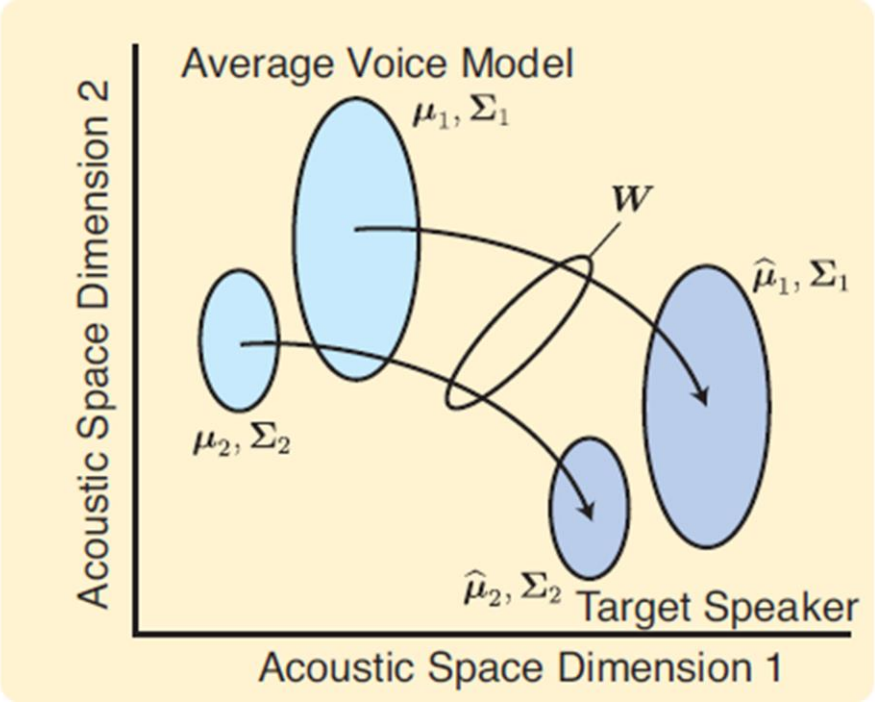


Figure 4.9: Speaker Adaptation Using average voice model (Yamagishi, J. 2006).

The main techniques for adaptation of HMMs are based on linear regression applied to the model parameters: Gaussian means and covariance matrices are adjusted using a linear transform. Maximum likelihood linear regression (MLLR) adaptation updates the mean vectors in order to maximize the likelihood of the data used for adaptation *(Yoshimura et al., 1999; Zen et al., 2007a; Ling et al., 2006; Black, 2006).*The same transform is shared across different states because the limited adaptation data might not cover all models.

In the speaker adaptation, initial model parameters, such as mean vectors of output distributions, are adapted to a target speaker using a small amount of adaptation data uttered by the target speaker. The initial model can be speaker dependent or independent. For the case of speaker dependent initial model, since most of speaker adaptation techniques tend to work insufficiently between two speakers with significant difference in voice characteristics, it is required to select the speaker used for training the initial model appropriately depending on the target speaker.

On the other hand, using speaker independent initial models, speaker adaptation techniques work well for most target speakers, though the performance will be lower than using speaker dependent initial models which matches the target speaker and has sufficient data. Since the synthetic speech generated from the speaker independent model can be considered to have averaged voice characteristics and prosodic features of speakers used for training as the "Speaker Independent" or "average voice model" (see figure 4.9).

In the HMM-based speech synthesis method, spectral and prosodic characteristics of synthetic speech can easily change by transforming HMM parameters appropriately since speech parameters used in the synthesis stage are statistically modeled by using the framework of the HMM (Yamengeshi 2006). In fact, (Tamure et al ,Lagger 1996;Yamengishi 2006) shown that the TTS system can generate synthetic speech which closely resembles an arbitrarily given speaker's voice using a small amount of target speaker's speech data by applying speaker adaptation techniques such as MLLR (Maximum Likelihood Linear Regression) algorithm (Leggetter,1996).In this thesis MLLR algorithm is adopted for speaker adaptation experiment.

4.7. Speech synthesis

4.7.1. Speech synthesis from spectral coefficients

We now turn to techniques used to synthesize speech from Capstral representations and in particular the Mel-frequency Capstral coefficients (MFCC) commonly used in HMM-based TTS. the main reason is that they are a representation that is highly amenable to robust statistical analysis because the coefficients are statistically independent of one another. Most of these operations are invertible, though the fact that we use the magnitude spectrum only and discard the phase means that information has been lost at this stage. The steps involved in MFCC analysis and we see that the situation is somewhat more complicated. First, it is common to perform pr-emphasis so as to remove the inherent tilt in the spectrum. Next we perform the filter bank operation that smooth the spectrum and performs the Mel-scaling. Finally after the spectrum has been created, we perform a "filtering" operation where we discard the higher Capstral coefficients, to leave typically only 12. The filtering operation and the filter bank operation are not invertible because information is lost at these points. A number of techniques have been developed which attempt to reverse these operations and generate speech. Here we follow one MFCC technique described in the following steps:

1. Remove the pr-emphasis and the influence that the Mel-scaling operation has on spectral tilt. This can be performed by creating Capstral vectors for each process, and then simply subtracting these from the MFCC vector.
2. Perform an inverse filtering operation by padding the Mel-scale spectrum with zeros. This then gives us a vector of the correct size.
3. Perform an inverse cosine transform, which gives us a Mel-scaled spectrum. This differs from the analysis Mel-scale spectrum because of the filtering, but in fact the differences in the envelopes of the original and reconstructed spectra have been shown to be minor, particularly with respect to the important formant locations.
4. Partially reverse the filter bank operation. This is not trivial and it is impossible to recover even an approximation of the original spectrum as we threw away all the information about the harmonics in the original filter bank operation. Instead, we attempt to find the spectral envelope, and do this by a process of "up-sampling" the filter bank to a very detailed spectrum, and then sampling this at the required intervals, so as to give a spectrum with 128 points.

5. From the magnitude spectrum, we calculate the power spectrum, and from this calculate a set of autocorrelation coefficients by performing an inverse DFT.

6. Calculate the LP coefficients from the autocorrelation coefficients.

7. Choose a value for F0, use this to generate an impulse train which then excites the LP coefficients. Use a noise source for unvoiced components.

This technique successfully reverses the MFCC coding operation. The main weakness is that because we threw away the harmonic information in the filter bank step, we have to resort to a classical LP style technique of using an impulse to drive the LP filter. A number of techniques have been developed to modify the pitch and timing. These include:

- **PSOLA** which operates in the time domain. It separates the original speech into frames pitch-synchronously and performs modification by overlapping and adding these frames onto a new set of epochs, created to match the synthesis specification.
- **Residual excited linear prediction** performs LP analysis, but uses the whole residual in re-synthesis rather than an impulse. The residual is modified in a manner very similar to that of PSOLA.
- **Sinusoidal models** use a harmonic model and decompose each frame into a set of harmonics of an estimated fundamental frequency. The model parameters are the amplitudes and phases of the harmonics. With these, the value of the fundamental can be changed while keeping the same basic spectral envelope.
- **Harmonic Noise models** are similar to sinusoidal models, except that they have an additional noise component which allows accurate modeling of noisy high frequency portions of voiced speech and all parts of unvoiced speech.
- **MBROLA** is a PSOLA like technique which uses sinusoidal modeling to decompose each frame and from this resynthesizes the database at a constant pitch and phase, thus alleviating many problems in inaccurate epoch detection.
- **MFCC synthesis** is a technique which attempts to synthesis from a representation that we use because of its statistical modeling properties. A completely accurate synthesis from this is not possible, but it is possible to perform fairly accurate vocal tract filter reconstruction. Basic techniques use an impulse/noise excitation method, while more advanced techniques attempt a complex parameterization of the source. The quality of these techniques is considerably higher than classical, impulse excited linear prediction. All these have roughly similar quality, meaning that the choice of which technique to use is mostly made on other criteria, such as speed and

storage. The speech segment will be represented in parameter coefficient values that will be the input for the next phase, Acoustic modeling process. HMM based TTS uses cap-strum analysis technique to extract Speech Feature. A number of improvements have been made to this, with the motivation of generating a more natural source, while still keeping a model systems where the parameters are largely statistically independent. For example in the technique of (Yoshimura et al 2001) a number of excitation parameters are used that allow mixing of noise and impulse, and allow a degree of aperiodicity in the positions of the impulses.

4.7.2. Synthesis from mixed excitation model

The HMM-based TTS system used excitation parameters with either a periodic impulse train or white noise .To overcome this problem; the excitation model should be replaced with more precise one. For low bit rate narrow-band speech coding at 2.4kbps, the mixed excitation linear predictive (MELP) vocoder has been proposed. In order to reduce the synthetic quality and mimic the characteristics of natural human speech, this vocoder has the following capabilities:

- mixed pulse and noise excitation
- periodic or aperiodic pulses
- pulse dispersion filter

The mixed excitation is implemented using a multi-band mixing model figure 14, and can reduce the buzz of synthesized speech. Furthermore, aperiodic pulses and pulse dispersion filter reduce some of the harsh or tonal sound quality of synthesized speech. In order to realize the mixed excitation model in the system F0, band pass voicing strengths and Fourier magnitudes excitation parameters are extracted from speech data.

In band pass voicing analysis, the speech signal is filtered into five frequency bands, with pass band of 0–1000, 1000–2000, 2000–4000, 4000–6000, 6000–8000Hz (Paul ,2007). Note that the TTS system deals with 16kHz sampling speech. The voicing strength in each band is estimated using normalized correlation coefficients around the pitch lag.

The correlation coefficient at delay t is defined by

$$c_t = \frac{\sum_{n=0}^{N-1} s_n s_{n+t}}{\sqrt{\sum_{n=0}^{N-1} s_n s_n \sum_{n=0}^{N-1} s_{n+t} s_{n+t}}},$$

Where s,n and N represent the speech signal at sample n and the size of pitch analysis window, respectively. The Fourier magnitudes of the first ten pitch harmonics are measured from a residual signal obtained by inverse filtering.

A block diagram of the mixed excitation generation and speech synthesis filtering is shown in Fig 4.11. The band pass filters for pulse train and white noise are determined from generated band pass voicing strength. The band pass filter for pulse train is given by the sum of all the band pass filter coefficients for the voiced frequency bands, while the band pass filter for white noise is given by the sum of the band pass filter coefficients for the unvoiced bands. The excitation is generated as the sum of the filtered pulse and noise excitations. The pulse excitation is calculated from Fourier magnitudes using an inverse DFT of one pitch period in length. The pitch used here is adjusted by varying 25% of its position according to the periodic/aperiodic flag decided from the bandpass voicing strength. By the aperiodic pulses, the system mimics the erratic glottal pulses and reduces the tonal noise. The noise excitation is generated by a uniform random number generator. The obtained pulse and noise excitations are filtered and added together.

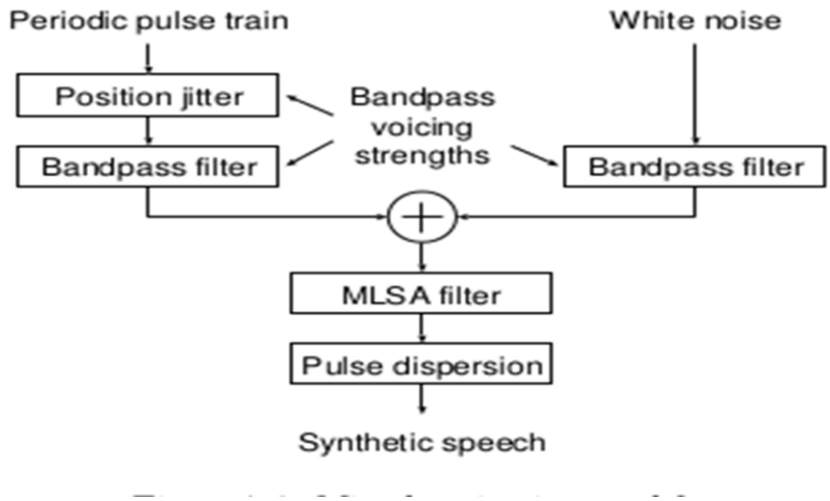


Figure 4.11: Mixed excitation model (Yoshimura, 2002).

By exciting the MLSA filter, synthesized speech is generated from the Mel-cepstral coefficients, directly. Finally, the obtained speech is filtered by a pulse dispersion filter which is a 130-th order FIR filter derived from a spectrally-flattened triangle pulse based on a typical male pitch period. The pulse dispersion filter can reduce some of the harsh quality of the synthesized speech.

4.8. HMM Training and synthesis tools

4.8.1. HMM TOOLKIT (HTS-2.0)

The HMM-based speech synthesis system (HTS) is an extension of HTK. The HMM Toolkit (HTK) is a de-facto standard toolkit for training and manipulating HMMs in speech research. It consists of a set of libraries and tools in C and provides basis of HMM-based speech synthesis research. HTS adds various functionalities for HMM-based speech synthesis. It is in C and released as a patch code to HTK. HTS version 2 used in this thesis for building the state-of-the-art speaker-dependent and speaker-adaptive synthesizers and some voices for the Festival speech synthesis system. Since December 2002, (Zen et al,2009) have publicly released an open source software toolkit named HMM based speech synthesis system (HTS) to provide a research and development platform for speech synthesis community. In this system, context-dependent HMMs are trained from databases of natural speech, and we can generate speech waveforms from the HMMs themselves. This system offers the ability to model different styles without requiring the recording of very large databases. The most attractive part of this system is that its voice characteristics, speaking styles, or emotions can easily be modified by transforming HMM parameters using various techniques such as adaptation , interpolation , eigenvector , or multiple regression (Zen et al,2009). (Acero 1999) notes that HMMs do in fact generate the sort of formant trajectories often observed in natural data. In particular, he notes that when a model is only used to generate a few frames of speech, its generated observations rarely reach their mean value but when the model is used to generate longer sequences, the mean values are in fact reached.

The hts_engine is a set of APIs of a run-time synthesis engine for HMM based speech synthesis. It is also in C and provides various functionalities required to set up and drive the synthesis engine.

Chapter 5

Developing HMM-based Amharic speech synthesizer

5.1. Introduction

This chapter discusses the design of the Amharic Speech Synthesizer along with the implementation principle of the model. The design process is based on the phonetic properties of the language presented in chapter three and statistical parametric approaches discussed in chapter two and HMM based TTS methods discussed in chapter four. Section 5.2 discusses the assumptions and conventions taken into consideration during the design and implementation of Amharic speech corpus. Section 5.3 describes the architecture of the Amharic Speech Synthesizer and the implementation procedures according to the designed model of the synthesizer. Finally, the performance of speaker independent Amharic speech synthesizer is evaluated in terms of percentage of mean open score result. The test results are also presented in both tabular and textual forms.

In Phoneme based Acoustic modeling category the items to be synthesized are usually modeled with hidden Markov models (HMMs), which are concatenated from phone models according to a pronunciation dictionary. This allows the construction of HMMs for Triphone topologies. There are several prerequisites of HMM-based Acoustic modeling:

- ❖ Annotated data to train appropriate statistical models of the phonemes has to be available.
- ❖ A transcription of each occurring word has to be available. The transcriptions are usually obtained from a pronunciation dictionary.
- ❖ An appropriate language model which defines the sequences of words which the synthesizer is able to understand and the probabilities of word sequences is necessary.

Figure 5.1 shows the system architecture of the Amharic speech synthesizer using HMM. As the system architecture shows, the synthesizer has two parts: The training and synthesis part. The training part includes data preparation, language modeling, feature extraction, and building the HMM. Whereas, the synthesis part includes preparing labeled text from the text input, selecting appropriate HMMs, extracting speech parameters from HMMs, and finally generating the speech waveform from the speech parameters.

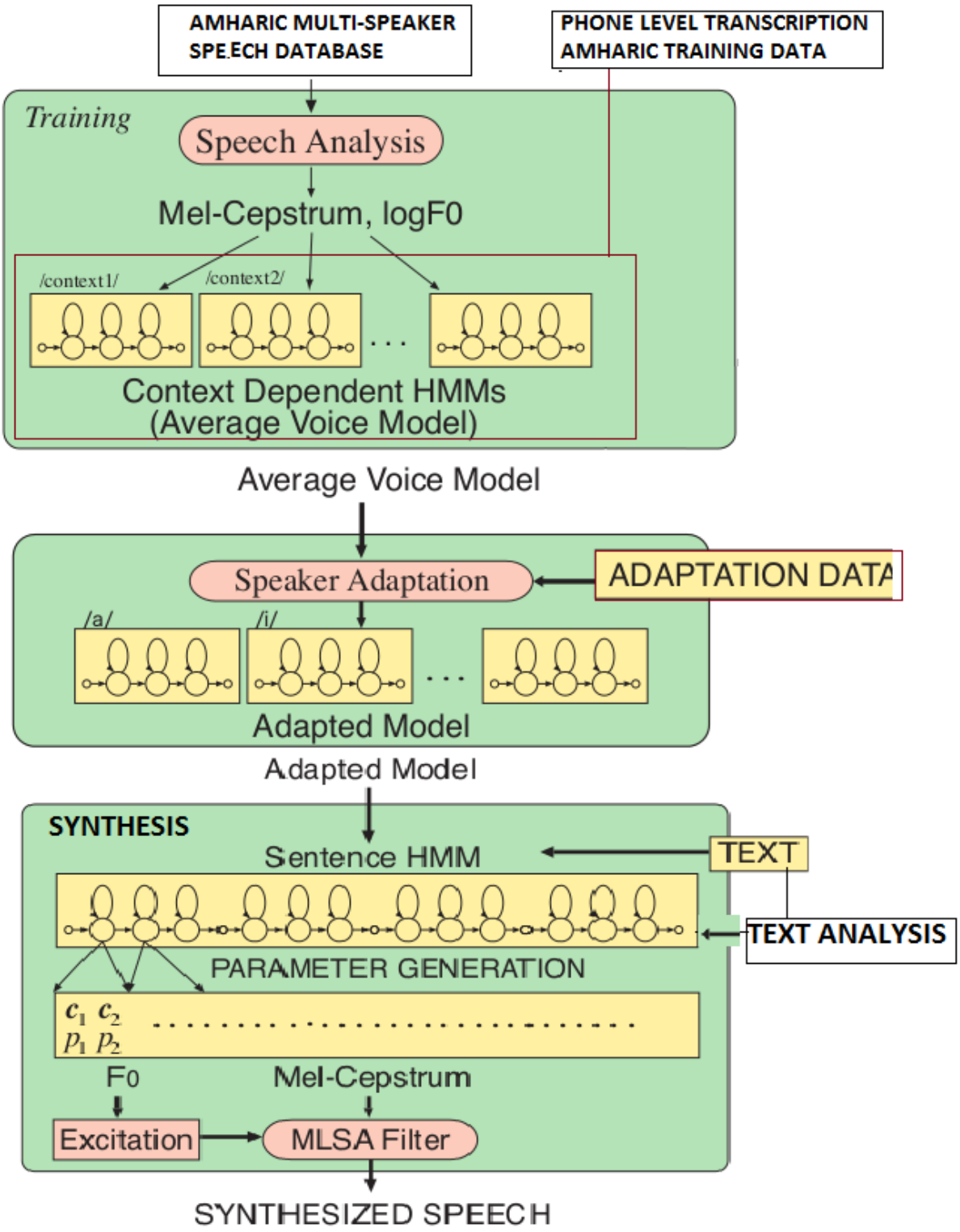


Figure 5-Error! No text of specified style in document.**-1 HMM based Amharic TTS (owner)**

5.2. Amharic corpus preparation

This section discusses the techniques used to design a good corpus for recording for use in general speech synthesis. The basic requirements for a speech synthesis corpus are (Black.2002):

➔Phonetically and prosodic-ally balanced

➔Targeted toward the intended domain(s)

➔ Easy to say by voice talent without mistakes

➔ Short enough for the voice talent to be willing to say it.

Amharic Corpus preparation is guided by The CMU ARCTIC (*Black, 2003)* designed for the purpose of speech synthesis research. The Arctic corpus, consists of nearly 1150 phonetically balanced English utterances. The speech corpus consists of four primary sets of recordings (3 male, 1 female) and single speaker speech databases, which have been carefully recorded under studio conditions for speaker dependent experiments plus several ancillary databases for training set and test sets. these extra files ,automatically segmented phonetic labels, were derived using the standard voice building scripts of the Festvox system (*Black,2000)*. In addition to phonetic labels, the databases provide complete support for the Festival Speech Synthesis System, including pr-built voices that may be used as is. So, Amharic Database prompt list ware created very much in this way.

To begin our data collection we selected a portion of Amharic speech corpus Databases that are designed for training and testing of ASR systems for Amharic it consists of large amounts of speech data by 120 multiple speakers with broadly varying accents and we extracted the text body proper, discarding the surrounding legal matter. We did this not to redistribute the texts absent of or under different copyright, but simply to avoid the expense of collecting and annotating data specifically for TTS. The details of prompt design and extraction are described in the next section as follows.

5.2.1. Designing the Amharic prompt

Our design decisions have been guided by the needs of building HMM voices operating with phoneme sized units and natural coverage. In the near term it is more tractable to design databases that are relatively small. This makes it easier to release multiple versions by multiple speakers with

broadly varying accents thereby enabling a larger variety of voices to be built and studied. with broadly varying accents. These characteristics are not well suited for constructing synthetic voices using unit selection method but HMM enabling solves this shortcoming.

Designing the Amharic prompt set followed seven stages.

- ✓ Decide on a target technology. (HMM synthesis)
- ✓ Decide on the target domain. (Short sentence 5-15 word length)
- ✓ Select a document source. (Project Amharic ASR)
- ✓ Select source documents.
- ✓ Automatically select sentences from the source text.
- ✓ Inspect and remove unsuitable sentences.
- ✓ Perform a trial recording and prune out difficult utterances.

5.2.2. Automatic Prompt Selection

Starting with our initial text corpus of 212 thousand words and 10 thousand utterances, we ran the Festvox script text2utts. This gave us a list of one thousand "nice" utterances. By nice we mean utterances (sentences or phrases) that are easily read by a native Amharic speaking voice talent. This has two aspects: length and pronounceable.

With respect to length, we filtered out sentences that are between 5 and 15 words. Short utterances often have a different prosodic delivery than sentences of normal length. Excluding these, though, does mean that synthesizers built from this database are likely to be less than optimal for reading very short phrases. Conversely, sentences longer than 15 words are difficult to read aloud without making a mistake. It is hard enough already to read over a thousand utterances consistently and correctly. Not being especially interested in modeling speech dis fluencies, we cannot afford to make the task more strenuous by the inclusion of lengthy sentences. The second key restriction is that all words of a selected utterance must already be defined in lexicon or word pronunciation dictionary. Although Festival has reasonable letter to sound rules, to reduce the chance of predicting pronunciations differently from how speakers actually say the prompts. Restricting sentences to contain only known dictionary words helps reduce (but not completely eliminate) errors of this kind. We consider including words with only a single pronunciation (i.e. by excluding homographs) but that turns out to be excessively restrictive.

Next the Festvox dataset_select script was run to search for the subset of the 1000 nice utterances

having the best phone coverage. In order to encourage more thorough coverage I tagged vowels with the stress value (0 or 1) of the syllable in which they are contained. Note that dataset_select employs a greedy algorithm, an elaborate method for selection based on coverage of a given large corpora of text and using an explicit modeling of the acoustic-phonetics of a particular speaker, and so is unlikely to find the global optimum, but will come close. In the construction of Amharic corpus, however, we used the simpler approach encoded in the dataset_select script, as it appears sufficient. At this point 277 utterances had been extracted from the candidate set. These were removed, followed by a second run of dataset_select. This resulted in a second set of 767 utterances with good phone coverage. This second list is smaller than the first because phones that appear only once in the corpus have already been extracted during the first pass.

5.2.3. Further Hand Pruning

The results of automatic selection are still not ideal. We further winnowed the prompt lists in two stages of hand pruning. The first examination is simply based on visual inspection. Criteria for exclusion include: archaic terminology, awkward grammar, confusable homographs, hard to pronounce foreign names, and various embarrassments such as swear words.

Next, we performed trial recordings of the prompt set and removed utterances deemed too hard to liable of mispronunciation. Deciding on the exact cutoff is tricky. In the end, the reduced sets A and B contain 406 and 320 prompts respectively, for a total of 726. Finally we normalized punctuation and updated spelling. Utterances should resemble declarative sentences in that they begin with a letter and end with a period. All these alterations help to deliver the prompts under consistent control. The Amharic prompt set is not perfect but does achieve the objectives of the thesis.

Rang	Occurrence
1000-500	14
500-100	44
100-50	31
50-25	38
25-10	38
<10	35
Total	200

Table 5-1 Number of utterances through three stages of filtering

	Female				Male			
	FAB	FHI	FWO	FTE	MAB	MHA	MBE	MTA
1-14	A	A	A	A	A	A	A	A
15-37	A-B	A-B	A-B	A-B	A-B	A-B	A-B	A-B
38-47	A-C	A-C	A-C	A-c	A-C	A-C	A-C	A-C
	48	48	48	48	48	48	48	48
	192				192			
Total	384							

Table 5-2the number of sentences per speaker and corresponding sentence sets

Tables 5.2 show the number of sentences per speaker and corresponding sentence sets used for training. The sentence sets A–C consist of 40 sentences, respectively.

As explained in section 5.1, one of the criteria used in the design of the Amharic prompt set is that it exhibit good phone coverage with a minimal amount of text.

As explained in section 5.1, one of the criteria used in the design of the Amharic prompt set is that it exhibit good phone coverage with a minimal amount of text.

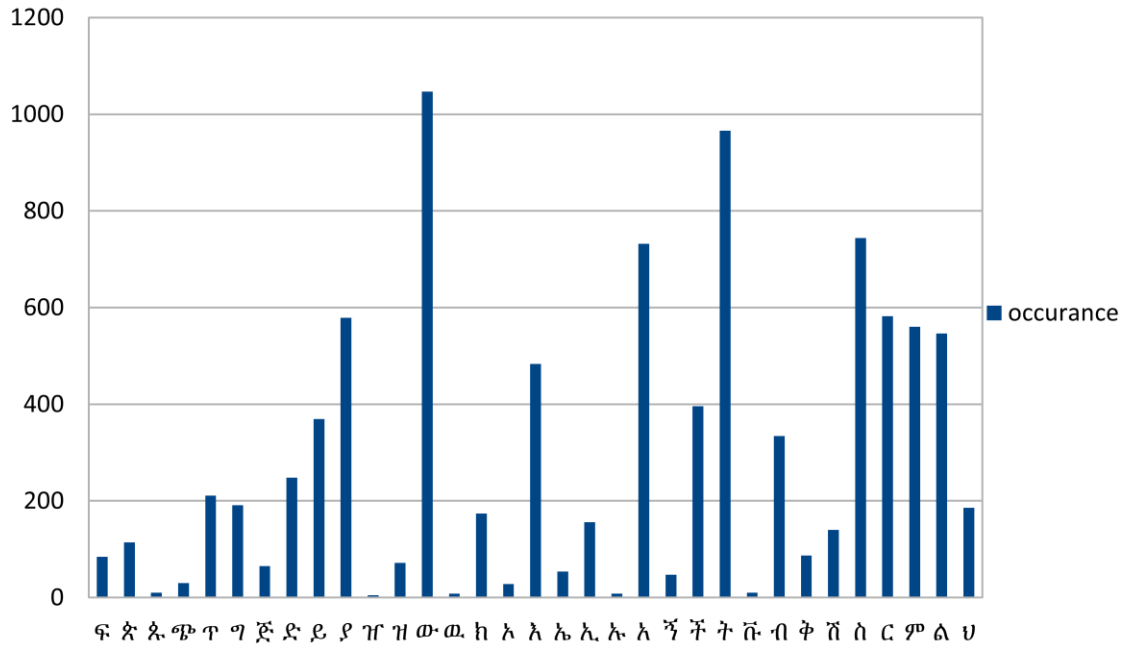


Figure 5.2: phoneme distribution

In the following table we see that the 767 prompts of Amharic corpus, 34 phones, and 200 Amharic characters translate into 25,393 phonemes. This figure comes from the front-end element of Festival that converts prompts into phonemic label files, and includes silence phones

at the beginning and ending of each utterance. For this tabulation the phoneme set has 41 elements consisting of the 34 phonemes from Amharic letter to sound rules, plus the reduced vowel schwa /ix / and the pause symbol / PAU/. The percentages for phone coverage are based on simple combinatory, not on an exhaustive list of n-gram phoneme sequences that are realizable in Amharic. Thus the number of possible phones is 204 = 34*6.

Amharic corpus achieves 200 phones nearly 98%. This phone coverage is significantly offer greater Triphone coverage and full Amharic phoneme coverage. Though the Amharic corpus offers a significant resource for speech synthesis research, it is not all-purpose. Also, the corpus needs the addition of a test set for system evaluation using Festvox script train-test command. This gave us a list of 80 utterances for test set. The design of Amharic corpus has been guided by the needs of HMM -based speech synthesis and that it will prove useful for prosody analysis, and voice conversion, among other things yet to be devised.

phoneme	occurrence
ወ	1047
ት	966
በ	782
የ	753
ስ	744
አ	732
ለ	680
ተ	605
መ	585
ር	582
ያ	579
ም	560
ል	546
ነ	539
እ	483
ና	460
ች	396
ደ	376
ይ	369
ስ	350
ላ	337
ብ	334
ገ	319
ሚ	311
ራ	295
ረ	290

Table 5.-3: Amharic character distribution

5.2.3. Amharic speech corpus preparation

We made three pr-processing stages on the speech corpus the first preprocessing task normalization speech corpus. The waveform files was normalized and changed to conform to 16 KH, 16 bit, RIFF format as required by the festvox system and to make it easier to create raw files of small sizes. The command used to achieve this normalization is bin/get_waves recording/*.wav. The new wave files were then converted to little en din raw files using EST. the second preprocessing activity was F0 extraction to estimate where the speech starts and ends. Using the festvox command ./bin/prune_silence wav/*.wav . A third pr-processing option is to shorted intra sentence silences using command ./bin/prune_middle_silence wav/*.wav, it can help labeling.

5.2.4. Labeling and voice building

As these databases are designed for speech synthesis research, thus the Festvox voice building tools ware used to build a complete HMM based voice for each of the recorded sets. Importantly, this process provides phonetic timing labels and the construction of Festival-style utterance structures, from which durational and other linguistic models may be derived.

There are several algorithms that can be used for labeling in addition to embedded EHMM labeler. An alternative to use CMU Sphinx Train is Hand labeling is also another alternative but it is a very tiresome task and it requires expertise. This experimentation covers one techniques of speech labeling, namely automatic speech segmentation using embedded EHMM labeler. Automatic labeling is implemented using festvox script "./bin/label" used to complete the task of automatic speech labeling. Hand labeling can produce even better results especially when applied to improve on an already automatically labeled data. In this stage no hand correction was done on labels. Corrected labels not only yield better sounding voices, but serve as a benchmark against which automatic labeling techniques can be compared.

5.3. Text Analysis

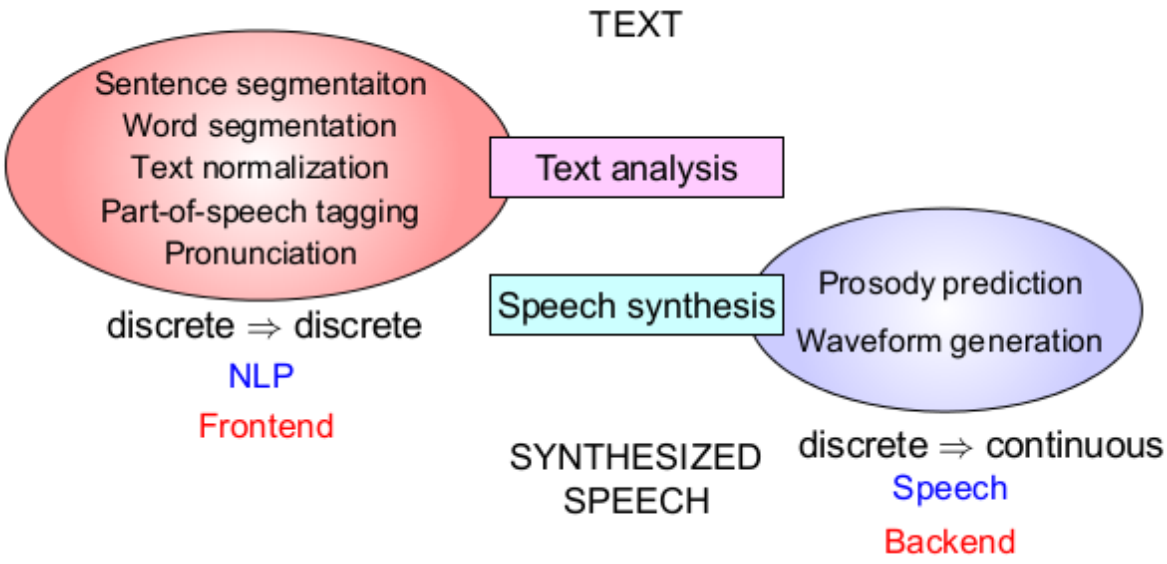


Figure 5.-2 typical flow of TTS system (Paul Taylor, 2007)

According to (Paul Taylor, 2007) the following text analysis and prosody prediction is processes ware applying for this thesis as follows;

1. Pr-processing: possible identification of text genre, character encoding issues, possible multi-lingual issues. Many languages, in particular Amharic languages have far more characters than can be included in 8 bits, and so cannot be accommodated in single byte encoding schemes. For this case we use UTF-8 character encoding .UTF-8 encodes enough extra characters for Amharic languages and the festvox schemes have been proposed to support UTF-8 character encoding.
2. Sentence splitting: segmentation of the document into a list of sentences.
3. Tokenization: segmentation of each sentence into a number of tokens, possible processing of XML.
4. Text analysis

a) Semiotic classification: classification of each token as one of the semiotic classes of natural language, abbreviation, quantity, date, time etc.

b) Decoding/parsing: finding the underlying identities of tokens using a decoder or parser

that is specific to the semiotic class.

c) Verbalization: Conversion of non-natural language semiotic classes into words.

1. Homograph resolution Determination of the correct underlying word for any ambiguous natural language token.

2. Parsing Assigning a syntactic structure to the sentence

3. Prosody prediction attempting to predict a prosodic form for each utterance from the text. This includes:

A) Prosodic phrase break prediction

B) Prominence prediction

C) Intonation tune prediction

5.3.1. Amharic Phone-set

Two phone-set was used for this thesis for Amharic languages. The first is using Festvox X-SAMPA grapheme-phoneme mappings and the second using the standard CMU phone-set for the transcribed training dataset.

All Unicode characters defined for Amharic languages are first mapped to a phoneme from the X-SAMPA set, similar to the mappings provided by Unitary. In addition, each Unicode character is mapped to its corresponding ordinal for ease of processing. Lastly, a set of rules are defined which are associated with language lists. The SAMPA phonetic alphabet for Amharic is used for the phonemic transcription. An extensive lexicon deals with known words, and a letter-to-sound conversion algorithm with unknown words; but first, a dedicated module adds inflection endings to ordinals and abbreviations.

In Festvox, the phone set of the language is described with the corresponding features like voicing, tongue position, tongue height, place of articulation, and manner of articulation. From the studies reported by (Sebsibe et al, 2005), we derived a set of phonetic features for the 39 phones. The lists of the phone sets are mentioned in table 5.4 (a) and (b).

The way Amharic orthographic characters are written is very similarly to the way they are spoken. It means Amharic is a phonetic language. The mapping of the written form and the spoken form is one to one except the epenthetic vowel which is mentioned above in transliteration scheme. Section ( ) covers method for finding the pronunciation of a word. This is

IPA	Amharic X-SAMPA									
	phone		vc	vl	vh	vf	vr	ct	cp	cv
[h]	ሀ	h	-	0	0	0	+	f	g	+
[l]	ለ	l	-	0	0	0	+	l	a	+
[m]	መ	m	-	0	0	0	+	n	l	+
[r]	ረ	r	-	0	0	0	+	l	a	+
[s]	ሰ	s	-	0	0	0	+	f	a	-
[ʃ]	ሸ	š	-	0	0	0	+	f	p	-
[k']	ቀ	q	-	0	0	0	+	s	v	+
[t]	ተ	t	-	0	0	0	+	s	a	+
[ʧ]	ቸ	č	-	0	0	0	+	a	p	+
[n]	ነ	n	-	0	0	0	+	n	p	+
[ɲ]	ኘ	ñ	-	0	0	0	+	n	p	+
[k]	ከ	k	-	0	0	0	+	s	v	-
[w]	ወ	w	-	0	0	0	+	r	l	+
[y]	ዐ	y	-	0	0	0	+	r	p	+
[d]	ዘ	d	-	0	0	0	+	s	a	+
[ʤ]	ጀ	j/ǧ	-	0	0	0	+	a	p	+
[g]	ገ	g	-	0	0	0	+	s	v	+
[t']	ጠ	t'/ṭ	-	0	0	0	+	s	a	+
[f]	ፈ	f	-	0	0	0	+	f	l	-
[p]	ፐ	p	-	0	0	0	+	s	l	-
[b]	በ	b	-	0	0	0	+	s	l	+
[ts]	ጸ	ts'	-	0	0	0	+	f	a	+
[ʧ']	ጨ	č	-	0	0	0	+	a	p	-
[p']	ጰ	p'	-	0	0	0	+	s	l	-
[ʒ]	ፘ	zh	-	0	0	0	+	f	p	+
[v]	ቨ	v	-	0	0	0	+	f	l	+
Amharic Vowel										
[a]	ኧ	ā/ā	+	d	2	2	-	0	0	0
[u]	ኡ	u	+	d	1	3	+	0	0	0
[I]	ኢ	ī/i	+	l	1	1	-	0	0	0
[a]	ኣ	a	+	a	3	2	+	0	0	0
[e/ɛ]	ኤ	ē/e	+	d	2	1	-	0	0	0
[ə]	እ	(ə)	+	l	1	2	-	0	0	0
[o]	ኦ	o	+	d	2	3	+	0	0	0

Table 5-4 (b): Amharic Phone-set for transcribed

phone	vc	vl	vh	vf	vr	ct	cp	cv
(pau	-	0	0	0	0	0	0	-)
(aa	+	l	3	3	-	0	0	0)
(ae	+	s	3	1	-	0	0	0)
(ah	+	s	2	2	-	0	0	0)
(ao	+	l	3	3	+	0	0	0)
(aw	+	d	3	2	-	0	0	0)
(ax	+	a	2	2	-	0	0	0)
(axr	+	a	2	2	-	r	a	+)
(ay	+	d	3	2	-	0	0	0)
(b	-	0	0	0	0	s	l	+)
(ch	-	0	0	0	0	a	p	-)
(d	-	0	0	0	0	s	a	+)
(dh	-	0	0	0	0	f	d	+)
(dx	-	0	0	0	0	s	a	+)
(eh	+	s	2	1	-	0	0	0)
(er	+	a	2	2	-	r	0	0)
(ey	+	d	2	1	-	0	0	0)
(f	-	0	0	0	0	f	b	-)
(g	-	0	0	0	0	s	v	+)
(hh	-	0	0	0	0	f	g	-)
(ih	+	s	1	1	-	0	0	0)
(iy	+	l	1	1	-	0	0	0)
(jh	-	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	+)
(ng	-	0	0	0	0	n	v	+)
(ow	+	d	2	3	+	0	0	0)
(oy	+	d	2	3	+	0	0	0)
(p	-	0	0	0	0	s	l	-)
(r	-	0	0	0	0	r	a	+)
(s	-	0	0	0	0	f	a	-)
(sh	-	0	0	0	0	f	p	-)
(t	-	0	0	0	0	s	a	-)
(th	-	0	0	0	0	f	d	-)
(uh	+	s	1	3	+	0	0	0)
(uw	+	l	1	3	+	0	0	0)
(v	-	0	0	0	0	f	b	+)
(w	-	0	0	0	0	r	l	+)
(y	-	0	0	0	0	r	p	+)
(z	-	0	0	0	0	f	a	+)
(zh	-	0	0	0	0	f	p	+)

Table 5.-4 (a): Amharic IPA and Phone-set

5.3.2. Amharic Numbers

The pronunciation of numbers highly depends on their meaning. Different number types, such as cardinal and ordinal numbers, currency amounts, or telephone numbers, must be identified as such, either from input text or from context, and replaced by appropriate token strings. While the expansion of cardinal numbers is straightforward, the expansion of ordinal numbers poses interesting problems in Amharic, because of their inflections.

The following rule is applied in this thesis for Amharic numbers the snap shott is :

when the number length equal to one in other words when one number is present, between [0-9] synthesis the corresponding token.

B) when the number length equal to two in other words when two numbers meet together less than one hundreds,

I) when 1 and 0 meet together, list asere, else list asera plus the next number.

II) when 2 and 0 meet together, list haya, else list haya plus the next number.

III) when 3 and 0 meet together, list selasa, else list selasa plus the next number.

IV) when 9 and 0 meet together, list zetena, else list zetena plus the next number.

C) when the number length equal to three in other words when three numbers meet together less than thousand ,if 1 plus just zeros , list meto, else list A plus meto plus B. for example 121 “ande meto haya ande.

D) when the number length equal to four in other words whenfour numbers meet together less than ten thousand ,if 1 plus just zeros , list “Shi”, else list A plus Shih plus C. for example 1121 “ande shih ande meto haya ande.

E) when the number length equal to five in other words when five numbers meet together less than one hundred thousand ,if 1 plus just zeros , list “asere shi”, else list B plus Shih plus C. for example 51121 “hamsa ande shih ande meto haya ande.

```
(define (aau_amharic::number_from_digits digits)
  "(aau_amharic::number_from_digits digits)
Takes a list of digits and converts it to a list of words
saying the number."
  (let ((l (length digits)))
    (cond
     ((equal? l 0)
      nil)
     ((string-equal (car digits) "0")
      (aau_amharic::number_from_digits (cdr digits)))
     ((equal? l 1);; single digit
      (cond
       ((string-equal (car digits) "0") (list "ዜሮ"))
       ((string-equal (car digits) "1") (list "አንድ"));;ሀያ አንድ
       ((string-equal (car digits) "2") (list "ሁለት"))
       ((string-equal (car digits) "3") (list "ሶስት"))
       ((string-equal (car digits) "4") (list "አራት"));;ሶ ስት ለ አ ራት
       ((string-equal (car digits) "5") (list "አምስት"))
       ((string-equal (car digits) "6") (list "ስድስት"))
       ((string-equal (car digits) "7") (list "ሰባት"))
       ((string-equal (car digits) "8") (list "ስምንት"))
       ((string-equal (car digits) "9") (list "ዘጠኝ"))
       ;; fill in the rest
       (t (list "equis"))));; $$$ what should say?
```

```
     ((equal? l 2);; less than 100
      (cond
       ((string-equal (car digits) "0");; 0x
     (aau_amharic::number_from_digits (cdr digits)))

       ((string-equal (car digits) "1");; 2x
     (if (string-equal (car (cdr digits)) "0")
         (list "አስር")
         (cons "አስራ"(aau_amharic::number_from_digits (cdr digits)))))

       ((string-equal (car digits) "2");; 2x
     (if (string-equal (car (cdr digits)) "0")
         (list "ሀያ")
         (cons "ሀያ"(aau_amharic::number_from_digits (cdr digits)))))

       ((string-equal (car digits) "3");; 3x
     (if (string-equal (car (cdr digits)) "0")
         (list "ሰላሳ")
         (cons "ሰላሳ"(aau_amharic::number_from_digits (cdr digits)))))

       ((string-equal (car digits) "4");; 4x
     (if (string-equal (car (cdr digits)) "0")
         (list "አርባ")
         (cons "አርባ"(aau_amharic::number_from_digits (cdr digits)))))

       ((string-equal (car digits) "5");; 5x
     (if (string-equal (car (cdr digits)) "0")
         (list "አምሳ")
         (cons "አምሳ"(aau_amharic::number_from_digits (cdr digits))))
```

```
    ((equal? l 3);; in the hundreds
     (cond

      ((string-equal (car digits) "1");; 1xx
     (if (just_zeros (cdr digits)) (list "መቶ")
         (cons "መቶ"(aau_amharic::number_from_digits (cdr digits)))))

      ((string-equal (car digits) "5");; 5xx
     (cons "አምስት·መቶ"(aau_amharic::number_from_digits (cdr digits))))

      (t;; ?xx
     (append (aau_amharic::number_from_digits (list (car digits)))
         (list "መቶ")
         (aau_amharic::number_from_digits (cdr digits))))
      ))

    ((< l 7)

      (let ((sub_thousands
          (list
           (car (cdr (cdr (reverse digits))))
           (car (cdr (reverse digits)))
           (car (reverse digits))))
         (thousands (reverse (cdr (cdr (cdr (reverse digits))))))))
     (set! x (aau_amharic::number_from_digits thousands))
     (append
      (if (string-equal (car digits) "1"))
      (list "ሺ")
      (aau_amharic::number_from_digits sub_thousands))))
```

5.3.3. Punctuation Marks

We define the following Amharic punctuation Marks for the the task of sentence splitting, to take the raw document and segment it into a list of sentences, and word splitting ,to create a separate token for each punctuation character. The most difficult case is . as this can be used as part of an abbreviation, a number and so on. In Amharic writing, a space always follows the full stops / arate neteb“::”other use of “ .” and / is in abbreviations. In fact, in Amharic writing, the use of .in this sort of abbreviation is not common.

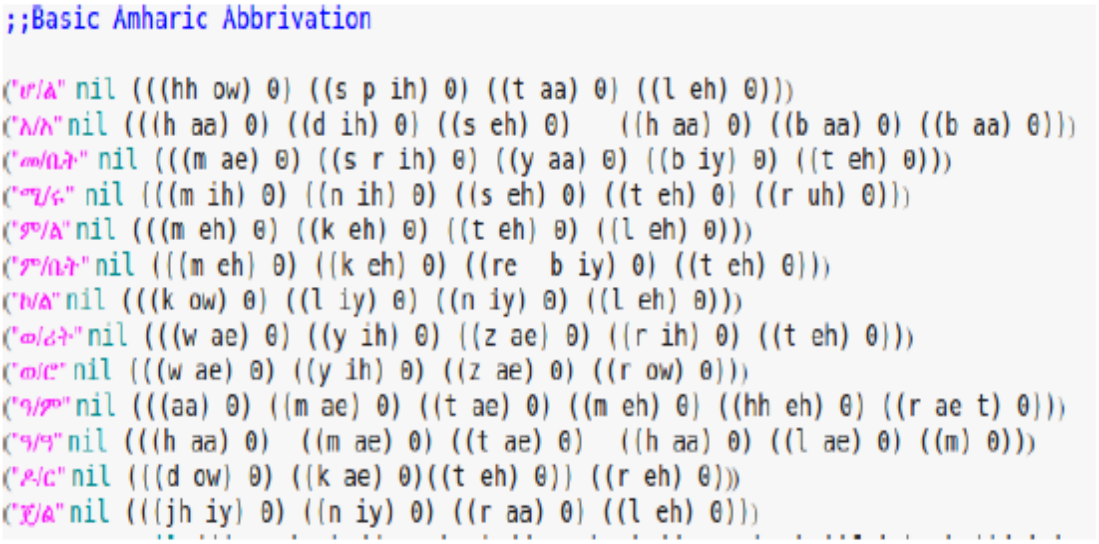
```
;; Punctuation for the Amharic language
(set! aau_amharic_amharic::token.punctuation "\"'`.,:;!?(){}/“”[]:")
(set! aau_amharic_amharic::token.prepunctuation "\"'`({[")
(set! aau_amharic_amharic::token.whitespace " \t\n\r :")
(set! aau_amharic_amharic::token.singlecharsymbols "")
```

```
;; Basic Amharic punctuation must be in with nil pronunciation

(lex.add.entry '("%" n (((b eh) 0) ((m e) 1) ((t o) 0))))
(lex.add.entry '("ᎁ"punc nil));;HURT
(lex.add.entry '("᎓"punc nil));;YIZET
(lex.add.entry '("᎒"punc nil));;DERET
(lex.add.entry '("᎑"punc nil));;RIKRIK
(lex.add.entry '("᎐"punc nil));;SHORTRIKRIK
(lex.add.entry '("᎒"punc nil));;DIFAT
(lex.add.entry '("ᎍ"punc nil));;KENAT
(lex.add.entry '("᎒"punc nil));;CHIRET
(lex.add.entry '("ᎌ"punc nil));;HIDET
(lex.add.entry '("ᎎ"punc nil));;DERET-HIDET
(lex.add.entry '("""" punc nil));;GEMINATIONMARK
(lex.add.entry '("※"punc nil));;SECTIONMARK
(lex.add.entry '("᎟" punc nil));;WORDSPACE
(lex.add.entry '("/"punc nil));;For Abbrivation use
(lex.add.entry '("።" punc nil));;FULLSTOP
(lex.add.entry '("፣" punc nil));;COMMA
(lex.add.entry '("፤"punc nil));;SEMICOLON
(lex.add.entry '("፥"punc nil));;COLON
(lex.add.entry '("፦"punc nil));;PREFACECOLON
(lex.add.entry '("፧"punc nil));;QUESTIONMARK
(lex.add.entry '("?" punc nil)) ;;QUESTIONMARK
(lex.add.entry '("᎔"punc nil));;PARAGRAPHSEPARATOR
```

5.3.4. Abbreviation

Two main groups of abbreviations are distinguished: Those that are spelled out, such as “EFDR”, and those that need expansion. The first group of abbreviations are correctly pronounced by spelling rules. The second group is pronounced using an expansion table, containing a grapheme and optionally a phonemic expansion. The latter is especially useful for foreign abbreviations, such as “FBI” “USA”. The When the inflection endings module finds an ordinal or an abbreviation with an adjectival role, it performs a unification of the morphological variables over the known tokens in the noun phrase to which the ordinal or abbreviation belongs. As Shown in figure 5.6 expansion table is developed that containing a grapheme and optionally a phonemic expansion for over 100 Amharic abbreviations.



5.3.5. Pronunciation Lexicon

The pronunciation lexicon is developed for the top 2000 Amharic words manually. The lexicon performs a simple compound treatment. If a word is not found in the lexicon but is the concatenation of two or more lexicon entries, the corresponding phonemic forms are concatenated. An optional "+s+" bounding morph, typical for Amharic noun compounds, is also allowed. For all parts of a compound except the first, primary word stress is reduced to secondary stress, i.e. the first part is considered the dominant one, which seems to be the default for Amharic. We are often confronted with the problem where we have to determine the correct sequence of words, phonemes or phones from a given a waveform of speech.

A pronunciation requires not just a list of phones but also a syllabic structure. In some languages the syllabic structure is very simple and well defined and can be unambiguously derived from a phone string. Amharic Lexicon is developed to provide the pronunciation of most frequently occurred word in a given training set. Through lexical model, various combinations of phones are defined to give valid words for the Synthesizer.

The lexicon structure that is manually developed for this thesis include the top 2000 words with a part of speech to find the given pronunciation. The lexicon development is based on how English is best dealt with and the basic assumption in Festival, having a large lexicon, tens of thousands of entries, that is a used as a standard part of an implementation of a voice. Then Letter-to-sound rules are used as back up when a word is not explicitly listed. However this is a very flexible

view, An explicit lexicon isn't necessary in Festival (Black 2002). so the top 2000 words with their part of speech were used as a bootstrap and much of the work was done using letter-to-sound rules (see section 5.3.7).

```
(lex.add.entry '("ሀላፊ" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0))))
(lex.add.entry '("ሀላፊነቱ" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((n ax) 0) ((t uh) 0))))
(lex.add.entry '("ሀላፊነታቸው" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((n ax) 0) ((t aa) 0) ((ch ax) 0) ((w eh) 0))))
(lex.add.entry '("ሀላፊነት" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((n ax) 0) ((t eh) 0))))
(lex.add.entry '("ሀላፊው" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((w eh) 0))))
(lex.add.entry '("ሀላፊዎቹ" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((w ao) 0) ((ch uh) 0))))
(lex.add.entry '("ሀላፊዎች" nil (((hh ax) 0) ((l aa) 0) ((f ih) 0) ((w ao) 0) ((ch eh) 0))))
(lex.add.entry '("ሀለጠ" nil (((hh ax) 0) ((l ae) 0) ((b eh) 0))))
(lex.add.entry '("ሀሌታ" nil (((hh ax) 0) ((l ae) 0) ((t aa) 0))))
(lex.add.entry '("ሀል" nil (((hh ax) 0) ((l eh) 0))))
(lex.add.entry '("ሀሉ" nil (((hh ax) 0) ((l ao) 0))))
(lex.add.entry '("ሀመልማል" nil (((hh ax) 0) ((m ax) 0) ((l eh) 0) ((m aa) 0) ((l eh) 0))))
```

5.3.6. Letter-to-sound conversion

Unknown words that cannot be phonemic with the help of the lexicon are analyzed by a "letter-to-sound conversion" algorithm. This algorithm is more complex than a simple application of letter-to-sound rules: On the one hand, correct phonemically relies in many cases on a correct identification of morpheme boundaries. On the other hand, for the phoneme string to be properly uttered, syllabification and word stress information needs to be added.

The following syllabification rules are adopted to define the pronunciation of Amharic words:

when two letters of six order meet together at the end of a vowel,both are mute,unless want of organic affinity,or gemination their being so;but such a word is augmented at the end the last letter of their order is sounded. "ስ ፖር ት " nil (s ax) 0) ((p ao) 0) ((r t) 0) ;

when a letter of the sixth form commences a word,its vowel is generally sounded. ("እ ጅ" nil (((eh) 0) ((jh eh) 0)))

In trilateral words ,where all the three letters are of the sixth order,the first is generally sounded; the two following are not; "ች ግ ር " nil (((ch) 0) ((g r) 1)) ; ("ይ ህ ን " nil (((y eh) 0) ((hh n) 1)

in trilateral words,where the two first letters are of the six order,the first is sounded;the second is not; ("ህ ዝ ቡ" nil (((hh eh) 0) ((z) 1)((b uh) 0))) ("ህ ብ ረ ት " nil (((hh eh) 0) ((b eh) 1) (r ax) 0) ((t) 0)))

The aim of pronunciation is to convert the discrete, linguistic, word based representation generated by the text analysis system into a continuous acoustic waveform (Taylor,2007). This can be thought of as a system which takes the word-based linguistic representation and generates

a phonemic or phonetic description of what is to be spoken by the subsequent waveform synthesis component. In generating this representation, we make use of a lexicon to find the pronunciation of words we know and can store. After doing this we may find that simply concatenating the pronunciations for the words in the lexicon may not be enough; words interact in a number of ways and so a certain amount of post-lexical processing is required. However, post-lexical rules are not considered in this work. Finally, there is considerable choice in terms of how exactly we should specify the pronunciations for words, and hence rigorously defining a pronunciation representation is in itself a key topic.

5.4. Developing HMM-Based Amharic Speech Synthesizer

The proposed Amharic text-to-speech synthesis system is based on the speech parameter generation algorithm from HMMs, and a Mel-cepstral speech analysis/synthesis technique. A block diagram of the proposed Amharic text-to-speech synthesis system is shown in Fig.5.1, which is almost equivalent to the general HMM-based text-to-speech system except that multiple speaker's HMM sets are trained and interpolated to generate a new speaker's HMM set.

The procedure can be summarized (Yoshimura, 2002) as follows:

Training representative HMM sets

Select several representative speakers S 1 , S 2 ,. . . , S N appropriately from speech database, and repeat (b), (c) for each speaker.

Obtain Mel-cepstral coefficients from speech of the representative speaker by Mel-cepstral analysis.

Train phoneme HMM set λ k using Mel-cepstral coefficients, and their deltas and delta-deltas.

2. Changing and adopting representative HMM sets

A) Generate a new phoneme HMM set λ with untrained speaker's characteristics by interpolating between the representative speakers' phoneme HMM sets λ 1 , λ 2 , . . ., λ N with an arbitrary interpolation ratio a 1 , a 2 , . . . , a N. we can gradually change the characteristics of synthesized speech from one's to the other's by changing the interpolation ratio.

B) Generate a new phoneme HMM set λ with untrained speaker's characteristics by adopting the generated phonemes to the target speaker by using MLLR algorithm.

3. Speech Synthesis from Interpolated HMM

Transform the text to be synthesized into a phoneme sequence, and concatenate the in-interpolated and adopted phoneme HMMs according to the phoneme sequence.

Generate speech parameter sequence from the sentence HMM by using Decision-tree based

model clustering algorithm.

Synthesize speech from the generated Mel-cepstral coefficients by using the MLSA (Mel Log Spectral Approximation) filter.

5.4.1. Feature Extraction

Speech signals typically need to be divided into small frames before Training can begin. Analysis of these frames can then determine the likelihood of a particular phoneme being present within the frame. Speech is non-stationary in the sense that frequency components change continuously over time, but it is generally assumed to be a stationary process within a single frame. Mel-spectrum methods currently used in speech analysis usually do not consider where phonemes begin and end, which causes complications to appear at the boundaries of phonemes.

General Speech setting	
HeaderBytes:	44
SamplingFreq:	16000
FrameSize:	160
FrameShift:	80
Lporder:	12
CepsNum:	16

Table 5.5: Speech analysis / parameter extraction settings

As discussed in chapter two, section 2.7.1, two parameters, the spectrum and excitation, are needed to train the model. The spectrum parameters consist of Mel-spectrum or linear prediction coefficient. On the other hand, the excitation parameter consists of the fundamental frequency (F0). In this thesis work Mel-cepstral coefficients are used as spectrum parameters. Using the raw data generated during data preparation, these speech parameters (features) are extracted using the tool SPTK-3.6.

For this research work, the acoustic units (phonemes) are selected during acoustic unit inventory

design process stage because phonemes are the basic unit to train the model HMM. (Toda et al, 2007) proposed Gaussian Mixture Model (GMM) the inversion mapping without constraints on phonetic information. The spectrum parameters consist of Mel-spectrum or linear prediction coefficient. On the other hand, the excitation parameter consists of the fundamental frequency (F0).

Speech signals were sampled at a rate of 16 kHz and they were windowed by using a 25-ms Blackman window with a 5-ms shift. Then, Mel-cepstral coefficients were obtained by Mel-cepstral analysis. Fundamental frequency was extracted using the festvox F0_sptk script. The feature vectors consisted of 25 Mel-cepstral coefficients including the zeroth coefficient, the logarithm of the fundamental frequency, and their delta and delta-delta coefficients. To incorporate all features, 5 streams of data were used within each state; where stream one is used for Mel-spectrum coefficients including their delta and delta-delta values and the other four streams were used to model the F0 and their delta and delta values. Using multiple acoustic frames and multiple mixtures is obviously effective for improving the mapping accuracy (Yoshimura, 2002). However, the degradation of accuracy is caused by excessively increasing the number of these parameters because a larger amount of training data is needed for estimating a larger number of parameters. Moreover, it can be observed that the optimum number of mixtures for each number of acoustic frames decreases as the number of acoustic frames increases. Consequently, the best mapping accuracy is achieved when the number of acoustic frames is set to 5 and the number of mixtures is set to 25 (Yoshimura, 2002).In this experiment dataset of 9174 vectors of 70 parameters 118 unit-types tree models as 4646 sub unit types vector parameters are dumped. In that case, RMSE is 0.1542 and Correlation is 0.9854 Mean (abs) Error is 0.0932 (0.1229) for the female speaker, and the RMS error is 0.24 mm (0.985 on abs) and the correlation coefficient is 0.74 for the male speaker.

5.4.2. Defining the structure of the HMM

Defining the structure of the HMM is the first step towards building a synthesizer using Hidden Markov Model. To do so a decision has to be made on the structure of the acoustic models, mono-phone HMMs, tied state HMMs or multiple mixture HMMs and on the number of states and whether to include dynamic feature of the Mel-spectrum coefficients and fundamental frequencies in modeling the speech signal (Yoshimura,2002).

HMM modeling without tied state or context-independent HMM: primarily letters/phonemes are modeled with mono-phone HMMs without considering its context. This is also termed as

acoustic modeling without tied state.

Context-dependent HMMs can better model the spectral movements in phonetic transitions. It implies that acoustic modeling which takes its context into consideration is also required as second technique of acoustic modeling. Having a set of mono-phone HMMs, trip-hone HMM model building is required to create context- dependent trip-hone HMMS. This is done in two steps. Firstly, conversions of mono-phone transcriptions into trip-hone transcriptions takes place and create a set of trip-hone models by copying the mono phones and re-estimating them. Secondly, similar acoustic states of trip hones are tied to ensure that all state distributions are robustly estimated. Context-dependent trip hones are made by simply cloning mono phones and then re-estimating using trip-hone transcriptions (Yoshimura, 2002). Then states are tied within trip-hone sets in order to share data and thus be able to make robust parameter estimates. Although some of them showed acceptable performances, most of the HMM -based method uses context sensitive trip-hone model to resolve co-articulation effect.

Acoustic modeling with Multiple Gaussian mixtures values are used to improve automatic speech segmentation and synthesis results considerably, because they help avoid the problem resulting from the usage of the same type of probability density distribution for different models and states. So that context dependent HMMs with different probability distributions are used for further improvement of speech synthesis. If an HMM state is made to contain multiple Gaussian mixture components, then the training vectors would be associated with highest likelihood mixture component. The number of vectors associated with each component within a state can then be used to estimate the mixture weights (Yoshimura, 2002).

In this thesis work, after making a comparison between different models having different number of states, a 3-state left-to-right trip-hone models with multiple Gaussian output distributions (multiple mixture HMM) model is chosen and applying default HTS configuration.

5.5. HMM training

5.5.1. Experiment #1 Building initial model

In the proposed system, there is no need for label boundaries when appropriate initial models are available since spectral, F0 and duration models are estimated by the embedded training. Therefore, the system can be automatically constructed by the following process:

As initial model, speaker independent and gender dependent model is prepared.

The target speaker is estimated by using initial model and speech data with transcription.

As initial model, gender dependent (GD), and speaker independent model (SI) was proposed. For GD, speech data from 3 male speakers in which target speaker MAB did not include was used. For Speaker Independent Training 476 sentences uttered by 3 male and 3 female speakers in which target speaker MAB did not include was used (these models denoted as SI models). The models were state clustered using a tree based clustering procedure. The total number of states in SI and GD models is 2213 and 1442 respectively. Table 5.6 and Table 5.7 show the result of the Preference scores of GD and SI models. The difference of quality of speech synthesized by using each initial model was evaluated by a pair comparison test. Preference scores are shown in table.5.8. From these table, it is seen that the difference of speech quality between initial model GD and SI.

	mean	std	Var	n
F0	26.1753	25.769	664.04	85215
noF0	0.19547	0.21723	664.04	85215
MCD	5.43662	2.25433	288.35	9039
all	7.25887	588.843	5.082	85215

Table 5.6: Preference scores SI

	mean	std	var	n
F0	32.19	15.78	249.16	34280
noF0	0.2	0.17	0.029	822720
MCD	5.66	2.06	4.26	34280
all	9.04	205.89	42394	857000

Table 5.7: Preference scores GD

	MCD	F0
SI	5.43	26
GD	5.66	32

Table 5.8: difference of initial model F0 MCD

5.5.2. Experiment result and discussion

The synthetic speech was compared with objective metric of speech synthesis quality based on the held-out reference scores by computing the Mean Mel- Capstral Distortion (MCD) of the predicted Capistrano, F0 and n0F0. Mean Mel- Capstral Distortion (MCD) is a distance measure, a lower value suggests better synthesis. (Kominek et al, 2009) has suggested that MCD is linked to perceptual improvement in the intelligibility of synthesis, and that an improvement of about

0.08 is perceptually significant and an improvement of 0.12 is equivalent to doubling the data. The MCD is a database-specific metric which cannot be compared across databases. From the table 5.6 -5.7, it is seen that there is no difference of speech quality between initial model GD and SI. The speaker independent training treats the training data which consists of several speakers' speech as that of one speaker and makes no distinctions among the training speakers of the average voice model. Figures 5.5 show the MCD and F0 of GD and SI models. Since F0 is not observed in the unvoiced region, the MCD log F0 error was calculated in the region where both the generated and the real F0 were voiced. The horizontal axes indicate the preference score, and the bars indicate the results for the SI and GD models.

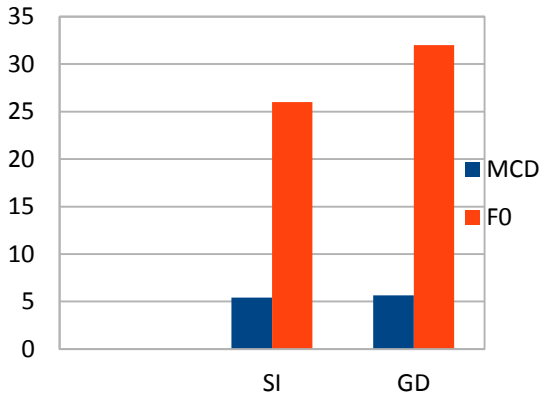


Figure 5.5: Effect of initial model GD and SI models

5.5.3. Experiment #2: Speaker Adaptation using MLLR

Initial (speaker independent) HMMs were trained and the target speaker is set to a male speaker MAB, who was not included in the speakers for training initial HMMs, in the database. Consequently, we adapted SI models to MAB using 120 sentences from the sentences uttered by MAB. It should be noted that the adaptation data was not contained in the training data. Since the amount of adaptation data was very small, we used one regression matrix, tied among all states, for MLLR adaptation.

Spectral Analysis

Spectral analysis, Spectral distances between synthetic speech and real speech, were calculated for the objective evaluation tests. Figure 5.6 and 5.7 shows the spectral ,F0 and Spectrogram

information of the original speech uttered by female speaker and synthetic speech from the average voice models respectively for the Amharic sentence /"ታፈሰ ልብሱን ሲያጥብ ቆየ ." /, which is not included in the training sentences. Dotted line and solid line show the F0 and the Spectrogram contours generated from the original female speaker ,average voice models (SI) and Adopted models, respectively.

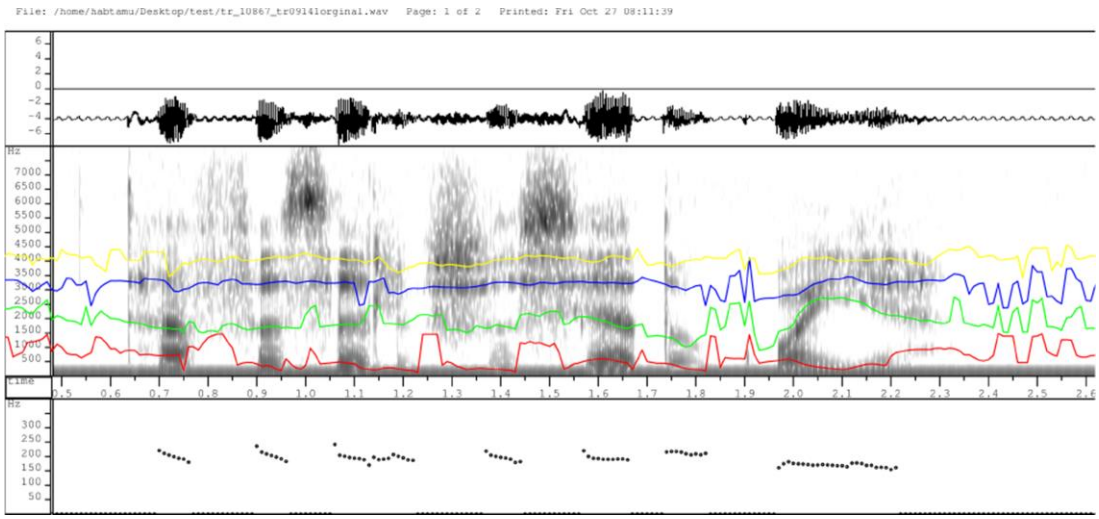


Figure 5.6: the spectral ,F0 and Spectrogram information of the original speech uttered by female Speaker

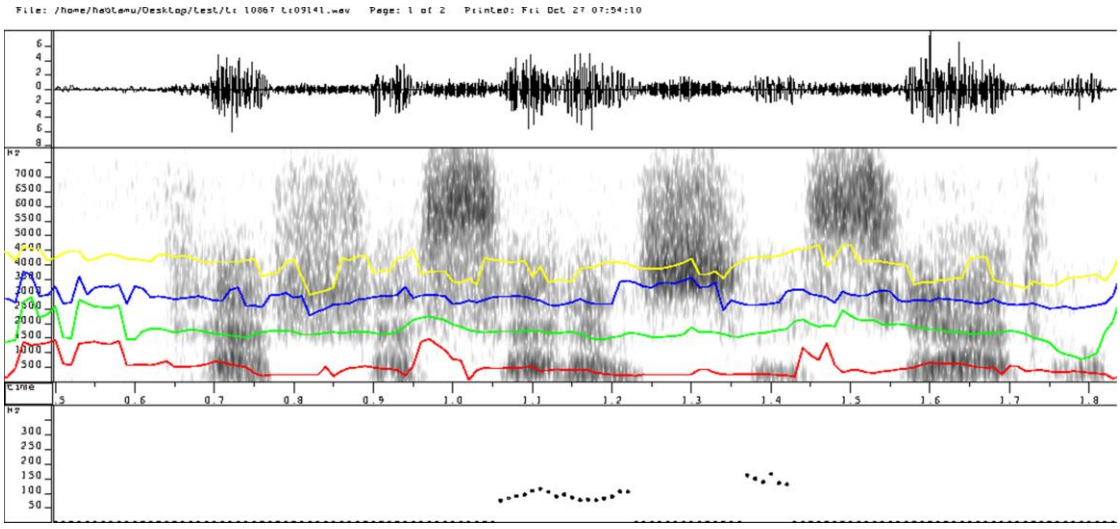


Figure 5.7: the spectral ,F0 and Spectrogram information of the Synthesized Speech

However, from the above figure we can see that there is a significant difference between the F0

contours generated from the speaker independent models and the original speech. This is due to the fact that the size of the intersection of sentence sets increases as the number of sentences for each speaker increases. For example, when the number of sentences for each speaker is 120, the sentence set F is included by all speakers' sentence sets. As a result, the number of leaf nodes biased to a speaker decreases in the speaker independent model.

5.5.4. Experiment #3 Trajectory HMM model SI training

To overcome this problem Trajectory HMM based on shared decision tree context clustering technique were applied to better the naturalness of the synthesized speech already obtained from the main (or SI) TTS synthesis system. An advantage of the technique is that every node of the decision tree always has the data of all speakers. In other words, there is no node lacking one or more speakers' data. This is done using the festvox script ./bin/do_clustergen trajectory_ola. Tabel 5.9 shows that the average voice models constructed using the shared decision tree context clustering technique. It can be seen that the average voices using the training data with F0 normalization sound more natural than those without F0 normalization.

When the sentence sets for respective training speakers are different, context sets of respective speakers' data become quite different, and the intersection of context sets becomes smaller as the number of sentences decreases. Even if the same sentence set is used for all speakers, the context sets of respective speakers' data are not usually identical. Using the conventional technique, as the context sets of respective speakers' data becomes more different, the number of leaf nodes lacking one or more speakers' data increases, and quality of average voice generated from Speaker independent models tends to degrade. On the other hand, difference of gender and context sets between training speakers' data, quality of average voice generated from the shared context decision tree clustering technique models does not degrade seriously.

	mean	Std	var	n
F0	198.015	52.9787	2806.7	9039
noF0	0.79276	37.7988	1428.7	108468
MCD	1.30491	16.9808	288.35	9039
all	56.8635	12596.3	2E+08	117507

Table 5.9: Trajectory HMM with F0 normalization (Shared Decision tree Clustering)

	MCD	F0	n0F0	All
GD	5.66	32	0.2	9.04
SI	5.43	26	0.19	7.25
SAT	1.3	198	0.79	56

Table 5.10: HMM Average voice model with out F0 Normalization

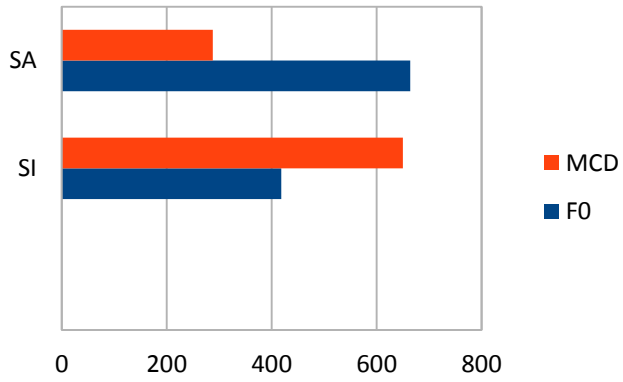


Figure 5.8 SA and SI diffèrent nimber sentences

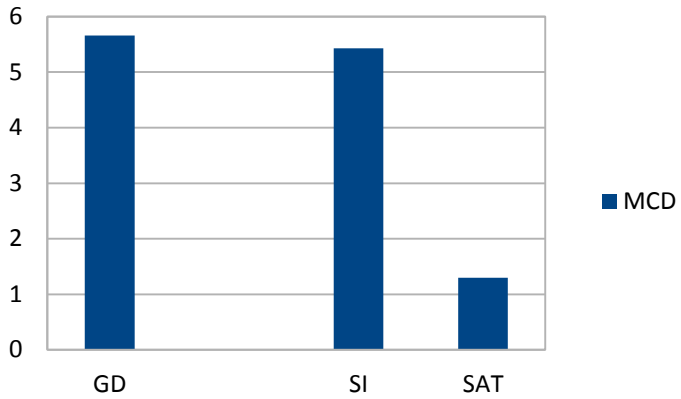


Figure 5.9: GD, SA and SI MCD scores

From figure 5.8 and 5.9, we can see that all features of synthetic speech generated from the adapted model become closer to the target speakers' features than those of average voice. The adapted model significantly outperforms the speaker dependent model especially when the available data is limited. We can also see that the adapted model gives results comparable to or a little better than the speaker dependent model even when sufficient adaptation data is available. Comparing the adaptation of F0 parameters and spectrum parameters, one sees that just a few adaptation sentences give good results in the adaptation of the F0 parameter, whereas about 47 to 120 sentences are needed to obtain good results in the adaptation of the spectral parameters. This is due to the different numbers of parameters for the features. As a result, when the available adaptation data is limited, the estimation accuracy of the spectral parameters' transformation matrix decreases compared with those of the transformation matrix for the F0 and duration parameters. On the other hand, in the adaptation of the duration parameters, the required number

of adaptation sentences varies with the target speaker. Therefore 120 sentences are utilized for the target speakers as the adaptation data for subjective evaluations.

5.5.5. Experiment #4 improvement of synthesized Amharic speech quality

This section describes improvements of Amharic HMM-based text-to-speech system. In the proposed system natural sounding speech was synthesized from the trained HMMs. However, synthesized speech has a typical quality of "vocoded speech" since the HMM-based TTS system used a traditional excitation model with either a periodic impulse train or white noise To overcome this problem, the excitation model should be replaced with more precise one (Yoshimura,2002). The mixed excitation is implemented using a multi-band mixing model, and can reduce the buzz of synthesized speech. Furthermore, aperiodic pulses and pulse dispersion filter reduce some of the harsh or tonal sound quality of synthesized speech.

In this thesis, the mixed excitation model of MELP and random forests has been applied to the proposed Amharic speech synthesis system. Mixed-excitation provides a better model for the excitation of the spectral signal. Random forests improves the speech quality by randomly varying which features to use, for spectrum and duration prediction and to sub select the set of models generated to find an almost optimal set. To do so the following processes are taken (Yoshimura, 2002).

First generate the mixed-excitation strengths and combine the extra 5 coefficients per frame to the standard combined coefficients using the festvox script "./bin/do_clustergen parallel str_sptk" and "./bin/do_clustergen parallel combine_coeffs_me" respectively.

Set the lisp variable in configuration file,"festvox/clustergen.scm (set! cg:mixed_excitation t)", to use mixed excitation.

Run the festvox script, build_cg_rfs_voice, to build a full Amharic voice (with mixed excitation, move label and random forests), as well as build flite versions of the voice.

As a result, the order of Mel-spectrum was increased to 64. the following section describe the subjective evaluation of the improved average voice model trained using mixed excitation and random forests.

	Average Voice Model (SI) as initial model			
	mean	std	var	n
F0	44.89	27.43	752	9741
noF0	0.15	0.3	0.09	233784
MCD	3.92	2.25	5.09	9741
all	10.46	880.65	775559	243525

```
RMSE 0.1143 Correlation is 0.9913 Mean (abs) Error 0.0696 (0.0907)
Dataset of 1695 vectors of 64 parameters from: festival/feats/ae_3.feats
Dataset of 1695 vectors of 64 parameters from: festival/feats/ae_3.feats
RMSE 0.1214 Correlation is 0.9897 Mean (abs) Error 0.0740 (0.0962)
Dataset of 4508 vectors of 64 parameters from: festival/feats/ae_2.feats
Dataset of 4508 vectors of 64 parameters from: festival/feats/ae_2.feats
RMSE 0.1139 Correlation is 0.9924 Mean (abs) Error 0.0713 (0.0889)
Dataset of 5462 vectors of 64 parameters from: festival/feats/ae_1.feats
Dataset of 5462 vectors of 64 parameters from: festival/feats/ae_1.feats
RMSE 0.1154 Correlation is 0.9931 Mean (abs) Error 0.0713 (0.0907)
Dataset of 12242 vectors of 64 parameters from: festival/feats/aa_3.feats
Dataset of 12242 vectors of 64 parameters from: festival/feats/aa_3.feats
RMSE 0.1122 Correlation is 0.9933 Mean (abs) Error 0.0702 (0.0875)
Dataset of 20053 vectors of 64 parameters from: festival/feats/aa_2.feats
Dataset of 20053 vectors of 64 parameters from: festival/feats/aa_2.feats
RMSE 0.1142 Correlation is 0.9913 Mean (abs) Error 0.0700 (0.0903)
Dataset of 12654 vectors of 64 parameters from: festival/feats/aa_1.feats
Dataset of 12654 vectors of 64 parameters from: festival/feats/aa_1.feats
RMSE 0.1138 Correlation is 0.9932 Mean (abs) Error 0.0709 (0.0890)
RMSE 0.1110 Correlation is 0.9936 Mean (abs) Error 0.0695 (0.0865)
Collect trees
117 unittypes as 8991 subunittypes dumped
Tree models and vector params dumped
```

5.6. Subjective evaluation

A good testing regime is one where user requirements are specified ahead of time and tests are subsequently designed to see if the system meets these requirements. Intelligibility testing is often performed with the modified rhyme test or semantically unpredictable sentences. While these give a measure of word recognition, care should be taken when using results from these tests to indicate performance in applications. naturalness is normally measured by mean opinion scores (MOS). Testing of individual components in TTS is often difficult due to there being continuous out puts, differences between objective measures and perception and components having more than one correct answer.

The only real synthesis evaluation technique is having a human listen to the result. Humans individually are not very reliably testers of systems, but humans in general are. However it is usually not feasible to have testers listen to large amounts of synthetic speech and return a general goodness score. More specific tests are required so we do subjective evaluation for the proposed Amharic speech Synthesizer with specific tests sets table 5.12 shows the list of sub test sets randomly selected among the whole test set. listening tests were conducted for evaluation of synthetic speech from the SI model using average voice techniques. Subjects were 15 (8 female

and 7 male) AAU under graduate students and Amharic speakers. For each subject, Among 72 test sets 30 test sentences ware randomly selected , which are not contained in the training data. The evaluation process is mainly based on the two concepts – naturalness, intelligibility, Naturalness has to do with the human-like sounding of the system, whereas intelligibility/ understandability has to do with the ability for one to hear the speech synthesized. Finally Subjects were presented the synthesized speech from the SI model, in random order, and each of subjects rank the synthesized speech how much is natural and Intelligible. evaluators indicate their assessments on a scale ranging from bad (1) to excellent (5). Then the average score of the opinion given will be taken as the performance of the system. As a result the speech generated from the synthesizer has 68 %, and 75% naturalness and intelligible respectively .

In addition to spectral analysis objective, we have performed a A,B opr test test for the effectiveness of a speech generated from the voice conversion models using 30 sentences for male target speaker and the original speech generated by female speakers. The result shows that we can easily vary voice characteristics by adapting HMM parameters to the target speaker. From the experimental results, 30 -40 sentences are sufficient to adapt HMMs and the speech generated from the model has 65 % naturalness.

Besides the intelligibility test, we have performed a unit test on the text normalizer. The performance of text normalizer is 90% for amharic numbers ,punctuation marks and abbreviation. The following amharic paragraph ware synthesized to test the text normalizer.

"በመጨረሻም ፍ/ቤት በሰጠው ትዕዛዝ መሰረት1. አቶ ታምራት ታረቀኝን የፍት ህሚኒስቴር ሚኒስትር ለተከታዩ ቀጠሮ ይዘው እንዲቀርቡ፣2. ሻ/ል ይልማጠቀን ፍ/ቤቱ በጠራቸው ጊዜ በሕመም ላይ ለመሆናቸው የህክምና ማስረጃ በማቅረባቸው በነፃ አሰናብቷል።3. የወረዳ10 ፖሊስ ጣቢያ አዛዥ ሁለቱን ረዳት ኮሚሽነሮች አስሮ ባለማቅረቡና ይህንን ያላደረጉበትን ምክንያት መጥተው ባለማስረዳታቸው የአ/ አ ፖሊስ አስሮ እንዲያቀርባቸው ወስኖ ለዛሬ ግንቦት 17 ቀን 2009 ዓ. ም ቀጠሮ ሰጥቷል።"

Table 5.12: test data set

	NATURALNESS	INTELLIGIBILITY
SI	3.5	3.4

Table 5.11: subjective result

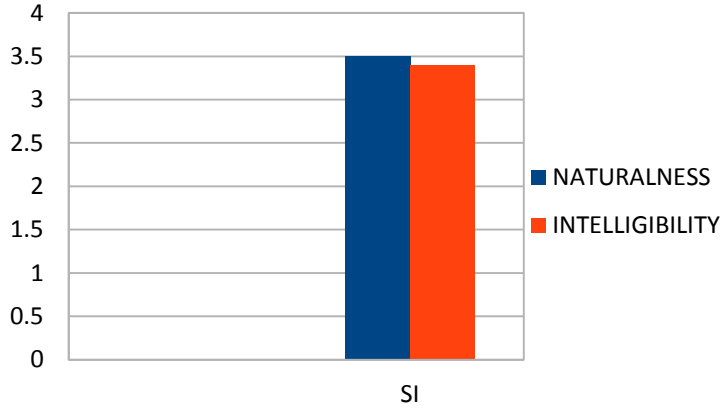


Figure 5.10: subjective evaluation result

From the figure, it can be seen that the speech generated from SAT adapted models using 120 sentences is judged to almost the same score of SI models. It has been shown that we can easily vary voice characteristics by adapting HMM parameters to the target speaker. From the experimental results, 47 -120 sentences are sufficient to adapt HMMs and the speech generated from SI has 74 %, and 70% naturalness and intelligible respectively this result shows that the system good. .

5.7. Test Data Set

No.	Test Sentence
1	የጣቢያውንም ሀላፊዎች ስንጠይቅ እኛ የእናንተ ጉዳይ አያገባንም ብለውናል.
2	ዛሬ የበላናት ቀይ ወጥ ሁርጥ ያለች ናት.
3	አቶ መለስ የ ሻእቢያ ዲፕሎማቶች ን ሰላዮች አሏቸው.
4	ጲማቸው ና ጹጉራቸው ቢያድግም ተበጥሮ ተስተካክሏል.
5	ዘነበች ሰራተኛዋን አታንጓጉ ብላ ተቆጣቻት.
6	ኤጲስ ቆጶስቱ ትላንትና ተሰብስበው ነበር.
7	ጣትሽን አንቋቁውም አታንቋቁውም ማንም አላየሽም.
8	መጋቢ ብሉይ ተክለ ማሪያም የተወለዱት ዳ ጌ ነው.
9	ሸክላ ሰሪዋ አለሚ ቱ በጉልቻ ስራ የታወቀች ነች.
10	ታፈሰ ልብሱን ሲያጥብ ቆየ.

Chapter 6

Conclusion and Recommendations

6.1. Conclusion

This research report presented Amharic TTS synthesis system that synthesizes speech that is both intelligible and fairly natural sounding. The TTS synthesis system designed was extended to include Natural language modules such as Amharic nonstandard words, Abbreviation, numbers and punctuation marks, Amharic Lexicon and letter-to-sound rules. In addition to this the Digital signal processing module also extended to include an adaptation phase using MLLR and mixed excitation model. The Amharic speech synthesis system was initially trained using data from three male and three female speakers for average voice Model (speaker independent). Both the voices from the initial (or main) baseline and the one adapted to the final system were found to exhibit intelligibility and fair naturalness. A hidden Markov model based approach was used for the development of the system. Although this method is a very flexible method that offers room for developing TTS synthesis systems with less corpus and data preparation challenges.

The TTS synthesis system developed showed an ability to synthesize understandable speech though it had no associated part-of-speech tagger, post-lexical rules and word stress information but only depended on manual lexicon entries, letter-to-sound rules simple syllabification rules, and the phone set. Several methods were applied to the system in an attempt to better the quality of the synthesized speech. The order of Mel-spectrum was increased to 64 and an attempt to use a mixed excitation model. to enhance the intelligibility of the speech in noise, and it produce the expected results as it show a significant improvement to the results already obtained from the main (or normal) TTS synthesis system. The results from the MOS were found to be 74% and 70% for intelligibility and naturalness respectively. The evaluation results show that the system built is both intelligible and fairly natural sounding. These findings are in accordance with the expectations of the system.

6.2. Recommendation

The issues listed below are some amongst many that should be looked into towards developing an even better TTS synthesis system or improving on the already built Amharic TTS synthesis system:

- ✓ Including automatic Amharic pronunciation dictionary and on the other hand, for the phoneme string to be properly uttered, part-of-speech tagger, syllabification and word stress information needs to be added.
- ✓ Incorporating intonation into the system will also undoubtedly better the quality of synthesized speech as it has been proven in several research articles.
- ✓ The tokenizer file should also be expanded to cover more Amharic numbers such as phone numbers,above ten thousand numbers and more abbreviation with their inflection.
- ✓ The statistical parametric based approach (In our case HMM-based speech synthesis) used in this research may be younger than other speech synthesis approach but not neural network-based speech synthesis method. using neural network may offers great opportunities for future research to outperform many of the other speech synthesis methods.

Reference

1. Breen. 1992. Speech Synthesis Models: A Review. Electronics & Communication Engineering Journal, February, 1992.

2. Acero. 1999. formant analysis and synthesis using hidden markov models, 1999

3. Allen,Hunnicutt and Klatt 1987. From Text to Speech: The MITalk System. Cambridge University Press, Inc. 1987.

4. Alula Tafere. 2010. A Generalized Approach to Amharic Text-To-Speech (TTS) Synthesis System. MSc Thesis, School of Information Science, Addis Ababa University, Ethiopia, July 2010.

5. Agazi Kiflu and Tibebe Beshah. 2012. Unit Selection Based Text-to-Speech Synthesizer for Tigrinya Language, HICST-December-2012-Vol 1-No 1.

6. Bahiru Demessie. 2017. Syllable-Based Amharic Speech Synthesis (TTS) Using HMM. MSc Thesis, School of Information Science, Addis Ababa University, Ethiopia, June 2017.

7. Baye Yimam. 1997. የአማርኛሰዋሰው·Amharic Grammar, Book,Addis Ababa,Ethiopia.

8. Bereket Kasaye. 2008. Developing a Speech Synthesizer for Amharic using Hidden Markov Model, MSc Thesis, School of Graduate Studies,Computer Science Department, Addis Ababa University, Ethiopia, October, 2008

9. Black, and Lenzo. 2002. Building Voices In The Festival Speech Synthesis System. http://Festvox.org/bsv. Available: http://Festvox.org/bsv/, Last Accessed: 11/06/2017

10. Black and Lenzo. 2001. Flite: a small fast run-time synthesis engine. In 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis.

11. Leggetter and Woodland. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech and Language, 9(2):171–185, 1995.

12. Cassia Valentin. 2013. Intelligibility Enhancement Of Synthetic Speech In Noise. Ph. D. Dissertation, University Of Edinburgh, Germany, 2013.

13. Cecilia caruncho. 2008. cepstral analysis synthesis on the mel frequency scale,and an adaptative algorithm for it., 2008

14. Charpentier and Stella.1986. Diphone Synthesis Using an Overlap-Add Technique for Speech Waveforms Concatenation. Proceedings of ICASSP 86 (3): 2015-2018. 1986

15. Dennis Klatt. 1987. Review of Text-to-Speech Conversion for English. Journal of Acoustical Society of America, pp. 737-793, 1987.

16. Donovan. 1996. “Trainable Speech Synthesis” PhD Dissertation, University of Cambridge, UK, 1996.

17. Ekapol Chuangsuwanich. 2016. Multilingual Techniques for Low Resource Automatic Speech Recognition.Ph. D. Dissertation, MASSACHUSETTS INSTITUTE OF TECHNOLOGYU,Department of Electrical Engineering and Computer Science, US,May 20, 2016.

18. Ethnologue. 2016. Ethnologue. 2004: Languages of the World, http://www.ethnologue .

19. Eyob Bayou .2011.Concatenative speech synthesis for amharic using unit selection method. MSc Thesis, Department of Computer Science, Addis Ababa University, Ethiopia, June, 2011

20. Getahun Amare.2001. Modern Amharic Grammer, Book, vol 9,Alpha printing press, Addis Abeba,Ethiopia (in Amharic). ጌታሁንአማረ(ረ/ ፕሮፌሰር) ፣(2001 ዓ. ም) “ ዘመናዊየአ ” ማርኛሰዋሰውበቀላልአቀራረብ፣ቁጥር9 ፣አልፋአሳታሚዎች፣ታተመ፣አዲስአበባ፣ ኢትዮጵያ፡፡

21. Heiga Zen, Keiichi Tokuda, and Black. 2009. Statistical parametric speech synthesis. Speech Commun., vol. 51, no. 11, pp. 1039–1064 , 2009.

22. Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako,Takashi Masuko, Alan W. Black and Keiichi Tokuda. 2007. The HMM-based Speech Synthesis System (HTS) Version 2.0. 6th ISCA Workshop on Speech Synthesis, Bonn, Germany, August, 2007

23. Flanagan, Allen, And Hasegawa . 2008. Speech Analysis Synthesis And Perception. 3rd Ed.

24. Junichi Yamagishi. 2006. An Introduction to HMM-Based Speech Synthesis, online availabele, www-ptex@ascii.co.jp, last Accessed Tue 17 Oct 2006 01:58:25 PM EAT, October 2006

25. Junichi Yamagishi and Takao Kobayashi. 2007. Average-Voice-Based Speech Synthesis Using Hsmm-Based Speaker Adaptation And Adaptive Training. IEICE Trans. Inf. & Syst., Vol.E90–D, No.2 ,Institute Of Electronics, Information And Communication Engineers, February 2007.

26. Junichi Yamagishi, Koji Onishi,Takashi Masuko And Takao Kobayashi (2003). Acoustic modeling of various speaking styles and emotions for HMM-based speech synthesis. EICE trans. Inf. & Syst., Vol.2–D, No.2 ,Institute Of Electronics, Information And Communication Engineers, February, 2003.

27. Shinoda and Watanabe. 2000. MDL-based context-dependent sub word modeling for speech recognition. J. Acoust. Soc. Japan (E), 21:79–86, March 2000.

28. Tokuda, Kobayashi, Masuko and Imai. 1994. Mel generalized cepstral analysis— a unified approach to speech spectral estimation. proc. ICSLP, pp. 1043–1045, Yokohama,

Japan, Sep. 1994.

29. Kuhn. 2009. Digital Signal Processing (DSP). University of Cambridge. Digital Signal Processing.

30. Laine Berhane. 1998. Text-to-Speech Synthesis of the Amharic Language. MSc Thesis, Faculty of Technology, Addis Ababa University, Ethiopia, 1998.

31. Tamura, Masuko,Tokuda, and Kobayashi. 2001. Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR. In Proc. ICASSP 2001, pages 805–808, May 2001.

32. Maël Pouget ,Thomas Hueber, and Gérard Bailly. 2015. HMM-BASED INCREMENTAL SPEECH SYNTHESIS. CNRS/GIPSA-Lab, " in Proceedings of Interspeech, Grenoble, France, 2015.

33. Marc Schröder. 2001. Emotional speech synthesis: A review, Institute of Phonetics, University of the Saar-land, Proc. EUROSPEECH 2001, vol.1, pp.561–564, DFKI, Saarbrücken, Germany, Sept. 2001.

34. Martha Tachbelie, Solomon Teferra and Laurent Besacier. 2014. Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic. Vol 56, 2014.

35. Moulines, Emerard, Larreur, Milon, Faucheur, Marty, Charpentier, Sorin. 1990. A Real-Time French Text-to-Speech System Generating High-Quality Synthetic Speech. Proceedings of ICASSP 90 (1): 309- 312.1990

36. Moulines and Charpentier. 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication, vol 9: pp 453–467, 1990

37. Mulat Shiferaw. 2012. Syllable-Based Text-To- Speech Synthesis (TTS) For Amharic" MSc Thesis, School of Information Science, Addis Ababa University, Ethiopia, June, 2012.

38. Nadew Tademe. 2008. Formant based speech synthesis for Amharic vowels. Master's thesis,Addis Ababa University, January, 2008.

39. Paul Taylor. 2007. Text-to-Speech Synthesis, University of Cambridge. Book, 2007

40. Young, Odell and Woodland. 1994. Tree-based state tying for high accuracy acoustic modeling, Proc. ARPA Human Language Technology Workshop, pp.307–312, Mar. 1994.

41. Sam Lemmetty. 1999. Review of Speech Synthesis Technology", Master's Thesis, Helsinki University of Technology, Department of Electrical and Communications Engineering, March 1999.

42. Santen, Sproat, Olive and Hirschberg. 1997. Speaking Styles: Statistical Analysis And Synthesis By A Text-To-Speech System. In Progress In Speech Synthesis,, Pp.495–510, Springer, New York, 1997.

43. Sebsibe, Kishore, Black, Kumar, and Sangal. 2005. Unit Selection Voice for Amharic using Festivox. 5th ISCA Speech Synthesis Workshop –Pittsburgh, page 103-107,2005

44. Masuko, Tokuda, Miyazaki and Kobayashi. 2000. Pitch pattern generation using multi-space probability distribution HMM. IEICE,Trans. Inf. & Syst., J83-D-II(7):1600–1609, July 2000.

45. Tadesse. 2009. Development of an Amharic Text-to-Speech System Using Cepstral Method", ICT Development Office, Addis Ababa University, Ethiopia. Computational Linguistics.

46. Tadesse, Gasser and Yoon. 2011. Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System, Ajou University, Graduate School of Information and Communication, South Korea. Technology and Communication, (May), 2-5.

47. Tadesse, Takara and Kim. 2010. Modeling Of Geminate Duration In An Amharic Text-To-Speech Synthesis System. ISCA Archive, Penang, Malaysia. Computer, 122-129.

48. Takayoshi Yoshimura .2002. Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based Text-To-Speech systems. PhD dissertation, Nagoya Institute of Technology, January ,2002.

49. Takayoshi Yoshimura, Takashi Masuko , Keiichi Tokuda, Takao Kobayashi and Tadashi Kitamura. 1999. Simultaneous Modeling Of Spectrum, Pitch And Duration In Hmm-Based Speech Synthesis. Proc. Of Eurospeech, Vol.5, Pp.2347–2350, 1999.

50. Tesfay. 2004. Diphone based TTS synthesis system for Tigrigna Language. Msc Thesis, Addis Ababa University, School of Information Science, Addis Ababa, Ethiopia.

51. Thierry Dutoit. 1993 .High Quality Text-To-Speech Synthesis of the French Language. Ph. D. dissertation,Submitted at the Faculté Polytechnique de Mons ,October 1993.

APPENDEX A: Amharic Seven Order Alphabet with IPA and Transliteration

	ā/ä [a]	u [u]	ī/i [i]	a [a]	ē/e [e/ɛ]	(i)/(ə) [ə]	0 [o/ɔ]
h [h]	ሀ ha	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ h(ə)	ሆ Ho
l [l]	ለ le	ሉ lu	ሊ li	ላ la	ሌ le	ል l(ə)	ሎ Lo
h/ḥ [h]	ሐ ha	ሑ hu	ሒ hi	ሓ ha	ሔ he	ሕ h(ə)	ሖ Ho
m [m]	መ me	ሙ mu	ሚ mi	ማ ma	ሜ me	ም m(ə)	ሞ Mo
s/ś [s]	ሠ se	ሡ su	ሢ si	ሣ sa	ሤ se	ሥ s(ə)	ሦ So
r [r]	ረ re	ሩ ru	ሪ ri	ራ ra	ሬ re	ር r(ə)	ሮ Ro
s [s]	ሰ se	ሱ su	ሲ si	ሳ sa	ሴ se	ስ s(ə)	ሶ So
sh/š [ʃ]	ሸ ʃe	ሹ ʃu	ሺ ʃi	ሻ ʃa	ሼ ʃe	ሽ ʃ(ə)	ሾ ʃo
k'/q [k']	ቀ k'e	ቁ k'u	ቂ k'i	ቃ k'a	ቄ k'e	ቅ k'(ə)	ቆ k'o
b [b]	በ be	ቡ bu	ቢ bi	ባ ba	ቤ be	ብ b(ə)	ቦ Bo
t [t]	ተ te	ቱ tu	ቲ ti	ታ ta	ቴ te	ት t(ə)	ቶ to
ch/č [ʧ]	ቸ ʧe	ቹ ʧu	ቺ ʧi	ቻ ʧa	ቼ ʧe	ች ʧ(ə)	ቾ ʧo
h/ḫ [h]	ኀ ha	ኁ hu	ኂ hi	ኃ ha	ኄ he	ኅ h(ə)	ኆ ho
n [n]	ነ ne	ኑ nu	ኒ ni	ና na	ኔ ne	ን n(ə)	ኖ no
ny/ñ [ɲ]	ኘ ɲe	ኙ ɲu	ኚ ɲi	ኛ ɲa	ኜ ɲe	ኝ ɲ(ə)	ኞ ɲo
’/ʾ [ʔ]	አ (ʔ)a	ኡ (ʔ)u	ኢ (ʔ)i	ኣ (ʔ)a	ኤ (ʔ)e	እ (ʔ)(ə)	ኦ (ʔ)o
k [k]	ከ ke	ኩ ku	ኪ ki	ካ ka	ኬ ke	ክ k(ə)	ኮ ko

	ā/ä [a]	u [u]	ī/i [i]	a [a]	ē/e [e/ɛ]	(i)/(ə) [ə]	o [o/ɔ]
h/ḵ [h]	ኸ he	ኹ hu	ኺ hi	ኻ ha	ኼ he	ኽ h(ə)	ኾ ho
w [w]	ወ we	ዉ wu	ዊ wi	ዋ wa	ዌ we	ው w(ə)	ዎ wo
‘/ʿ [ʔ]	ዐ ʔa	ዑ ʔu	ዒ ʔi	ዓ ʔa	ዔ ʔe	ዕ ʔ(ə)	ዖ ʔo
z [z]	ዘ ze	ዙ zu	ዚ zi	ዛ za	ዜ ze	ዝ z(ə)	ዞ zo
zh/ž [ʒ]	ዠ ʒe	ዡ ʒu	ዢ ʒi	ዣ ʒa	ዤ ʒe	ዥ ʒ(ə)	ዦ ʒo
y [j]	የ je	ዩ ju	ዪ ji	ያ ja	ዬ je	ይ j(ə)	ዮ jo
d [d]	ደ de	ዱ du	ዲ di	ዳ da	ዴ de	ድ d(ə)	ዶ do
j/ǧ [ʤ]	ጀ ʤe	ጁ ʤu	ጂ ʤi	ጃ ʤa	ጄ ʤe	ጅ ʤ(ə)	ጆ ʤo
g [g]	ገ ge	ጉ gu	ጊ gi	ጋ ga	ጌ ge	ግ g(ə)	ጎ go
t’/ṭ [t’]	ጠ t’e	ጡ t’u	ጢ t’i	ጣ t’a	ጤ t’e	ጥ t’(ə)	ጦ t’o
ch’/č̣ [ʧ’]	ጨ ʧ’e	ጩ ʧ’u	ጪ ʧ’i	ጫ ʧ’a	ጬ ʧ’e	ጭ ʧ’(ə)	ጮ ʧ’o
p’/p̣ [p’]	ጰ p’e	ጱ p’u	ጲ p’i	ጳ p’a	ጴ p’e	ጵ p’(ə)	ጶ p’o
ts’/ṣ [ts]	ጸ ts’e	ጹ ts’u	ጺ ts’i	ጻ ts’a	ጼ ts’e	ጽ ts’(ə)	ጾ ts’o
ts’/ṣ́ [ts]	ፀ ts’e	ፁ ts’u	ፂ ts’i	ፃ ts’a	ፄ ts’e	ፅ ts’(ə)	ፆ ts’o
f [f]	ፈ fe	ፉ fu	ፊ fi	ፋ fa	ፌ fe	ፍ f(ə)	ፎ fo
p [p]	ፐ pe	ፑ pu	ፒ pi	ፓ pa	ፔ pe	ፕ p(ə)	ፖ po
v [v]	ቨ ve	ቩ vu	ቪ vi	ቫ va	ቬ ve	ቭ v(ə)	ቮ vo

APPENDEX B EST_File utterance

DataType ascii
version 2
EST_Header_End
Features max_id 153 ; type Text ; iform "\"እሳት አደጋዎች በተጠንቀቅ ላይ ናቸው.\"" ; filename prompt-utt/mes006.utt ; fileid mes006 ;
Stream_Items
1 id _1 ; name እሳት ; whitespace "" ; prepunctuation "" ;

2 id _2 ; name አደጋዎች ; whitespace " " ; prepunctuation "" ;
3 id _3 ; name በተጠንቀቅ ; whitespace " " ; prepunctuation "" ;
4 id _4 ; name ላይ ; whitespace " " ; prepunctuation "" ;
5 id _5 ; name ናቸው; whitespace " " ; prepunctuation "" ;
6 id _6 ; name . ; whitespace " " ; prepunctuation "" ;
7 id _12 ; name . ; pbreak B ; pos punc ;
8 id _11 ; name ናቸው; pbreak B ; pos nil ;
9 id _10 ; name ላይ ; pbreak NB ; pos nil ;
10 id _9 ; name በተጠንቀቅ ; pbreak NB ; pos nil ;
11 id _8 ; name አደጋዎች ; pbreak NB ; pos nil ;
12 id _7 ; name እሳት ; pbreak NB ; pos nil ;
13 id _14 ; name syl ; stress 0 ;
14 id _21 ; name syl ; stress 0 ;
15 id _27 ; name syl ; stress 0 ;
16 id _33 ; name syl ; stress 0 ;
17 id _46 ; name syl ; stress 0 ;
18 id _51 ; name syl ; stress 0 ;
19 id _58 ; name pau ; dur_factor 1 ; end 0.125 ; source_end 0.045125 ;
20 id _150 ; name pau ; end 0.23 ;
21 id _151 ; name pau ; end 0.285 ;
22 id _152 ; name pau ; end 0.52 ;

APPENDIX C: INTERVIEW QUESTION/DATA COLLECTION INSTRUMENT

Addis Ababa University

School of Information Sciene

Users' Evaluation of Amharic Speech Synthesizer

The aim of this questionnaire is to evaluate the intelligibility and naturalness of Amharic Speech Synthesizer . All the information that you fill in this form is very critical to the conclusions we make at the end of the research work. So, I request you to answer for the questions freely and honestly.Listen to the 10 Amharic sentence and answer the following questions.your level of agreement with the following statements relating to indicators of speech Synthesis Use a scale of 1-5, where:

5= Excellent

4= Very-good

3= Good

2=Fair

1= Poor

0=bad

APPENDIX D: INTERVIEW RESPONDENTS PROFILE

Gender	Qualification	Institution
Female	Bachelor	AAU
Male	Bachelor	AAU
Female	Bachelor	AAU
Female	Bachelor	AAU
Male	Bachelor	AAU
Female	Bachelor	AAU
Male	Bachelor	AAU

Thank you

APPENDEX D F0 TREE

```
;; Auto generated list of cluster-gen f0 trees
("&_1"
 ((R:mcep_link.parent.R:segstate.parent.R:SylStructure.parent.parent.R:Word.n.gpos
   is
   0)
  ((R:mcep_link.parent.lisp_cg_duration < 0.0589997)
   ((lisp_cg_position_in_phrasep < 0.940547)
    ((11.4549 111.097))
    ((R:mcep_link.parent.lisp_cg_duration < 0.0149994)
     ((14.315 105.449))
     ((9.02371 108.542))))
   ((7.20442 105.027)))
  ((lisp_cg_state_pos is b)
   ((R:mcep_link.parent.R:segstate.parent.p.ph_cvox is +)
    ((lisp_cg_position_in_phrasep < 0.780936)
     ((R:mcep_link.parent.R:segstate.parent.p.seg_onsetcoda is coda)
      ((lisp_cg_position_in_phrasep < 0.161366)
       ((15.1726 129.972))
       ((R:mcep_link.parent.R:segstate.parent.n.ph_asp is -)
        ((lisp_cg_position_in_phrasep < 0.594828)
         ((lisp_cg_position_in_phrasep < 0.464415)
          ((lisp_cg_position_in_phrasep < 0.312672)
           ((15.7581 127.422))
           ((17.0546 125.585)))
          ((16.823 129.359)))
         ((16.9709 124.412)))
        ((17.7345 122.68))))
      ((15.6774 119.628)))
     ((17.4843 118.043)))
    ((R:mcep_link.parent.R:segstate.parent.n.ph_cvox is +)
     ((lisp_cg_position_in_phrasep < 0.675608)
      ((R:mcep_link.parent.R:segstate.parent.n.ph_ctype is n)
       ((21.0422 124.715))
       ((R:mcep_link.parent.R:segstate.parent.p.ph_ctype is s)
        ((26.3235 117.885))
        ((25.7687 120.217))))
      ((23.1471 116.216)))
```

Declaration

This thesis is my original work and has not been submitted as a partial requirement for a degree in any university. This thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A list of references is appended.

Name: Habtamu Abate Demessie

Signature: _____

Date: _____

Confirmed by advisor:

Name: Wondowssen Mulugeta (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa.