ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DEVELOPING A SPEECH SYNTHESIZER FOR AMHARIC LANGUAGE USING HIDDEN MARKOV MODEL

By:
Bereket Kasaye Tikui

A THESIS SUBMITTED TO
THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

October, 2008

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUSTE STUDIES

FACULITY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

DEVELOPING A SPEECH SYNTHESIZER FOR AMHARIC LANGUAGE USING HIDDEN MARKOV MODEL

By:

Bereket Kasaye Tikui

APPROVED BY

EXAMINING BOARD:

1 Sebsibe H/Mariam, Advisor _____

2 _____ _____

3 _____ _____

4 _____ _____

5 _____ _____

ACKNOWLEDGMENT

Many people have contributed to complete this thesis work, so I would like to thank all of them for their support. I would like to express my sincere gratitude to my advisor, Sebsibe H/Mariam for his technical support, encouragement, and guidance. Also, I would like to express my gratitude to Mr. Daniel Yacob, for his technical support.

I would like to thank Ato Nesredine Sulieman for providing me the corpus. I would also like to take this opportunity to thank all my friends for their comments on the document and their encouragement during my work.

Finally, I would sincerely like to thank my family for their support and encouragement in my study.

Table of Contents

List of tables

List of Figures

List of Appendixes

List of Acronyms

CPU – Central Processing Unit

DSP – Digital Signal Processing

F0 – Fundamental Frequency

FD-PSOLA – Frequency-Domain Pitch Synchronous Overlap Add

HMM – Hidden Markov Model

HTK – Hidden Markov Model Tool Kit

HTS – Hidden Markov Based speech synthesis tool kit

HTS-FA – Hidden Markov Model based Speech Synthesis for Amharic

IPA – International Phonetic Alphabet

LPC – Linear Predictive Coding

MLSA – Mel Log Spectral Approximation

MOS – Mean Opinion Score

NLP – Natural Language Processing

PC – Personal Computer

PDF – Probability Density Function

TD-PSOLA – Time-domain Pitch Synchronous Overlap Add

TTS – Text To Speech

Abstract

Speech synthesis systems are concerned with generating a natural sounding and intelligible speech by taking text as input. Speech Synthesizers are very important in helping impaired people, in teaching and learning process, for telecommunications and industries. Though it has many applications, generating intelligible and natural sounding synthetic speech has been a challenging task for years. To overcome these challenges, different techniques have been studied and implemented.

Though speech synthesizers based on HMM are done for foreign languages, they are not applicable for Amharic language since the languages special characteristics are not considered in these synthesizers. Hence, in this thesis work Hidden Markov Model based speech synthesis for Amharic language (HTS-FA) is done.

The HTS-FA has two phases: the training and synthesis phase. The main activities included in the training phase are preparation of the training dataset, language modeling, feature extraction and training the model. In the synthesis phase, models are selected according to the text to be synthesized, and then speech parameters are generated from them. Finally, the synthesized speech is generated from the speech parameters.

A total of five hundred sentences are used for training the model from a corpus having a size of 11,670 sentences, and twenty sentences, which are not included in the training dataset, are used for testing the performance of the system. In this thesis, the Mean Opinion Score (MOS) evaluation technique is used. The results from the MOS were found to be 4.12 and 3.6 for intelligibility and naturalness respectively for speeches synthesized by HTS-FA. Using

concatinative method the result obtained for intelligibility and naturalness are 3.54 and 3.25 respectively.

Keywords: *Speech synthesis, HMM, HMM based speech synthesis, Language Modeling*

CHAPTER ONE

INTODUCTION

1.1 General Background

The activity of getting up-to-date information is a concern to modern society, specially these days where information is considered as a day to day need. Language plays a main role to suffice this need. Language as a means of communication uses various media to facilitate information exchange. The well known means of communication using a language are spoken, written and sign. Information can be exchanged to the maximum if one uses each of them appropriately at the right time.

Most of the time, one prefers spoken form to exchange information than the other forms. This is mainly because speech carries more features in addition to what can be transferred by a written form such as melody condition and state of the speaker. Moreover, the target group will be more satisfied if the information is perceived in spoken form. This is mainly because the target may be busy to focus on the text or sign languages. Even though, there are conditions in which speech can not be involved in the process of information exchange, its role is significant in most cases. Some of the conditions where speech can not used in information exchange are when the intended target has hearing problem or the source of information has problem to generate speech that can be understood by the target or if there is no appropriate communication channel between the source and the destination.

Therefore, it will be prominent if there is a mechanism to convert information which is in text form into spoken (speech) form. One can change the text into speech by either recording the speech while reading or by using speech synthesis systems. Hence, these days two types of

speeches are there: the natural speech and the synthetic speech. Natural speech is produced by human being using the speech production organs like tongue, mouth, teeth, lips, etc. On the other hand, synthetic speech is generated by computers using speech synthesis systems. The process of generating speech from text input is called speech synthesis [5].

When speech synthesis systems are introduced, in early times, people were not happy to use them except users who have little choice as in the case with visually impaired people. The reason for this was synthetic speech was having less quality both in naturalness and intelligibility. However, speech synthesis techniques have been developed steadily in the last few decades and this has contributed a lot in increasing the quality of the synthesized speech. Because of this the application area of synthesized speeches has also increased dramatically [3, 5].

Now a days, it becomes a common custom to use synthesized speech in games to teach kids; and its role has become significant in telecommunication, and teaching and learning process. In industries, rather than using some blinking lights to give signals or messages it is becoming more appropriate to use synthesized speech to transmit full information [3]. In general, the application area of speech synthesis can be classified into two: limited domain and unlimited domain. Applications with limited domain include: emergency alarm, simple messages, announcement, warning systems, etc. Whereas, speech synthesis with unlimited domain is the concern of this thesis and its main application areas are: helping impaired people, education, and communication (email, telephones).

1.2. Application of the Study

Speech synthesis systems can be used to help visually impaired people in getting information from electronic text documents. Moreover, it is also possible to help illiterate people to get information from electronic magazines because the speech synthesizer can read it for them. Even for educated people, who are very busy to read magazines, newspapers, etc., the speech synthesizer can read it and they can hear it as a radio program while they are doing their job. Speech synthesis systems are also applicable in situations where visual attention is needed. In addition to this, they can be used in teaching and learning process.

1.3 Motivation

Computers are becoming necessity properties as they are serving as tools for communication, information retrieval, entertainment, and computing. As speech is one means of communication, so many efforts have been made to incorporate speech synthesis software into computers. Thus, different speech synthesis techniques were developed and implemented. However, speech synthesis systems that are developed for one language are not applicable for others. The reason for this is because speech synthesis systems are highly dependent on the characteristics of a language [12].

So far, concatenative and formant speech synthesis techniques have been studied and implemented by other researchers for Amharic language. Each of these techniques has got an advantage over the other. Concatenative approach has the advantage of generating intelligible and natural sounding speeches, though it requires high memory space, and processing time. Moreover, the synthesized speech exhibits discontinuity nature and is not flexible. On the other hand, formant speech synthesis system has the advantage of generating flexible,

intelligible and smooth speech. However, the synthesized speech is not natural sounding and the one that is developed so far is only meant for Amharic vowels. A speech synthesis system which can provide the benefits of these two techniques is a Hidden Markov Model based speech synthesis technique. This technique is implemented for different languages and has shown good performance in generating synthetic speech and it is efficient in resource usage like memory, storage media and CPU time [13], [16], [33]. However, this technique is not yet used for Amharic language. Therefore, it will be prominent to study and implement the technique for Amharic language.

1.4. Statement of the problem

A speech synthesis system that generates natural sounding and intelligible speech with small resource requirement is vital for many application areas. In addition to this a speech synthesizer to be useful for any one, it should generate speech with appropriate melody and prosody. However, there is no such synthesizer for Amharic yet. Therefore, speech synthesis systems that fulfill the mentioned requirements need to be developed for Amharic language.

1.5. Objective

1.5.1 General Objective

To develop unlimited domain speech synthesizer for Amharic language that can generate a natural sounding and intelligible synthetic speech with less resource requirement.

1.5.2 Specific Objectives

The specific objectives of this research work are:

- Investigation of various speech synthesis techniques

- To evaluate previously developed Amharic synthesizer
- To collect appropriate training and testing dataset
- To asses the nature of Amharic language inline with HMM based speech synthesis technique.
- To develop speech sensitizer for Amharic using HMM.
- To analyze the effectiveness of HMM based synthesizer for Amharic.
- To evaluate the performance of HMM based speech synthesis.

1.6 Scope of the Study

The HMM based speech synthesis technique has got the capability to produce a synthetic speech with different melody and prosody once the model is built and trained. However, this capability was not addressed in this research work. Moreover, in Amharic language intonation of a word changes based on the meaning of the word in that specific sentence or phrase, but the newly developed synthesizer does not address this feature and hence generates the same sound always. In addition to this, all text normalization activities are assumed to be done before the text is given to the synthesizer.

1.7. Methodologies

1.7.1 Data Collection Methodology

The main objective of the data collection methodology is to prepare appropriate training and testing dataset for the desired system. The training dataset must have proper representation of all the sound units (phonemes) in different contexts. To this end, a corpus of size 11,670

sentences is used for selecting the training and testing dataset using appropriate sampling technique.

1.7.2 Development Methodology

Existing HMM based speech synthesis system is chosen for development of Amharic speech synthesis system using HMM approach. 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 40GB of hard disk, 256 MB 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.

- Microphone for recording speech data.
- Speaker to utter speech waveforms.
- SPTK – for extracting speech parameters and to re-synthesize speech from the parameters.
- WaveSurfer – for displaying and labelling the speech waveform.
- Sox – for changing the format of the speech from one form to another.
- Festival and festivox – for generating utterance structure.

1.7.3 Testing Methods

One testing techniques was used to measure the performance of the new system taking intelligibility and naturalness as a main criteria. And that is MOS (Mean Opinion Score).

1.8 Organization of the thesis

This thesis is organized as follows. Chapter two and chapter three deals with literature review. In chapter two human speech production system, speech parameters and different techniques of speech synthesis are discussed in detail. Moreover, the chapter also overviews previous works related to this thesis work. The Amharic phonetics are discussed in chapter three that describe the characteristics and way of creation of the Amharic phonemes both consonants and vowels. In chapter four, the system architecture is depicted and description of each component of the system is given together with how each component is implemented. Results of the performance test are presented in chapter five along with the analyzed values and their interpretation. Finally, in chapter six the overall activities and achievement of this thesis work is discussed. In addition to these, the chapter also listed future works that help to improve the performance and/or to add more functionality to the developed system.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In this chapter the thesis will address human speech production system, speech parameters and different techniques of speech synthesis. With regard to human speech production, the thesis concentrates on vocal organs that participate in the production of speech[1]. There are several parameters of speech signal, some of them will be discussed here in this thesis and different speech synthesis techniques will be reviewed together with their advantage and disadvantage. In addition to this, the chapter will finally take a look at different related works, that is, speech synthesizers that have been done for Amharic language and those that have used HMM as their technique of generating synthesized speech.

2.2 Human Speech Production System

To deal with speech synthesis in detail, the research first considers some of the basic concepts related to generation of voice, articulatory organs, and parameters that are used to characterize a speech.

Voice is the sound produced by the expiration of air through vibrating vocal cords. After voice is produced, it is resonated in the chest, and vocal organs like throat, and cavities of the mouth. The quality of the voice is determined by resonance and the manner in which the vocal cords vibrate, which in turn can be characterized in terms of pitch, quality, and intensity/ loudness [4, 25].

[1] Each part corresponding to the sound they generate is discussed in chapter three

The produced voice becomes speech as it goes through the vocal organs presented in Figure 2.1 [25]. The main energy source is the lungs together with the diaphragm. When we speak, the air flow is forced through the glottis (10) between the vocal cords (12) and the larynx to the three main cavities of the vocal tract: the pharynx (9), the oral and nasal cavities (1). From the oral and nasal cavities, the air flow exits through the nose and mouth, respectively. The V-shaped opening between the vocal cords, called the glottis (10), is the most important sound source in the vocal system. The vocal cords may act in several different ways during speech. It is used to modulate the air flow by rapidly opening and closing so as to cause buzzing sound from which vowels and voiced consonants are produced [25]. With respect to this, the generated sounds can be seen as voiced and unvoiced[2] sound based on whether the vocal cord has vibrated or not when the sound is generated [25, 28].

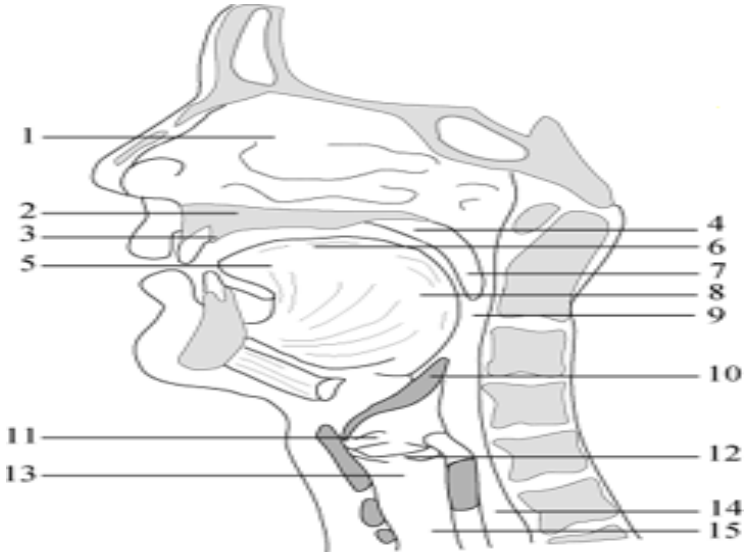

Figure 2.1: The human vocal organs[3].

[2] Voiced and unvoiced phonemes of the Amharic Language are discussed in chapter three.
[3] The label for each vocal organs: (1) Nasal cavity, (2) Hard palate, (3) Alveoral ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Glottis, (11) False vocal cords, (12) Vocal cord, *(13) Larynx, (14) Esophagus, and (15) Trachea*

As it can be understood from the above explanation, the production of speech goes through many stages and different parts of human organs cooperate to generate the speech. Hence, the characteristics of these organs need to be parameterized in the case of text-to-speech synthesis systems based on rules[4] [3].

2.3 Speech Parameters

Speech can be well characterized by its prosody, which in turn includes fundamental frequency (pitch), the duration and stress. The rate of vibration of the vocal folds is called fundamental frequency, which is denoted by F0. The term pitch is used for the rate of vibration that is perceived by the listener, and in general the pitch and fundamental frequency can be taken as same concepts [39].

Prosody is simply the way we speak out words and this depends on ones feeling at the time of speaking. For example, when someone speaks out the word "ሂዶ" the sound will be different depending on whether he/she is angry or not. This sound difference is expressed by the prosody of the speech. The emotional and intentional effects intended by the speaker or inferred by a listener are part of the message's connotation. Prosody has an important role in guiding the speaker's attitude toward the message, by expressing the connotation. In general, prosodic phenomena can be expressed in terms of pause, pitch (fundamental frequency), rate (relative duration), and loudness [3].

As we speak, we systematically vary our fundamental frequency (pitch) to express our feelings about what we are saying, or to direct the listener's attention to especially important aspects of our spoken message. On the other hand, pauses are used to indicate phrases in speech [35].

[4] Rule based speech syntheses are explained in section 2.5

Prosodic effects not always express emotions, moods and/or intentions. For example, in languages like Chinese, the prosodic effects are usually used to change the meaning of a word. However, every language, and especially English, allows some range of pitch variation that can be exploited for emotive and intentional purposes [3].

2.4 Speech Synthesis Systems

Speech synthesis, usually called Text-to-Speech synthesis (TTS), is one of the key concepts in speech processing; and it is a technique for creating speech signal from a given text in order to transmit message via voice [2,22]. Technically speaking, speech synthesis system is the process of adding speech parameters, prosodic effects, into text and then generating speech signals [3]. The TTS system consists of two main phases: The Natural Language Processing (NLP), which does the text analysis, phonetic analysis and prosodic analysis, and the Digital Signal Processing (DSP), which is responsible for speech generation [3, 39].

The conversion of text to speech is not an easy task even if one can store a huge speech data for any language. The reason for this is that TTS systems still need to deal with millions of names and acronyms [22]. So, in speech synthesis before the text is given as input for synthesis, it has to first go under some basic preprocessing steps.

The natural language processing takes the text to the appropriate form so that it can be easy for synthesis. In general, it includes text analysis, phonetic analysis, and prosody generation as it is shown in Figure 2.2 [22] with example that takes the number 2 as input. The text analysis component, usually called text normalization, takes the row text and if there are some

abbreviated words and acronyms, they will be expanded. Moreover, numeric values will be written using sequence of graphemes[5] based on their context [22,25].

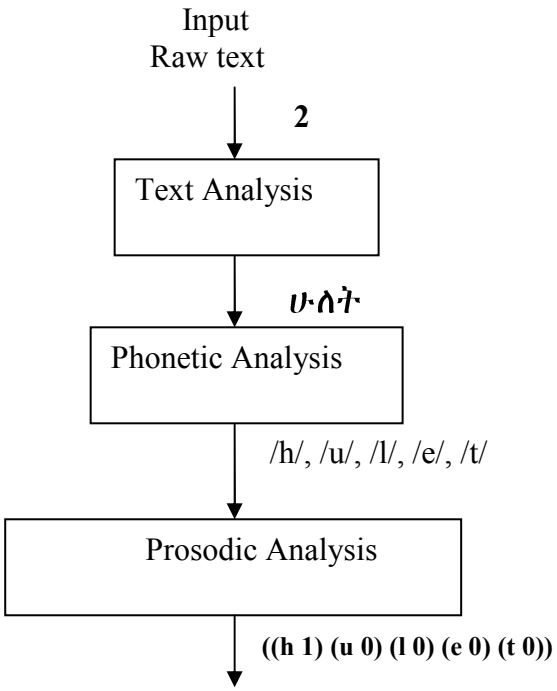

Figure 2.2: General steps of Natural Language Processing

The phonetic analysis component converts the analyzed text into the corresponding phonetic sequence. As it is shown in Figure 2.2 the word"**ሀ·ለት·**" is changed into phonetic sequence /h/, /u/, /l/, /e/, /t/. The need for phonetic analysis is in most languages the written text does not correspond to its pronunciation. So, in order to describe the correct pronunciation some kind of symbolic presentation is needed, which is the phonetic analysis. This step is also called grapheme to phoneme conversion. Phoneme is the smallest unit of speech that distinguishes meaning [4, 23, 24].

[5] Grapheme is the fundamental unit in written language, which includes alphabetic letters, punctuation marks, and all the individual symbols of any of the world's writing systems.

After the text and phonetic analysis are finished, the next step is prosodic analysis. Although the task of finding the correct prosodic features is a very challenging problem, it is a must to do it seriously to get intelligible and natural sounding synthetic speech. These features may be considered as melody, rhythm, and emphasis of the speech at the perceptual level [24]. During prosodic analysis, appropriate prosodic features will be attached to the phonetic sequence. As it is shown in Figure 2.2 during prosodic analysis, prosodic features are attached to each phoneme. Finally, the speech synthesis component takes the phonetic sequence along with the prosodic information from the fully tagged phonetic sequence to generate the corresponding speech waveform, which is accomplished during the DSP phase [23, 24].

The prosody of continuous speech depends on many factors. Figure 2.3 [25] shows the factors that affect the prosody of the synthesized text.

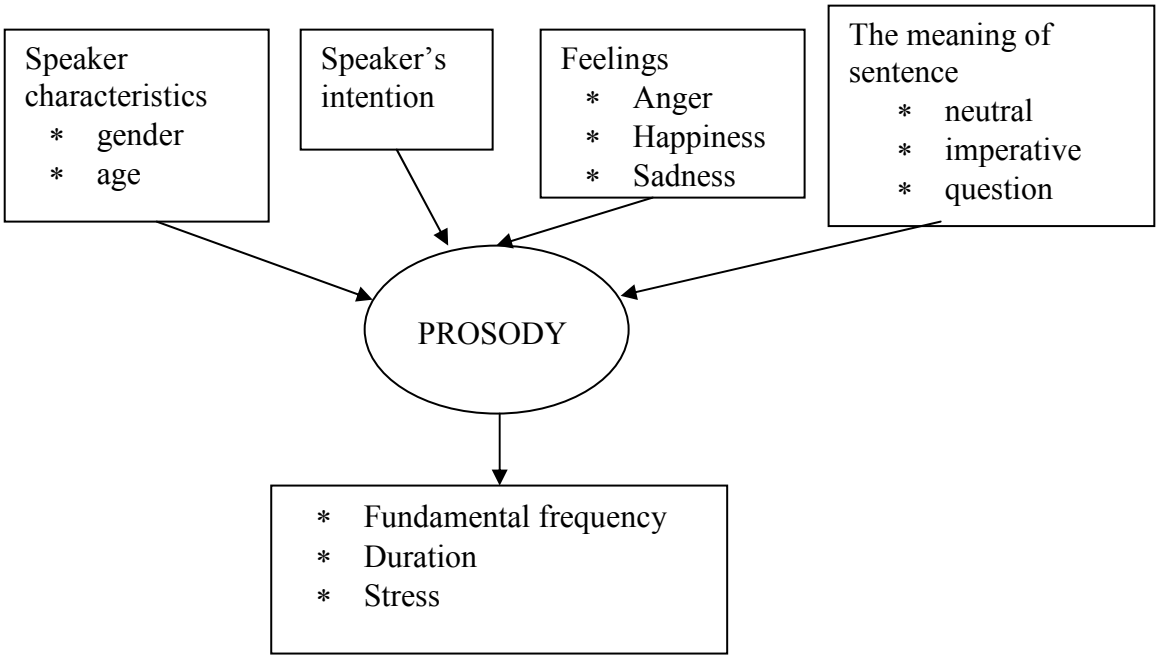

Figure 2.3: Factors that affect the prosody of a speech adapted from .

As it is shown in Figure 2.3, speaker characteristics, feelings and the meaning of the sentence has an effect on the prosody of a speech. For example, the pitch value of a woman's speech is different from the man speech. This difference is caused because of the difference in speaker's characteristics.

Besides handling the text and prosodic analysis, the simplicity and complexity of NLP depends on the nature of the language. Some languages like Italian and Spanish have very regular pronunciation; and this has a positive contribution for speech synthesis. However, languages like French have very irregular pronunciation for the same letter combination that makes the speech synthesis a bit complex [25].

In Chinese and many other Asian languages which are based on non ASCII alphabet, words are not delimited with white-space and word boundaries must therefore be reconstructed for such languages separately. However, these languages usually contain a designated symbol as sentence delimiter which makes the end-of-the-sentence detection easier, unlike English where a period is used as sentence delimiter or to mark abbreviation. In some tonal languages, such as Chinese, the intonation[6] may even be used to change the meaning of the word, which needs further analysis of the word during synthesis [23, 25].

[6] Intonation means how the pitch pattern or fundamental frequency changes during speech.

2.5 Speech Synthesis Techniques

Once the preprocessing activities are done, the next step is generating the synthesized speech by using different techniques. These techniques, in general, are classified into three categories: Rule-based, Data-driven, and Model-based methods [3, 35]. Rule-based and Model-based are also called bottom-up approaches [39].

The rule-based speech synthesis systems do not use any human speech samples but rely on rules written by linguists to generate the parameters that are required to synthesis the speech and to deal with the transition from one phoneme to another, that is, the co-articulation[7] [37].

Data-driven method uses pre-recorded speech signals for each phoneme or word, according to the chosen unit, from the database and then the selected speech waves will be concatenated and generated as a synthetic speech [25, 35].

The model-based synthesis system is a hybrid of rule-based and data-driven techniques. Such technique requires speech data to generate the parameters from which the models are built; this makes similar with data-driven techniques. To generate a synthetic speech for a given text, first the parameters will be extracted from the model and then the speech will be generated based on the extracted parameters like the rule-based techniques [35].

To mention some examples for the said categories, Articulatory and Formant based speech synthesis techniques are categorized under rule based speech synthesis technique, and HMM speech synthesis system is categorized under model-based approach while concatenative is

[7] Co-articulation is the modification in the pronunciation of a sound because of its phonetic context

data-driven. Each of these speech synthesis techniques has some benefits and limitations with regard to the quality of the speech (intelligibility and naturalness) [13].

Articulatory based speech synthesis technique attempts to parameterize the human speech production system directly, that is, it tries to model the human vocal organs as perfectly as possible in such a way that each synthetic speech will be similar to the natural speech produced by each vocal organ. This technique basically uses five articulatory parameters: area of lip opening, constriction formed by the tongue blade, opening to the nasal cavities, average glottal area, and rate of active expansion or constriction of the vocal tract [1, 26, 36].

Although Articulatory based speech synthesis is potentially the most satisfying method to produce synthetic speech with high quality, it is the most difficult method to implement, and the computational load is very high when it is compared to the other methods. Thus, it has been given less attention than other synthesis methods and has not yet achieved the same level of success as the others do [1, 36].

The formant based speech synthesis is one type of rule based speech synthesis technique that uses the source-filter model of speech description, which simulates the human speech production. In this model, the source is the air flow through the vocal cords and the filter represents the resonance at the organs of articulation: vocal cord, nasal cavity, lips, jaws, tongue, etc. [21].

Formant based speech synthesis is based on the set of rules used to determine the parameters necessary to synthesize a desired utterance. It also provides infinite number of sounds by changing the parameters which makes it more flexible than concatenative method [2].

Concatenative based speech synthesis technique uses various length of pre-recorded voices derived from natural speech. By connecting pre-recorded natural utterances, the intelligible and natural sounding synthetic speech will be produced. One of the most important activities in concatenative synthesis is to find appropriate unit length of the utterance to be stored in the database. With longer units, it provides high naturalness, less concatenation points and good control of co-articulation. However, the amount of units and memory required significantly increases as unit length increased. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex. In present systems units are usually words, syllables, phonemes, and diaphones [2,25].

The dominant data-driven speech synthesis technique is a unit-selection technique, shown in Figure 2.4 [13] Unit-selection technique is the extension of concatenative systems, and deals with the issues of how to manage large numbers of units (phoneme, diphone, etc.), how to extend prosody beyond just F0 and timing control, and how to alleviate the distortions caused by signal processing [13].

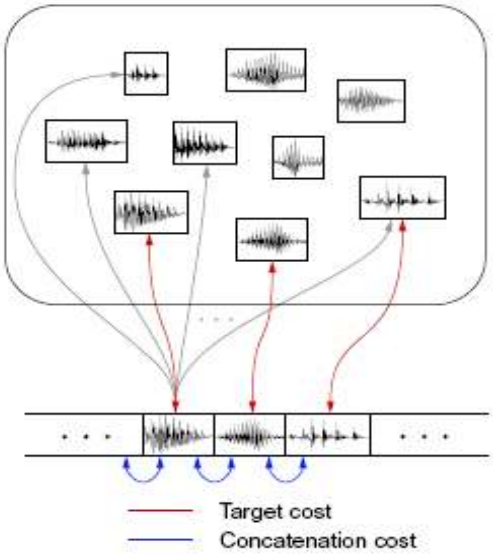

Figure 2.4: Unit-selection scheme

As it is shown in Figure 2.4, unit-selection uses pre-recorded speeches that are stored in the database for the synthesis phase. For a given input text, the appropriate speeches will be selected and concatenated to be generated as a synthesized speech.

In the past decades, until 1995, TTS systems based on speech unit-selection (waveform concatenation) techniques have been proposed and shown to be able to generate natural sounding speech, and are becoming widely used with the increasing availability of large speech databases [13, 33]. However, concatenative synthesizers are usually limited to one speaker and one voice only and usually require more memory capacity and it is not simple to make these systems have the ability of synthesizing speech with various voice characteristics, prosodic feature, speaking styles and emotional expressions. That is, one can not collect enough data to cover all the effects we wish to synthesize, and often the coverage we have in the database is very uneven. Furthermore, the concatenative approach always limits us to recreating what we have recorded; in a sense all we are doing is reordering the original data [13, 19, 20].

An alternative is to use statistical models, some times called as machine learning techniques. This and the concatenative approach can both be described as data-driven. In the concatenative approach, the data are directly used; whereas in the statistical approach we are attempting to learn the general properties of the data. Two advantages that arise from statistical models are that firstly it requires less memory to store the parameters of the model than memorize the data. Secondly, one can modify the model parameters in various ways so as to get a voice with different melody and prosody [33].

HMM-based speech synthesis is one of the most powerful statistical methods for modeling speech signal [29]. In this method, the frequency spectrum (vocal tract), fundamental

frequency (vocal source), and duration (prosody) of speech are modeled simultaneously by HMMs and also the speech waveforms are generated from HMMs themselves based on the maximum likelihood criterion [9].

Speech synthesis using HMM works by considering the internal speech production to be a sequence of hidden states, where each state models segment of a speech, and the resulting sound to be a sequence of observable states that at best approximates the (hidden) states [7, 30]. However, there are so many different combinations of hidden states that results in a given observation so selecting the one that is the appropriate one is an issue. For this reason, different algorithms have been used to select the appropriate route (combination of states) that best estimates the observed result [7, 37]. The HMM based speech synthesis technique is described in detail in section 2.7 after the general framework of the HMM is discussed in the next section.

2.6 The Hidden Markov Model

The Hidden Markov Model is one of the statistical models that are used in text-to-speech synthesis systems for modeling the features of speech signals. At each time unit, the HMM changes its states at Markov process in accordance with a state transition probability, and then generates observation data $\boldsymbol{O}$ in accordance with an output probability distribution of the current state [12, 41].

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

$$\lambda = (\mathbf{A}, \mathbf{B}, \Pi).$$

Figure 2.5[34] shows an example of a typical HMM structure, specifically a 3-state left to right model with no skip. With no skip means each nth state of the model can be reached from (n-1)th state of the model. For this particular example, transition from state 1 to 3 is not possible.

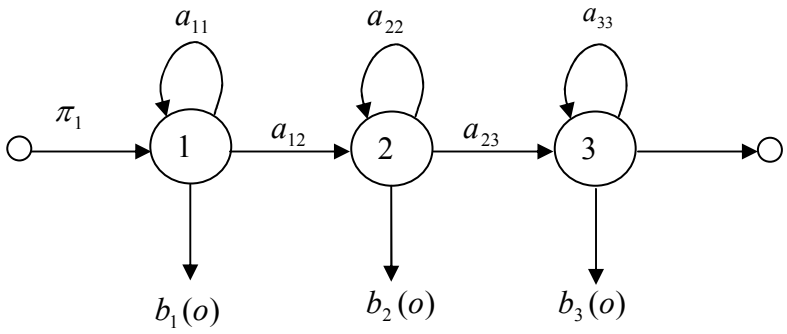

Figure 2.5: A 3-state left to right Hidden Markov Model

The output probability distribution $b_i(\boldsymbol{o})$ of the observational data $\boldsymbol{o}$ of state i can be discrete or continuous depending on the observations. In Continuous Distribution HMM (CD-HMM) for the continuous observation data, the output probability distribution is usually modeled by a mixture of multivariate Gaussian distributions as follows [34]:

$$b_i(o) = \sum_{m=1}^{M} w_{im} N(o; \mu_{im}, \sum_{im}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

where M is the number of mixture components for the distribution, and w_{im}, $\boldsymbol{\mu_{im}}$ and $\boldsymbol{\Sigma_{im}}$ are a weight, a L-dimensional mean vector, and a $L \times L$ covariance matrix of mixture component m of state i, respectively. A Gaussian distribution $N(o; \mu_{im}, \sum_{im})$ of each component is defined by [34]

$$N(o;\mu_{im},\sum_{im}) = \frac{1}{\sqrt{(2\pi)^{L} \mid \sum_{im} \mid}} \exp(-\frac{1}{2}(o-\mu_{im})^{2} \sum_{im}^{-1}(o-\mu_{im})) \quad \text{..... (2.2)}$$

Where L is the dimension of the observation data ***O***.

The need for using multiple Gaussian distribution is that, in realty a given speech signal does not exhibit a normal distribution. Hence, to increase the accuracy of modeling the speech signal a sum of Gaussian distribution is used as shown in Figure 2.6.[34]

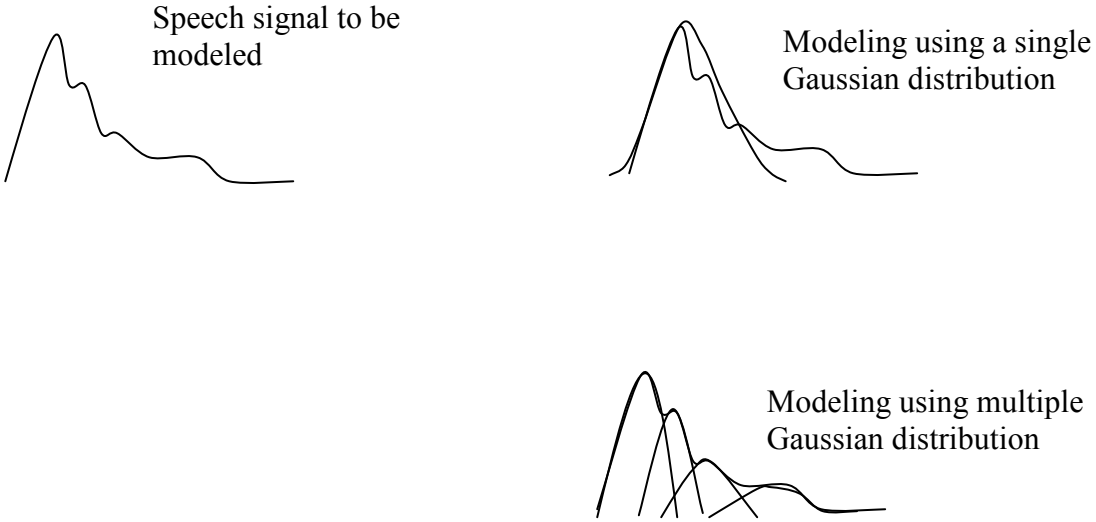

Figure 2.6: Single and Multiple Gaussian Distributions in speech signal modeling

As it is shown in Figure 2.6, when a speech signal is modeled using a single Gaussian distribution, some signals are not included to built the model. During re-synthesizing the speech from the model obviously there will be a difference with the original speech. But, when multiple Gaussian distributions are used in modeling the speech, most of the signals are included in the modeling process; as a result during re-synthesizing the speech from the model, a speech similar to the original one will be generated [34].

2.7 HMM Based Speech Synthesis Systems

When one considers different types of speech synthesis techniques which have been used so far, for example the unit-selection speech synthesis, the quality of the synthetic speech highly depends on the contents of the underlying database. With such techniques, to get a synthetic speech with high quality, either the database should be made larger, with a relative increase in the cost of database querying and storage, or a method should be found to parameterize any given inventory of speech units [16].

HMM-based speech synthesis is a synthesis technique proposed by Tokuda and Masuko in 1995 that solves the problems faced by other techniques. Unlike concatenative synthesis, where the real waveforms of the phonetic units are concatenated, in HMM-based speech synthesis, speech is generated from a concatenation of mathematical models that represent the statistics of the acoustic parameters of each phonetic unit. Moreover, unlike the acoustic models of formant synthesis, these statistical acoustic models are created automatically from real speech data. As a result, HMM-based speech synthesis has gained the advantage from both the flexibility of formant methods and the quality of concatenative synthesis [18, 33].

Hidden Markov Models have proven to be efficient parametric models for the scientific study of speech in the framework of speech synthesis [16]. It is a well-known model which can capture the dynamics of the training data by using states where each state has the capability to generate all possible observable values [13]. HMM models the probability distribution of a feature vector according to its actual state, and it also models the dynamics of vector sequences with transition probabilities between states.

One way of interpreting HMMs is to view each state as a model of a segment of speech. Figure 2.7[23] provides a diagram that shows how a speech utterance using 5-state left-to-right HMM is represented. The utterance is divided into five (number of states) segments where each segment d_i is modeled by state $i\ (S)_i$. The transition probability a_{ij} defines the probability of moving from state i to state j and satisfies $a_{ii}+a_{ij} = 1$, that means the sum of all transition probabilities from a given state should be equal to 1. Each state models the respective speech segments using M-mixture Gaussian density function $b_i(o_t)$, as given in Equations 2.1 and 2.2.

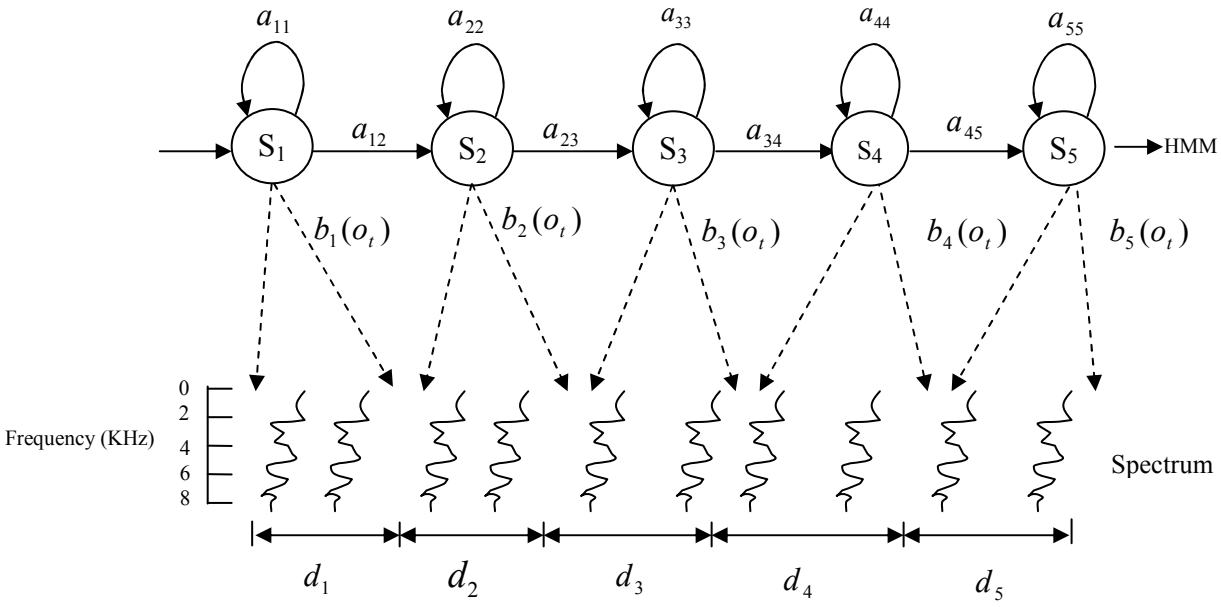

Figure 2.7: Representation of speech utterance using a five-state HMM

In the next sub sections, the chapter describes the overall activities of speech synthesis systems using HMM. The general architecture is shown in Figure 2.8.[13]

As can be seen from Figure 2.8, the HMM speech synthesis system has two major phases: the training and synthesis phases.

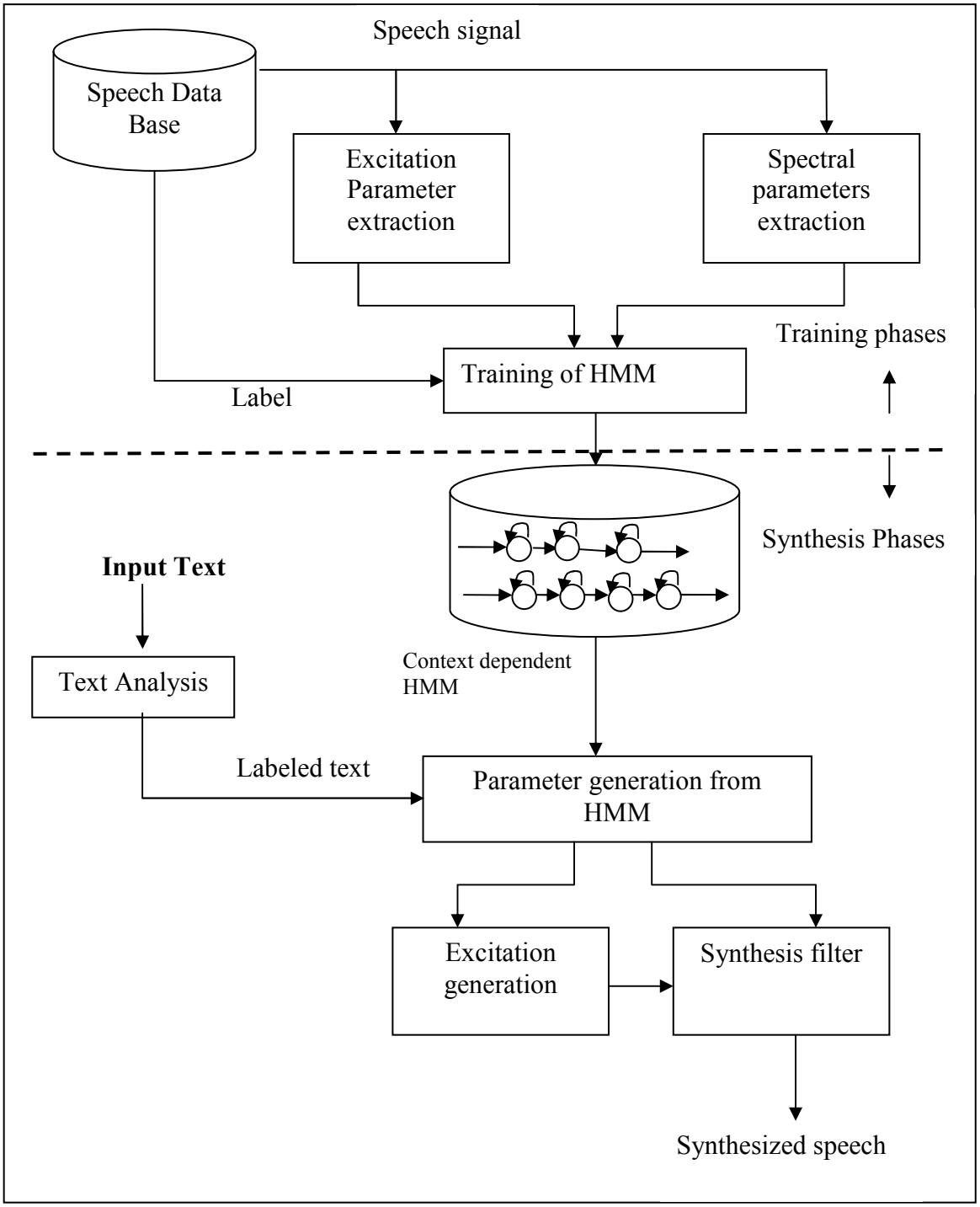

Figure 2.8: The HMM based speech synthesis system

2.7.1 Training Phase

The training phase part comprises extracting speech parameters and training the model. The speech parameters, that is the spectrum and excitation parameters are extracted from the speech signal. The spectrum parameters consist of either mel-cepstral coefficient or linear prediction coefficients (perceptual linear prediction coefficients) including the zeroth coefficients, and their delta and delta-delta coefficients. On the other hand, the excitation part consists of the logarithm of fundamental frequency (log F_0), and its delta and delta-delta coefficients [17, 18].

As far as training the model is concerned, two models are built and trained: modeling the speech parameters and the durations of each speech signal in each state. So, HMMs model the speech parameters and also have state duration densities to model the temporal structure of speech, and each state's duration is modeled using a single Gaussian distribution; hence, the HMM models spectrum parameters along with state durations and *F0* [33].

2.7.1.1 Spectrum and F0 Modeling

The features that are going to be used to train the model can be seen as four streams of data, as it is shown in Figure 2.9.[3] The first stream contains the spectrum parameter (C) along with the delta and delta – delta values. The second, third, and fourth streams contain the logarithm of the fundamental frequency and their delta and delta – delta values respectively. Each stream of information is modeled separately. The delta and delta – delta values are used to model the dynamic nature of the speech.

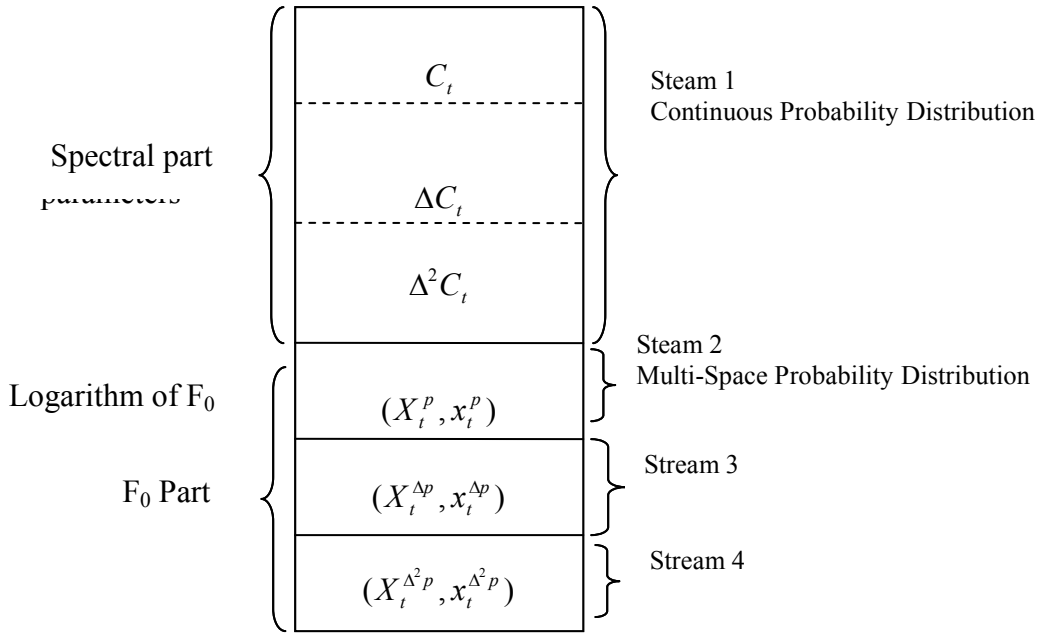

Figure 2.9: Feature Vector

The spectrum parameters are modeled by N-dimensional Gaussian distribution. Each Gaussian distribution overlaps with each other to form a multi-mixture probability density function, as it is explained in section 2.6. Moreover, the dynamic feature of the spectrum parameters, which are the delta and delta-delta values are calculated and they are also modeled in the same way as of the extracted spectrum parameters [18, 33].

Moreover, as it is explained at the end of section 2.2 of this chapter, the sound that is generated by the vocal organs is divided into two: voiced and unvoiced. Therefore, to model fundamental frequency of a speech, these two variants have to be considered. Hence, values of the fundamental frequency (F0) are composed of one-dimensional continuous and discrete values which represent voiced and unvoiced sounds respectively. This means within one stream of the feature values of the fundamental frequency, two values one for voiced and the other for unvoiced are used. Therefore, the conventional discrete or continuous HMMs can not be applied to model F0. To model such observation sequences, a new kind of HMM based on

multi-space probability distribution (MSD-HMM) has to be used. The MSD-HMM includes discrete HMM that has a single value and continuous values represented by a Gaussian distribution [6, 17, 33].

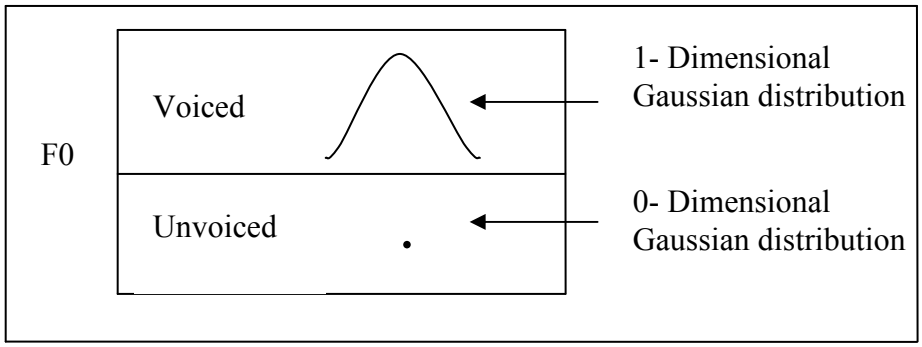

Figure 2.10: The PDF of fundamental frequency

Figure 2.10 [33]shows that the fundamental frequencies (F0) are modeled using one dimensional Gaussian distribution to model voiced sounds and a single value for unvoiced sounds. Moreover, the dynamic features that are the delta and delta-delta values of the fundamental frequencies are also modeled in the same way.

2.7.1.2 Duration modeling

State duration of each HMM is modeled by N dimensional Gaussian distribution. The dimension of state duration density of an HMM is equal to the number of states in the HMM, that is, the duration of a speech, which is the segment of the utterance, for a given state is modeled by a single Gaussian distribution. As a result of this, the *n*-th dimension of state duration densities is corresponding to the *n*-th state of HMMs [18, 33].

2.7.2. Synthesis Part

In the synthesis part of HMM based speech synthesis, first, an arbitrarily given text to be synthesized is converted to a labeled text, which is the text analysis part. Second, according to the labeled text, an HMM sentence is constructed by concatenating context dependent HMMs. State durations of the sentence HMM are determined [13, 18], and then a sequence of spectrum parameters and log *F0* values including voiced/unvoiced decisions is determined in such a way that its output probability for the HMM is maximized using the speech parameter generation algorithm [18].

2.7.2.1 Text Analysis

The text which needs to be synthesized will not be directly given to the synthesizer; rather it will go through a preprocessing activity called text analysis. In this process, the text will be changed into context based label sequences which have more information than the plain text [33].

The stream of characters in a given text must be broken into distinct meaningful units (or tokens) before any speech synthesis beyond the character level is performed [15]. If languages were perfectly punctuated, this would be a trivial issue to do, for example, if a period is always used at the end of a paragraph or a sentence. Nevertheless, real languages are not punctuated without ambiguity, and the situation is always more complicated. For example, a period might be used for abbreviating a word or showing the end of a sentence. Even in a well (but not perfectly) punctuated language like English, there are cases where the correct tokenization cannot be determined simply by knowing the classification of individual characters, and even there are cases where several distinct tokenizations are possible [35]. For example, the English

word *chap*. can be taken as either an abbreviation for the word *chapter* or as a word *chap* appearing at the end of a sentence. The same is true if we consider the abbreviation *Jan*. which can be regarded either as an abbreviation for *January* or as a sentence-final proper name. The period should be part of the word-token in the first cases but taken as a separate token of the string in the second. As another example, white-space is a fairly reliable indicator of an English token boundary, but there are some multi-component words in English that include white-space as internal characters (e.g. *to and fro*, *General Motors*). Thus, tokenization is a very important step in text analysis before the word is synthesized into wave forms. Otherwise, the synthesized speech either will be wrong or doesn't reflect the correct context of the word [35].

After tokenization is done, a decision rule, as to how the tokens are changed to words, has to be clearly defined without any ambiguity. For example, if there is a token "1988", the possible words to this token could be either "One thousand nine hundred eighty eight" when it refers to a number or "nineteen eighty eight" when it refers to a year. Therefore, such kinds of ambiguities have to be clearly identified and resolved before the speech synthesis is taken place [15, 35].

2.7.2.2 Speech Parameter Generation from HMM

Once the text analysis part is finished, the next step is to generate the parameters form the HMM that correspond to the labeled text. The parameters are extracted from the HMMs using maximum likelihood criterion [14].

After the speech parameters, that is, the spectrum parameters, fundamental frequencies, and durations are generated, the synthesized speech will be generated using a source filter model [21].

2.8 Source Filter Model

It is a model of generating speech waveform from parameters. Figure 2.11 [33] shows the source filter model. The transfer function h(n) models the structure of the vocal tract. The excitation source is chosen by a switch which controls the voices/unvoiced character of the speech. The excitation signal is modeled either a periodic pulse train for voiced speech or a random noise sequence for unvoiced speech. To produce speech signal $x(n)$, the parameters of the model must change with time. The excitation signal $e(n)$ is filtered by a time varying linear system h(n) to generate speech signal $x(n)$ [33,40].

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

$$x(n) = h(n) * e(n)$$

Where the symbol * stands for discrete convolution.

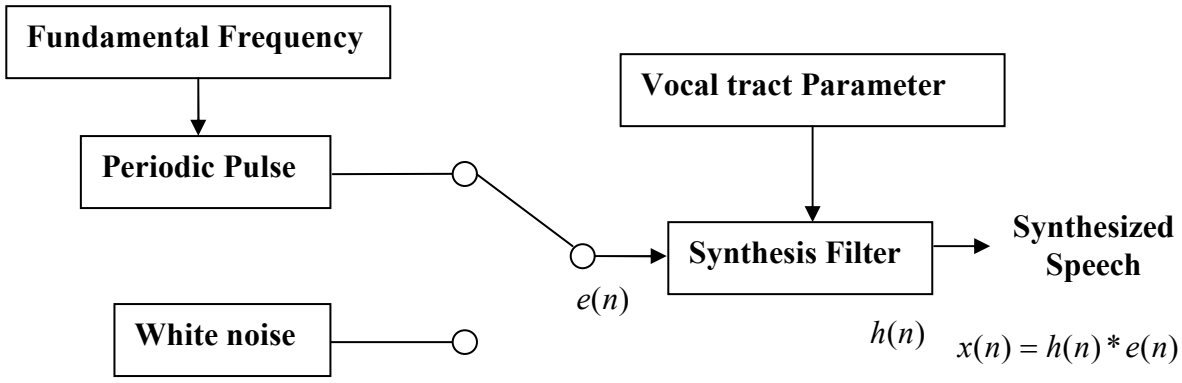

Figure 2.11: Source-filter model

2.9 Constructing Context-Dependent HMM

In continuous speech, parameter sequences of particular speech unit (e.g. phoneme) can vary according to contextual factors (phoneme identity factor, stress related factor, location factors). Hence, constructing a model for each phoneme separately is not sufficient to generate an intelligible and natural sounding speech. Therefore, to manage the variations in contextual factors appropriately, context dependent models, such as triphone models, are often employed. For example, a phoneme */p/* may have *l-p+r* model, which denotes the context-dependent version of the phone /*p/* which is to be used when the left phone is /l/ and the right phone is /r/. However, this will lead to a very large set of models, and ultimately there will be little training data for each model. To tackle this problem an approach called context clustering is used. Context clustering is a technique where contextually related models are grouped together and share model parameters. In general, two types of context clustering are used: Data driven context clustering and decision tree based context clustering [41].

Data driven clustering technique works first by building a composite model for each triphone that is included in the training dataset. Then, those phonemes that have the same contextual factor will be selected and clustered together so as to share one common model. Look at Figure 2.12[41], the phoneme /ih/ has four models for each occurrence of the phoneme in different four triphones. However, after data driven clustering is done, the first three are clustered and share one model. The phonemes are clustered together if they have the same value for the contextual factors considered. By doing so, for all phonemes included in the dataset the total number of models can be minimized.

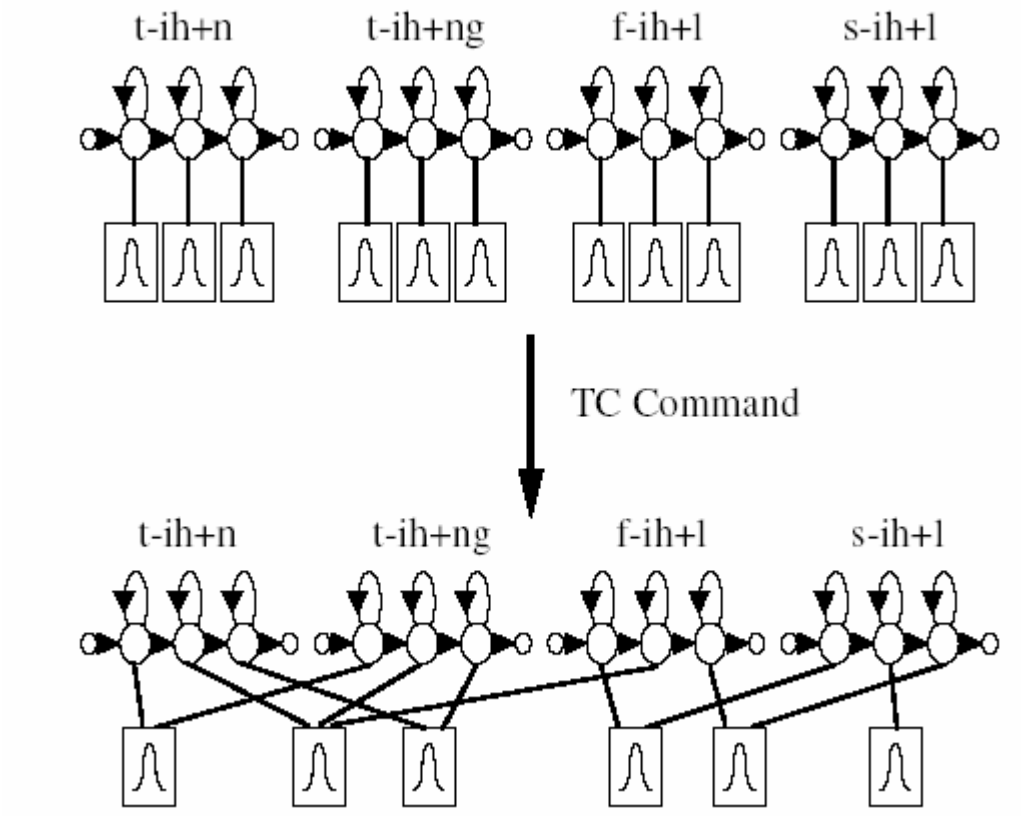

Figure 2.12: Data driven context clustering

One limitation of the data-driven clustering procedure described above is that it does not deal with triphones that are not included in the training dataset. An alternative technique which is a decision tree based context clustering mechanism provides a similar quality of clustering but offers a solution to the limitation of the data driven technique [41]. A decision tree is a binary tree in which a yes/no phonetic question is attached to each node. Initially models for a given phoneme are placed at the root node of a tree. Depending on each answer, the model changes its group and this continues until the models have filtered down to leaf-nodes. All states in the same leaf node are then clustered together. Figure 2.13[41] shows how the decision tree based context clustering works.

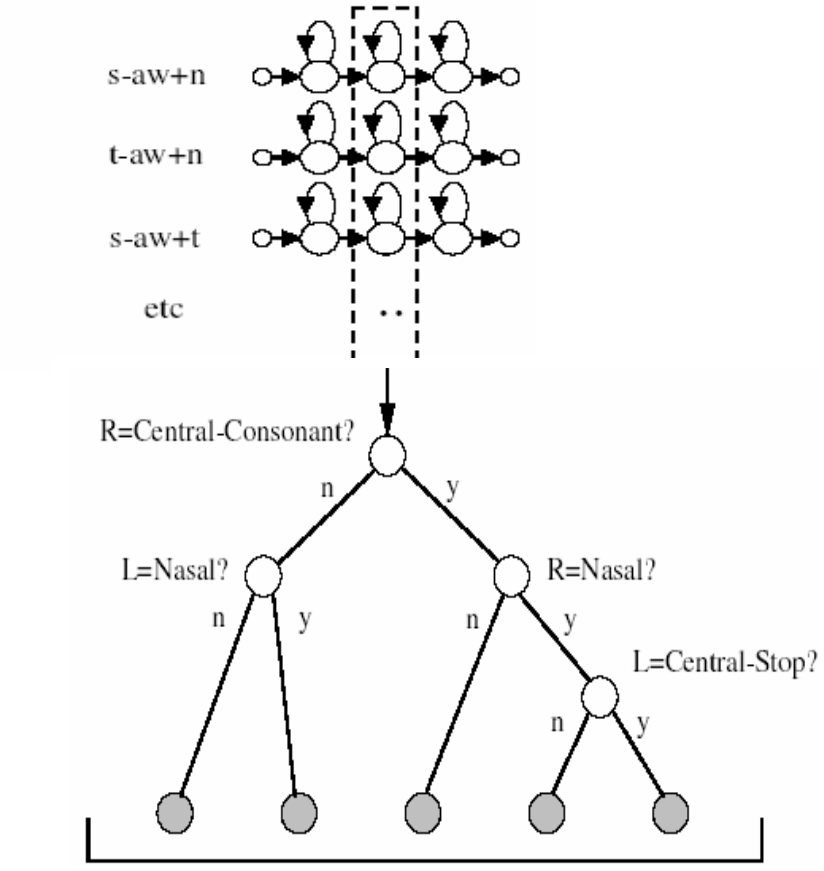

Figure 2.13: Decision tree based context clustering

In Figure 2.13, the model of s-aw+n would join the second leaf node from the right since its right context is a central consonant, and its right context is a nasal but its left context is not a central stop. Therefore, this shows that before any tree building is taken place, all possible phonetic questions must be prepared. Each question takes the form “Is the left or right context in the set P?” where the context is the model context as defined by its logical name. The set P is represented by an item list and for convenience every question is given a name. As an example, consider the following question

QS "L_Nasal" { ng-*,n-*,m-* }

The name of the question is L-Nasal, and defines the question “Is the left phoneme a nasal?” So then, if the left phoneme to the current context is in the set { ng-*,n-*,m-* }, ‘yes’ will be returned [41].

2.10 Related Works

Taking the advantage of speech synthesis systems into consideration, some interesting works have been done on speech synthesis for the local languages. Although none of the works used Hidden Markov Model (HMM) as their technique of speech synthesis, they have put a milestone to make understand the basic concepts and to introduce different techniques of speech synthesis. In the following sections, different works related to speech synthesizers for local languages and speech synthesis using HMM for any language will be discussed. Both issues are related to this research either by the language behavior or the synthesis technique employed.

2.10.1 Speech Synthesis Systems for Local Languages

Different researchers have developed a speech synthesis system for local languages. Out of such attempts, the thesis reviewed those that are built for Amharic Language by giving due attention to the technique they used and the language feature they considered.

The first work, titled “Text to speech synthesis for the Amharic Language”, regarding speech synthesizer for local language was done by Laine [8]. The technique that he used in his research was concatenative speech synthesis technique and diphones were used as the basic concatenation units. There are different methods to produce the synthesized speech using concatenation, like LPC (Linear Predictive Coding), TD-PSOLA (Time-domain Pitch Synchronous Overlap ADD), and FD-PSOLA (Frequency-Domain Pitch Synchronous Overlap ADD). Though he did not explain why he chooses it, out of these techniques, he has used the linear predictive coding method (LPC).

For his experimental proceedings he has prepared a corpus that consists of words and phrases. These words and phrases are used for extracting each diphone and these words and phrases were selected in such a way that they contain the required phoneme sequences . To minimize the discontinuous nature of the synthesized speech, he has used interpolation scheme to smoothen the boarders of successive segments. Although he did not put his experimental result quantitatively, he expressed that he had a successful experiment and had a satisfactory result .

A research work by Henok [9] deals with the same technique as of Laine, but Henok has used time-domain Pitch Synchronous Overlap and ADD (PSOLA) technique to generate the synthetic speech than Linear Predictive Methods (LPM). In addition to this, he has also considered prosodic effects, like anger, happiness and emotions, into account which Laine

didn't. Moreover, the concatinative speech synthesis scheme is also implemented by Habtamu [10] by using diphones as speech unit.

A recent work has also been done on speech synthesis for Amharic language by Nadew [11]. In contrast to the previous works which have used data-driven techniques, he has used a rule based technique called formant based speech synthesis technique that synthesizes Amharic vowels.

So far, there is no any research work that develops a speech synthesizer using HMM (Hidden Markov Model) for local languages. This thesis can be considered as a first attempt to use this technique for a local language. However, the technique is used for other languages, such as English, Germany, and Japanese.

2.10.2 Speech synthesis systems for foreign languages using HMM's

HMM speech synthesis system is applied to the German language. The research work in [16] deals with adapting HMM methods to German expressive speech particularly assessing the quality of HMM-based synthesis when it is applied to the German language. The researchers have assessed two previous works that deal with the same technique, HMM-based synthesis to German; however, they come to a conclusion that these works have used limited training sets, and have therefore reached limited intelligibility. In their work to enhance the intelligibility, they have used a speech database targeted at the development of unit-selection systems, which includes more than 3 hours of speech for each of four speakers, and there by tried to include all the German diphones.

For training purpose they have used German speech synthesis corpus . This corpus contains 1683 sentences designed to have an optimal coverage of the German diphone and the speech are taken from 2 male and 2 female speakers. They made the recordings in a sound-proof, low echo room, at a 48 KHz sampling rate/16bits resolution and they have used professional recording equipment.

Following the above work, they have used MARY speech synthesis system for extracting the context feature and for training and synthesis they have used HTS 2.0 open-source speech synthesizer tool kit after making some modifications to it. The modification includes, adaptation of the German speech synthesis corpus and Bundesliga German databases, and the use of their own context features.

Concerning their final output, they have presented results about the adaptation of the synthesis system to a very limited set of expressive football comments and they have explained that the result they got is better than any other technique to have both flexible and natural sounding speech.

The HMM speech synthesis technique is also used for English language [13]. In their work,the Authors, they have used festival for text analysis and feature extraction, like contextual factors. For training the model they have used 524 sentences and the speech signal was sampled at a rate of 16 kHz. Their model has 5 states left-to-right HMMs. The contextual factors have been considered during speech synthesis.

Though they did not put quantitative analysis, they have concluded that they have generated a natural sounding synthesized speech unlike any other rule based speech synthesizers, like formant based approach.

CHAPTER THREE

PHONETICS OF AMHARIC LANGUAGE

3.1 Introduction

Language is one of the tools for humans to pass their message to others. Each language has its own set of phonemes[8] from which sounds of words are generated. Using set of words, one can form a phrase; and a combination of phrases forms a sentence. Hence, it is possible to say that the basic unit for a language is the phoneme set and due attention should be given to phonetics. Phonetics is the study of the physical sounds of human speech and it is concerned with the physical properties of phonemes, and the processes of their physiological production [32].

Languages differ in their set of phonemes. For instance, English has 40 phonemes [43] whereas Amharic has 34 phonemes, as shown in Appendix C. Amarigna (Amharic Language) is an official and working language of Ethiopia. It is derived from Ge'ez and hence the scripts (Graphemes) are also inherited from Ge'ez. Among 73 languages which are registered in the country, Amharic is the widely spoken language. It has close to 20 million speakers worldwide, of which slightly over 17 million live in Ethiopia and the rest are in Israel and rest of the world [27, 31].

[8] Phoneme is the smallest unit of speech sounds that has a meaning.

In the next section and subsections, the thesis addresses the nature of the Amharic language's script and the characteristics and way of generation of the Amharic phonemes.

3.2 Nature of Amharic Language Script

As it is explained above, the Amharic scripts are inherited from Ge'ez script. The Ge'ez or Ethiopic script is thought to be developed from the Sabaean script and the earliest known inscriptions in the Ge'ez script dated back to the 5th century BC. At first, the script represented only consonants but later vowel indication started to appear in the 4th century AD during the reign of king Ezana [42,38].

This time orthographic representation of the language is organized into seven orders. The list of all Amharic letters is depicted in Appendix D. But, to see how the letters are organized, consider one letter /ሀ/ and it has the following different orthographic ordering.

ሀ ሁ ሂ ሃ ሄ ህ ሆ

Therefore, having these orthographic variations for each of the 33 core letters, totally the language has more than 230 orthographic symbols; Table 3.1[31] summarizes the total number of symbols used in the languages' writing system.

Table 3.1: Total number of symbols used in writing system of Amharic

No.	Type of Amharic Characters	Number of characters
1	Letters	231
2	Labialized characters	51
3	Punctuation marks	8
4	Numbers	20
	Total	**310**

In addition to the letters listed in Appendix D, which are the core characters, the language includes some additional letters that do not have any ordering. Such characters are referred as labialized characters, in Table 3.1, and some of such letters are listed in Appendix E. Moreover, the language has also its own punctuation marks and numberings [31]. The punctuation marks and the numbers are listed in Tables 3.2 and 3.3. [31]

Table 3.2: Amharic Numbers and their equivalent Latin numerals

1	2	3	4	5	6	7	8	9	10
20	30	40	50	60	70	80	90	100	1000

Table 3.3: Amharic Punctuation Marks

word separator	Period	comma	colon	Semi-colon	Preface colon	Question mark	Exclamation mark

3.3 Amharic Phoneme set

As it is explained in chapter two, the air generated from the lung is resonated with different articulatory organs and as a result different phonemes are generated [28]. Based on which articulatory organs are used and the way they behave, phonemes are divided into two: consonants and vowels. It is indicated in [27] that the Amharic language has 34 phonemes. These phonemes are all distinct since there is a difference in the speed and way of air flow within the articulatory organs to generate each of them. Basically, each phoneme differs in

three different aspects: vibration level, place of articulation, and manner of articulation [27]. In the following sections the thesis deals with Amharic consonants and vowels.

3.3.1 Amharic Consonants

In Amharic language there are 27 consonants out of the total 34 phonemes. As it is explained above, each of them mainly differs in the manner of articulation, place of articulation, and vibration level. In the next discussion each issue is explained with some examples.

When the air pressure generated by the lung passes through nearly closed vocal cords, vibration is created since the air vibrates the vocal folds. Therefore, vibration level is one way of classifying consonants. Consider two sounds /ስ/ and /ዝ/, the /ዝ/ sound creates more vibration than the sound /ስ/. One can easily see the difference between these two consonants by filling the vibration created on throat. Thus, these two sounds differ mainly in their level of vibration. Consonants that are blocked by the vocal cord while they are generated are called voiced sounds ("ንዛሪ ድምጽ") and others which don't are voiceless sounds ("ኢ-ንዛሪ ድምጽ"). Based on this, Amharic consonants are divided into two: voiced sound (ንዛሪ ድምጽ) and voiceless sound (ኢ-ንዛሪ ድምጽ).

After the air passes the vocal cord, it has two options. One is to go through the nasal cavity and the other is to go through the mouth and this is determined by velum. As the air passes through either the mouth or the nose, the sounds which will be generated are different. In general, consonants that are generated by nasal cavity are called nasal sounds (የስርጉ ድምፆች). Whereas, those that are created by mouth cavity are called oral sounds (የአፍ ድምፆች). These differences in consonants occur because of the difference in place of articulation. The place of articulation (also point of articulation) of a consonant is the point of

contact, where an obstruction occurs in the vocal tract between an active (moving) articulator (typically some part of the tongue) and a passive (stationary) articulator (typically some part of the roof of the mouth) [28]. Even for nasal sounds the active and passive articulator are needed. The other way to see the difference in consonants is to consider how the air generated from the lung behaves based on the way active articulators behave. This is referred as manner of articulation.

In general, the Amharic consonants based on the way they are generated are classified and depicted in Table 3.4 [27, 28]. As can be seen from Table 3.4, there are six types of consonants: stops, fricatives, nasals, affricates, semivowels and liquids, which are classified based on manner of articulations. Nasals, liquids and semivowels are always voiced; stops, fricatives and affricates can be voiced or unvoiced. They can also be classified as labials, alveolar, palatals, velars, labio – velar and glottals based on place of articulation.

Table 3.4: Consonants with their feature

Manner of Articulation	**Voicing**	**Place of articulation**											
		Labials (ከናፍራዊ)		Alveolar (ድጻዊ)		Palatals (ላንቃዊ)		Velars (ትናጋዊ)		Labio-velar		Glottals (ማንቁርታዊ)	
Stops (እግድ)	Voiceless	p	ፕ	t	ት			k	ክ	kx	ኪ		
	Voiced	b	ብ	d	ድ			g	ግ	gx	ጊ		
	Glottalized	px	ጵ	tx	ጥ			q	ቅ	qx	ቂ		
Fricatives (ሹልክልክ)	Voiceless	f	ፍ	s	ስ	sx	ሽ					h	ሀ
	Voiced	v	ቭ	z	ዝ	zx	ዥ						
	Glottalized			xx	ጽ							hx	ኀ
Africatives (ፍትግ)	Voiceless					c	ች						
	Voiced					j	ጅ						
	Glottalized					cx	ቈ						
Nasals (ሰርናዊ)	Voiced	m	ም	n	ን	nx	ኝ						
Liquids (ጉናዋ/ላሽ)	Voiced			l	ል								
				r	ር								
semi-vowels (ክፊል አናባቢ)	Voiced	w	ው			y	ይ						

Stop consonants are generated when the air is blocked and immediately released by articulatory organs. Hence, the stop consonants are distinguished by which articulatory organ the air is blocked, that is, whether the air is blocked by lips, tongue and velars, or tongue and alveolar. From this perspective when we see the sounds listed in Table 3.4, / ፕ፣ ብ፣ ጵ፣ ት፣ ድ፣ ጥ፣ ክ፣ ግ፣ and /ቅ/ are stop consonants. /ፕ፣ ብ፣ ጵ፣ / are created when the air is blocked by lips thus they are labials, and / ት፣ ድ፣ ጥ/ are created when the air is blocked by alveolar. In short, they are made by placing the tongue against the alveolar ridge, the hard

ridge in the top of the mouth, and behind your teeth. /ክ፣ግ፣ቅ/ are created by velars; in this case the back of the tongue stops the air at the back of the hard palate [28].

Out of these stop consonant phonemes, /ጵ/ and /ፐ/ are not Amharic phonemes, rather they are borrowed from other languages like, Greek and Latin. These phonemes came with words like /ጳጳስ፣ ጠረጴዛ፣ ፖሊስ/ which means Pope, table and police respectively [28].

Fricatives involve letting the air slide through a narrow opening in the mouth. They can be prolonged for some time and the air is not completely blocked. For instance, in the case of consonants like ፍ, ስ, and ሽ, the air is neither blocked nor free while they are generated.

Nasal sounds like /ም፣ን፣ኝ/, can not be generated if the air is blocked from passing through the nasal cavity. Hence, there should be an air flow through the nasal cavity to produce them. The air will be directed to the nose since different parts of the mouth blocks it from passing through the mouth. Therefore, the difference between these sounds is that, in the case of /ም/, the air is blocked by lips, for the sound /ን/ the air is blocked by alveolar and tip of tongue; whereas the sound /ኝ/ is created when the air is blocked by palatals and middle part of the tongue.

The affricates begin as stops and slide into fricatives, and hence are represented as a stop followed by a fricative. For example, the phoneme ች needs two things to be generated. First the air has to be blocked by palatals and middle part of the tongue and then it will be released to pass through the sides of the mouth.

3.3.2 Amharic Vowels

Phonemes have been classified as vowels and consonants. In the previous section we have seen the set of consonants and their way of generation in detail. This section will concentrate on the characteristics and way of generation of Amharic vowels.

One of the basic differences between consonants and vowels is that in the case of vowels, as it is explained in [28], the air generated by the lung will vibrate the vocal cord since the vocal cord is slightly closed when vowels are generated. Moreover, in generating different vowels the tongue plays an important role. The contribution of tongue can be seen in two different aspects. The first one is the way the tongue is positioned in terms of height and the second one is the movement of tongue to the front and back of the mouth. In addition to tongue, the shape of lips has also a contribution in changing the vowels sound when it varies from rounded to non-rounded (flat) [28].

In Amharic there are seven vowels, these are ኧ, ኡ, ኢ, ኣ, ኤ, እ, and ኦ. All are voiced and oral sounds. These vowels can be found in each letter, that is, each letter in Amharic is not a single sound rather it is a combination of two sounds, one from vowel and one from consonant. Look at the following example given in Table 3.5. With the seven orders of "በ", each of the vowels ኧ, ኡ, ኢ, ኣ, ኤ, እ, and ኦ also occurs. Therefore, the sounds of letters of the Amharic languages are a combination of a consonant and vowel [28].

Table 3.5: Amharic vowels along with seven orders of a consonant.

Be	Bu	Bi	Ba	Bie	B	Bo
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ብ	ብኡ	ብኢ	ብአ	ብኤ	ብእ	ብኦ

Amharic vowels are divided into two based on the shape of the lip as rounded and non-rounded. When one generates the sounds of ኢትዮጵያ and ሃዱ, the sound /ኢ/ in ኢትዮጵያ needs the lips to be flat and some what spread. Whereas, the sound ኡ in ሃዱ does need the lips to be rounded in order to be generated. In general, vowels ኡ and ኦ are classified as rounded. Whereas, ኧ, ኢ, አ, ኤ, and እ are classified as non-rounded as far as the lip shape is concerned [11, 28].

The Amharic vowels, with respect to the movement and position of the tongue, exhibit different characteristics. The vowels /ኢ/, /እ/ and /ኡ/ are created when the tongue moves to the roof of the mouth. The sound /አ/ is created when the tongue takes the lower part of the mouth. Whereas the sounds /ኧ/, /ኤ/, and /ኦ/ are created when the tongue takes the middle position, that is when it is not to high toos or too low. Hence, based on the height of the tongue vowels are divided into high, middle, and low.

The tongue also moves to the front and back of the mouth while we speak vowels. When it does this movement, it creates different vowels at different positions. In this perspective, /ኤ/, and /ኢ/ are front, /እ/, /ኧ/, and /አ/ are middle and /ኡ/, /ኦ/ are at back. Figure 3.1[27, 28] summarizes all combination of positions and movement of tongue along with all possible vowels that can be created.

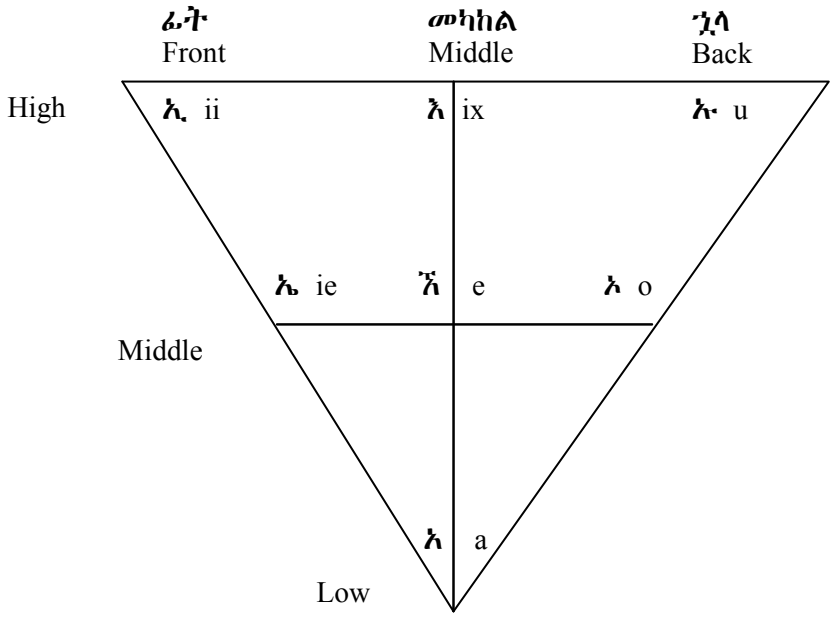

Figure 3.1: IPA maps of the Amharic Vowels

CHAPTER FOUR

SYSTEM ARCHITECTURE AND IMPLEMENTATION OF THE SYNTHESIZER

4.1 Introduction

Systems usually have a number of components that interact to each other in a logical fashion. The processing logic of a component may generate an output that may be the input to another component, may also trigger another component to activate itself or may affect the processing logic of another component. Hence, in such cases it will be very important to clearly put the system architecture that shows the whole process of the system and the interaction among components.

Speech synthesis systems incorporate two basic components: the Natural Language Processing (NLP) and Digital Signal Processing (DSP). As it is explained in section 2.4 of chapter two, the natural language processing deals with the pre-processing activities that includes text analysis and language modeling (phonetic and prosodic analysis). Digital Signal Processing is responsible for speech generation from text input.

Speech synthesizers for different languages mainly differ because of the processing logic of the NLP component. Two languages are different phonetically, morphologically, syntactically, and semantically that leads to different ways of doing the natural language processing component of the speech synthesizer. To this end, the processing logic of the Amharic speech synthesizer using HMM is similar to other language's speech synthesizers that uses HMM

technique with regard to DSP. However, there is a difference as far as the NLP component is considered. Hence, the HMM based speech synthesis for Amharic will address the text, phonetic, and prosody analysis inline with the language's characteristics.

In the following sections, the paper has presented the system architecture of the Amharic speech synthesizer, detailed activities of the subsystems of the synthesizer and how each component is implemented in this thesis work.

4.2 System Architecture

Figure 4.1 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.

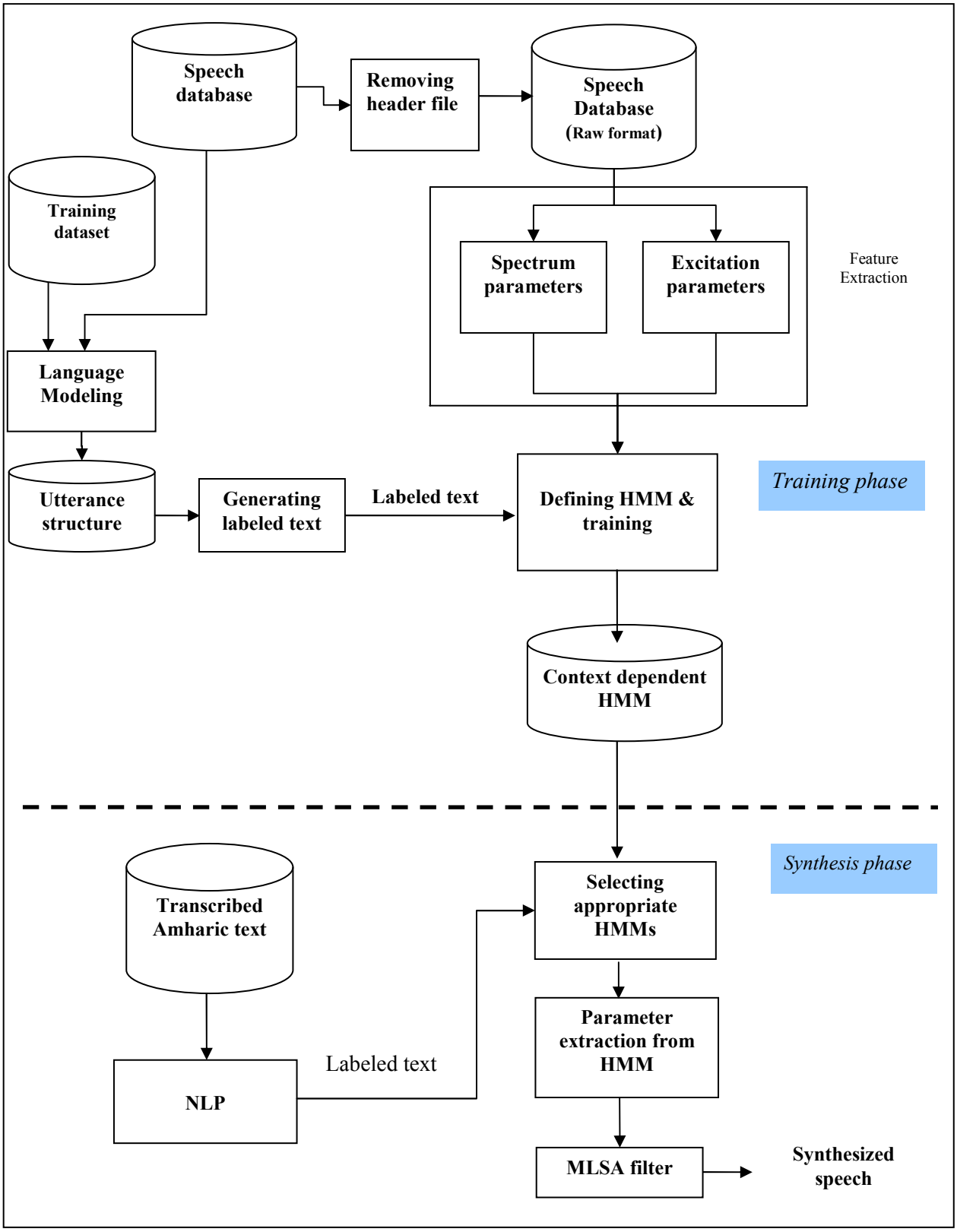

Figure 4.1: System Architecture of the Amharic speech Synthesizer using HMM.

4.3 Language Modeling

Out of the total system architecture, the language modeling component is responsible in adding phonetic and prosodic information into the text that will be used for training the model. As it is shown in Figure 4.2, this component includes activities like, transcription of the original text, labeling the utterance, and generating the utterance structure. In the next subsections each activity of the language modeling is discussed in detail.

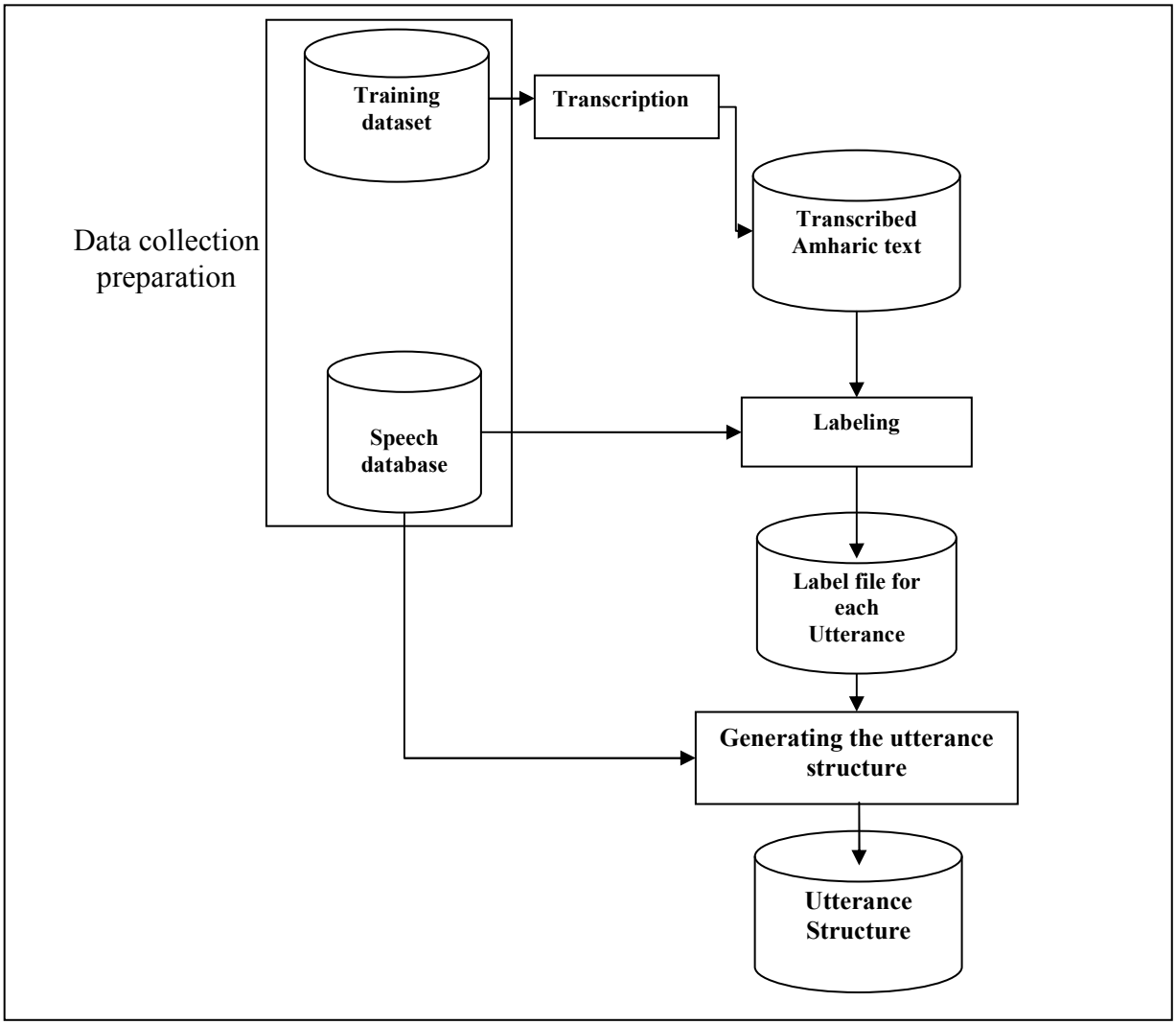

Figure 4.2: Language modeling architecture

4.3.1 Data collection and preparation

In this thesis work a non-random sampling called purposive sampling is used to prepare the training dataset. Using this technique, five hundred sentences were collected from a corpus that has 11,670 sentences. Purposive sampling is used to select the sentences for two reasons. The first reason is that when the sentences are selected from the data source randomly, some rare phonemes, like ቨ and ዠ, may not be included in the sentences since phonemes are not distributed uniformly in a data source [38]. The second reason is that HMM based speech synthesis technique requires each phoneme to be included in the training dataset so many times, with different contexts, so as to model each phoneme as accurate as possible. Table 4.1 shows the distribution of all phonemes that are included in the training dataset.

The total number of phonemes included in the training dataset is 16,178. As the phoneme distribution shows, in Table 4.1, 50 % of the total phonemes included in the training dataset are covered by only seven phonemes out of 34. This clearly shows that, even with non-random sampling it is hard to include all phonemes equally in the training dataset. However, with non-random sampling all phonemes are guaranteed to be included in the training dataset.

Table 4.1: The distribution of the Amharic Phonemes in the training dataset.

Amharic Phonemes	frequency of occurrence
ቭ	19
ፕ	23
ሯ	26
ሸ	33
ቷ	36
ሗ	56
ጸ	60
ጇ	62
ኝ	75
ፍ	161
ዝ	190
ኤ	205
ች	218
ጥ	246
ቅ	268
ክ	292
ሀ	294
ኦ	341
ድ	378
ግ	413
ኢ	449
ኡ	483
ስ	495
ር	541
ወ	588
ብ	659
ት	760
ይ	766
ል	790
ም	848
እ	848
ን	898
አ	1818
ኧ	2839
ድምር	16,178

After the data is collected using the Amharic script, the next step is transcription. In this thesis work the transcription was done manually. Finally, the speech was recorded using a PC with sampling rate of 16 KHz, sample size of 16-bit and encoding format of PCM waveform. By doing so the speech database is built. Moreover, the recorded utterance is changed into raw data format for training purpose using Sox-14.0.1 tool. Following the preparation of raw data format of the recorded utterance and after the utterance structure is generated, defining and training the HMM is proceeded.

4.3.2 Labeling the Utterance

It is shown in Figure 4.2 that, to generate the utterance structure the speech data and labeled utterances are needed. From the activities of data collection and preparation, the speech data and transcribed texts are already available. Hence, the labeled utterance is left to generate the utterance structure. The process that generates the labeled utterance is labeling. Labeling is the process of giving a label for each speech signal in the utterance.

In this thesis work, labeling is done manually using a tool called WaveSurfer. Although, there are different tools which are developed for automatic labeling, these tools are error prone since they automatically segment the speech and label it based on some prior information about the acoustic data. In addition to this, labeling the utterance manually has additional advantage to make the transcribed text and uttered speech the same, that is, sometimes for a given text the recorded sound may have additional phonemes which are not included in the original text. For instance, the sentence "በጣም አመሰግናለሁ" is transcribed as "betxam amesegnalew" as per the rule of IPA given in Appendix C. However, when the sound is recorded, additional phoneme እ is included after the phoneme ግ. This means the transcribed text for the word

“በጣም አመሰግናለዉ” should be corrected as “betxam amesegixnalew” by adding phoneme /ix/ after phoneme /g/. Hence, in this thesis work such cases are seriously addressed and corrected accordingly since they have an effect during generating the utterance structure and ultimately the model. Figure 4.3 shows the speech wave form along with the spectrogram and labels for the text “betxam amesegixnalew”.

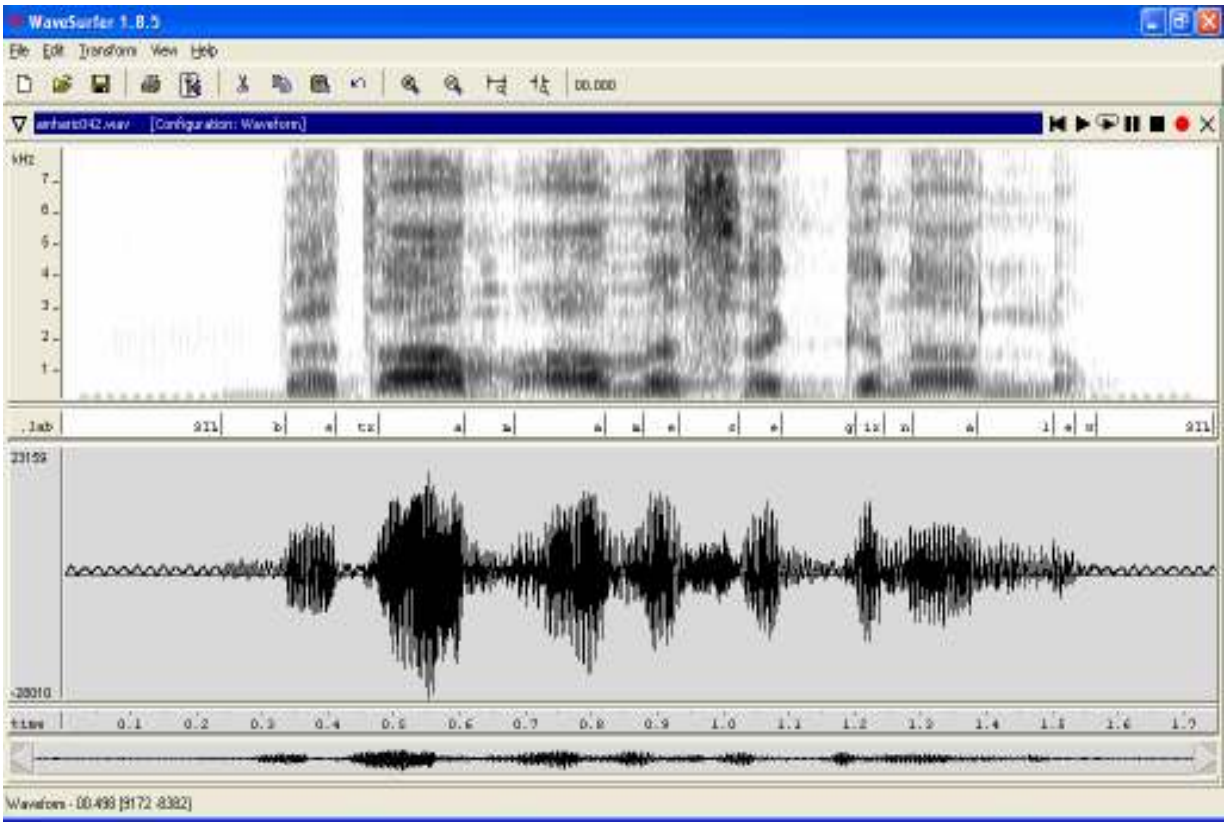

Figure 4.3: A speech wave form corresponding to its spectrogram and labels.

The labels shown in Figure 4.3 will be saved in a file that contains the time slice of each phoneme from the total utterance time. This file is called label file. As it is shown below the phoneme /tx/ is a speech signal uttered from 0.4126311 second to 0.4759736 second of the signal and the same is true for all phonemes.

```
0.0000000 0.2388917 SIL
0.2388917 0.3348103 b
0.3348103 0.4126311 e
0.4126311 0.4759736 tx
0.4759736 0.6080879 a
0.6080879 0.6822891 m
0.6822891 0.8198328 a
0.8198328 0.8795557 m
0.8795557 0.9320395 e
0.9320395 1.0243385 s
1.0243385 1.0876810 e
1.0876810 1.2016975 g
1.2016975 1.2433226 ix
1.2433226 1.2867574 n
1.2867574 1.3844858 a
1.3844858 1.5003121 l
1.5003121 1.5346980 e
1.5346980 1.5672742 w
1.5672742 1.7446331 SIL
```

4.3.3 Generating the Utterance Structure

A file that contains the full transcribed text along with phoneme list of the transcribed text and prosodic information is called utterance structure. The utterance structure also consists of a set of relations over a set of items. Each item represents an object such as a word, segment, syllable, phonemes, etc. Whereas, relations show how these items relate to each, for example, the relation between two phonemes in that specific utterance [35].

The label files generated using WaveSurfer were not directly used to generate the utterance structure since there is a difference in data format needed by the tools festival and festvox, which are used to generate the utterance structure. Hence, in this thesis the format of the label file generated by WaveSurfer is changed so that it will be used as input for the tools. Moreover, these tools were not directly used since the tools were developed as a full fledge

speech synthesizer for other languages. Hence, language specific knowledge such as phoneme list, label file, text data and wave files of Amharic must be provided. In addition to this, there is a program called ***parser*** in these tools and its role is to generate a phoneme sequence, syllable structure and stress information from a given word. Hence, this program should be provided for festival to handle Amharic words. Moreover, these tools demands each phoneme to be characterized with the features given in Table 4.2.

The Amharic phonemes set that are included in festival and festvox, corresponding to their characterizing features, are shown in Appendix F. Each phoneme has eight features that describe how the vocal organs behave when the sound is uttered. For example, if we take the sound "*l"*/ል/ it has -, 0, 0, 0, +, l, a, and + values for the features described in Table 4.2. These listed eight values are described on the eight rows of Table 4.2 respectively. The zero value is used when the feature is not applicable for a particular phoneme.

Table 4.2: Features of phonemes.

features	possible values	description
vowel or consonant	- and +	**-** for consonant and + for vowels
vowel length	s l d a 0	**s**-short **l**-long **d**-dipthong **a**-schwa
vowel height	1 2 3 0	**1**-high **2**-mid **3**-low
vowel front-ness	1 2 3 0	**1**-front **2**-mid **3**-back
lip rounding	+ - 0	**+** for rounded and **–** for non-rounded
consonant type	s f a n l 0	**s**-stop **f**-fricative **a**-affricative **n**-nasal **l**-liquid
place of articulation	l a p b v 0	**l**-labial **a**-alveolar **p**-palatal **b**-labio-velar **v**-velar
consonant voicing	+ - 0	+ for voiced and – for non-voiced

After incorporating this information into these tools, the utterance structure was generated for each recorded speech. The whole content of an utterance structure is given in Appendix G for the transcribed text "betxam amesegixnalew."

4.4 Generating Labeled Text

As it is shown in the system architecture, Figure 4.1, to train the model the labeled texts are needed. This label text can be generated from the utterance structure or from the transcribed text. In this research the labeled text are generated using the festival tool from the utterance structure. The labeled text for the transcribed text "`betxam amesegixnalew`", which is extracted from the utterance structure, is given in Appendix H.

Considering one line of the labeled text given in Appendix H, let us see the information that labeled text carries for each phoneme included in the transcribed text.

```
b^e-tx+a=m@1_1/A:0_0_1/B:0-0-1@3-3&3-15#1-2$1-2!2-3;2-3|novowel
/C:0+0+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=
2@1=1|NONE/I:0=0/J:17+2-1
```

Each line of the labeled text considers five sequences of phonemes from the transcribed text. In the above labeled text the five phoneme sequence are b, e. tx, a, and m. the third phoneme is considered as the current phoneme, in this case the phoneme "tx" is the current phoneme. And the others are referred as L, LL, R and RR respectively. L means left of the current phoneme, LL means left of the left phoneme to the current phoneme. The same is true for R and RR, but this time to the right direction. And the values given after each of the letters A to J are the phonetic and prosodic information. For example, the three values given after the letter A are whether the previous phoneme is stressed (1) or not (0), whether the previous phoneme is accented (1) or not (0), and the number of phonemes in the previous syllable respectively. The values given after the letter J are the number of phonemes included in the transcribed text, the number of word in the transcribed text, and the number of phrases respectively.

4.5 Feature Extraction

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-cepstrum or linear prediction coefficient. On the other hand, the excitation parameter consists of the fundamental frequency (*F0*). In this thesis work mel-cepstrum coefficients are used as spectrum parameters.

Using the raw data generated during data preparation, these speech parameters (features) are extracted using the tool SPTK-3.4[9] and then the delta and delta-delta values of the mel-cepstrum coefficients are calculated as follows.

$$\Delta c_t = \frac{c_{t+1} - c_{t-1}}{2} \quad \text{..... 5.1}$$

$$\Delta^2 c_t = \frac{1}{4} c_{t-1} - \frac{1}{2} c_t + \frac{1}{4} c_{t+1} \quad \text{..... 5.2}$$

Where c is mel-cepstrum coefficients

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

$$\Delta\, p_t = \frac{p_{t+1} - p_{t-1}}{2} \quad \text{..... ..5.3}$$

$$\Delta^2 p_t = \frac{1}{4} p_{t-1} - \frac{1}{2} p_t + \frac{1}{4} p_{t+1} \quad \text{..... 5.4}$$

Where p_t is the logarithm of the F0.

[9] Shot description of tools used in this thesis work is given in appendix J.

4.6 Defining 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 number of states and whether to include dynamic feature of the mel-cepstrum coefficients and fundament frequencies in modeling the speech signal. In this thesis work, after making a comparison between different models having different number of states, a model that has seven states per phoneme is chosen. In addition to this, the 25 mel-cepstrum coefficients along with their delta and delta-delta values and pitch feature vector consisted of logarithm of F0 and its dynamic values (delta and delta-delta) are used. Therefore, the HMM is defined in such a way that feature vectors consisting of mel-cepstrum coefficients, F0 and their dynamic features will be included during modeling. To incorporate all features, four streams of data were used within each state; where stream one is used for mel-cepstrum coefficients including their delta and delta-delta values and the other three streams were used to model the F0 and their delta and delta values.

As it is shown in Appendix A, the topology of the HMM used in this thesis work has 7 states left to right HMMs with no skip. In the HMM definition, the transition matrix of the prototype specifies both the allowed transitions and their initial probabilities. The allowed transitions will have non-zero probability values. Thus, the HMM definition used in this thesis work only allows to make a transition from left to right.

It is explained above that each state consists of four streams where the first stream contains 75 pair of mean and variance which models the 25 spectrum parameters and their delta and delta-delta value. The second, third and fourth streams contains two pair of mean and variance to

model the voiced and unvoiced value of the fundamental frequencies, the delta, and delta-delta values respectively. Hence, the transition probabilities of the model are a seven by seven matrix.

To model the duration of each speech segment a single dimensional Gaussian distribution is used; that means five pairs of mean and variance are used for each emitting states of the HMM.

4.7 Training the Model

Training the model means estimating the HMMs parameters, which are the mean, the variance, and the transition probabilities based on the utterance structure and the extracted parameters (features) [41]. Once the parameters are extracted, training of HMMs is performed with the Hidden Markov Model Toolkit, which is software that provides a set of library modules and tools for building and manipulating HMMs. HTK supplies four basic tools for HMM parameter estimation: HCompV, HInit, HRest and HERest [41]. In addition to these tools one additional HTK tool is used for context analysis, which is HHEd.

HCompV and HInit are used for initialization of the model parameters. HCompV is used to set the mean and variance of every Gaussian component in a HMM definition. This tool is typically used for initializing the model if the speech utterance is not labeled. Alternatively, a more detailed initialization is possible using HInit which computes the parameters of a new HMM. In this research work HInit is used to initialize the model since the speech utterance is labeled. HRest and HERest are used to refine the parameters of existing HMMs using Baum-Welch Re-estimation (forward/backward) algorithm [41]. In Appendices A and B the HMM parameters before training and after training are given.

As it is explained in section 4.6, the defined HMM for this thesis work incorporates both mel-cepstrum coefficients and F0. Therefore, to model a segment of speech signal a pair of mel-cepstrum and F0 values together with their delta and delta-delta values are needed. Hence, the mel-cepstrum and F0 values are combined together, so that they will be used to model the speech segment. The over all procedure of training the HHM is given in Figure 4.4

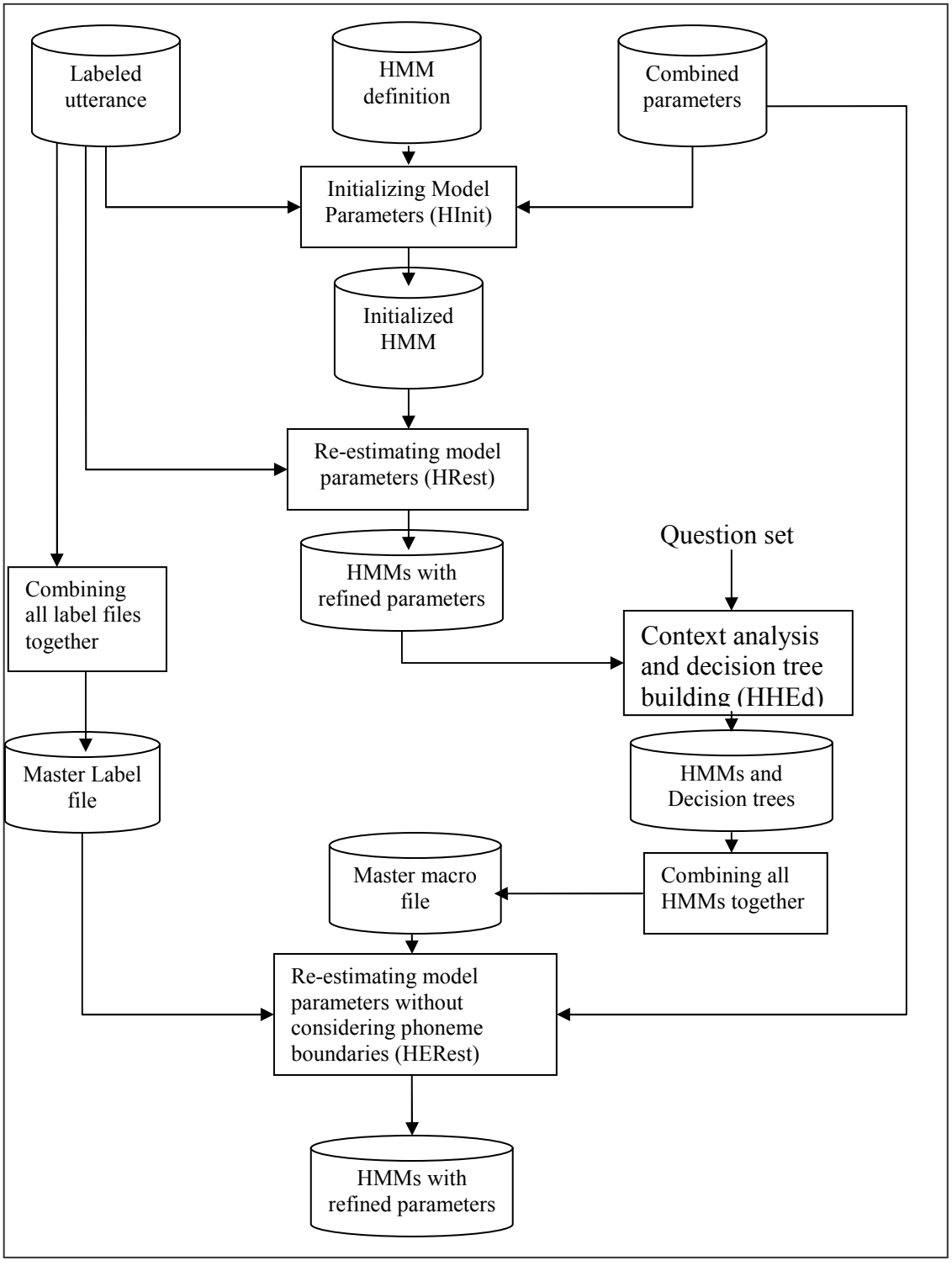

Figure 4.4: The procedures to generate HMMs for each phoneme.

In the following paragraphs each step depicted in Figure 4.4 is described. As it is shown in Figure 4.4, the first step in generating HMM parameters is to provide initial estimates for the parameters of every HMM model of the phoneme set. This process takes as input HMM definition, combined parameters, and label for each phoneme and initializes them one by one.

Initializing the HMM parameters requires some data (combined parameters). To circumvent this problem, the speech signal of each phoneme will be uniformly segmented and each segment will be used to initialize the parameters of each state of the model. Such initialization only makes sense if the HMM is left-to-right. In this work HInit is used for model initialization process. Each operations of HInit are given in Figure 4.5.[41]

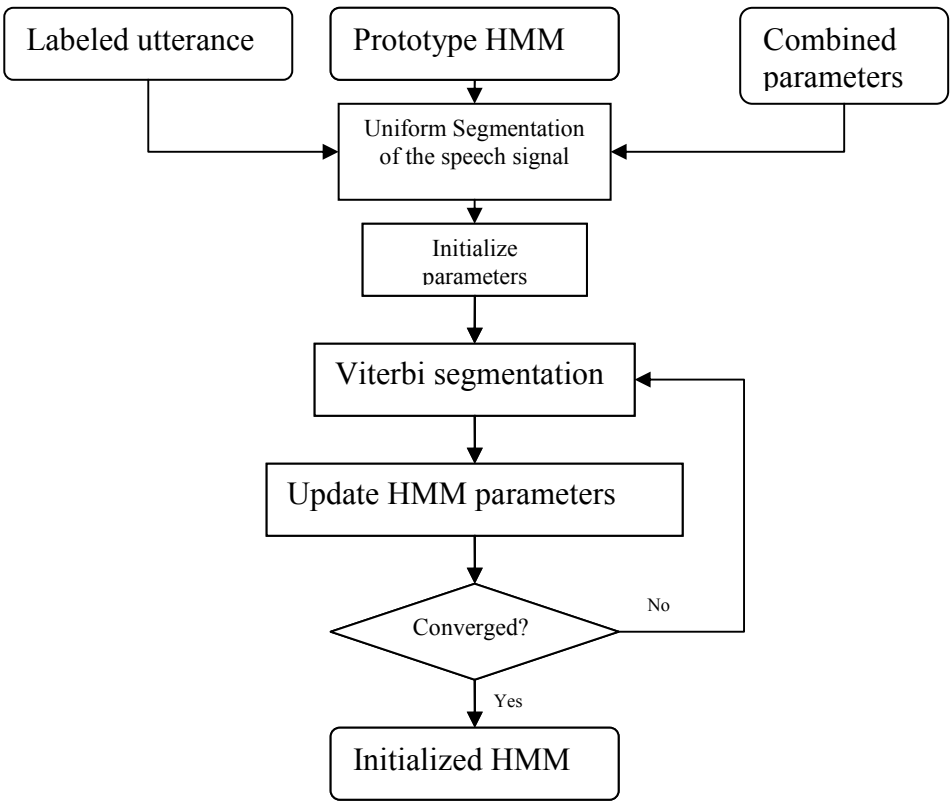

Figure 4.5: HInit operation Adapted from

The initialized model and the combined parameters selected from all utterances are used to refine the model parameters for a given phoneme using forward/ backward algorithm. The labeled utterance is used to select the phoneme boundary from all utterances and to collect the parameters that only belong to a phoneme. For refining the model parameters of each phoneme HRest tool is used. The operations of HRest are given in Figure 4.6.[41]

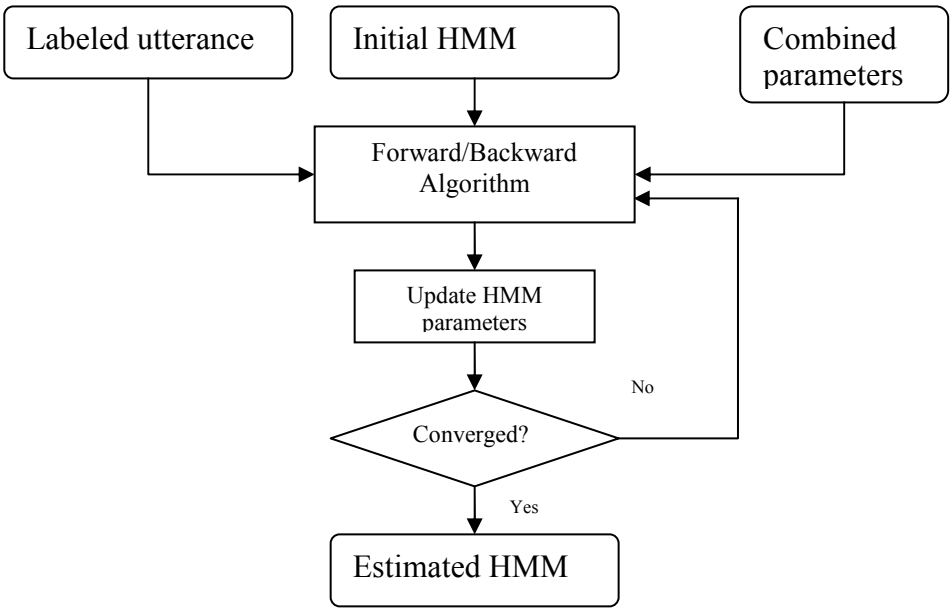

Figure 4.6: HRest operation Adapted from

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

phoneme boundaries so as to include contextual factors. That is, unlike the processes described so far, this process simultaneously updates all of the HMMs of each phoneme in a given utterance. For this process HERest tool is used.

HERest processes each training file in turn and the whole utterance is considered rather than a single phoneme. It uses the associated transcription to construct a composite HMM which spans the whole utterance. This composite HMM is made by concatenating instances of the phoneme HMMs corresponding to each label in the transcription. When all of the training files have been processed, the new parameter estimates are formed and the updated HMM set is generated as an output, Figure 4.7 shows the operations included in HERest [41].

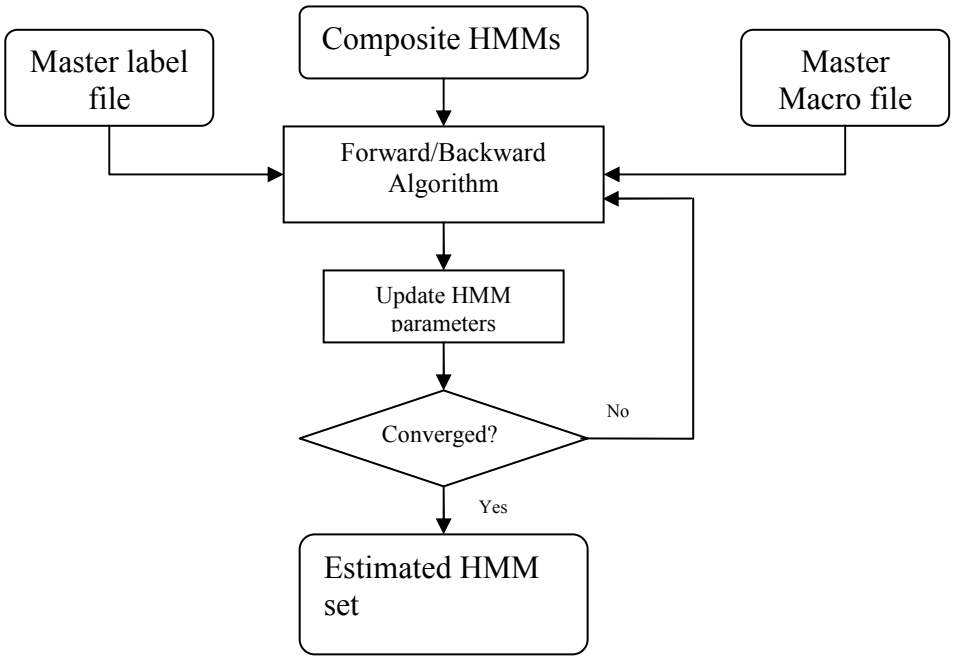

Figure 4.7: HERest operation Adapted from

The contextual factors that are considered in this thesis are phoneme identity factor, stress-related factor, and location factors. Phoneme identity factor checks whether a given phone is vowel or consonant. Stress related factor is a factor that determines whether a given phoneme

is stressed or not. Location factors deals with the location (position) of a given phoneme in a given word.

Considering the above contextual factor obviously enables to obtain appropriate models. However, as contextual factors increase, their combination also increases exponentially, which increases the searching and computation cost. As it is explained in section 2.9, to overcome this problem there are two methods: Data driven context clustering and Decision tree based context clustering. In this thesis work decision tree based context clustering is used by using the HTK tool HEEd. To do so, a question set for Amharic Language is prepared, which is needed to build the decision tree. As it is shown in Appendix K, a question set is a file that consists of a set of questions. Considering the fist line of the question set,

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

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

4.8 Synthesis Phase

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

The texts that are going to be synthesized will not be given directly to the synthesizer. Rather the text will undergo through different pre-processing activities. These activities include the conversion of numerical values into their corresponding orthographic representation,and the expansion of abbreviated words and acronyms into full text. In this thesis work these activities are done manually. However, a program is written using Java to generate the labeled text; and then appropriate models were selected corresponding to each phonemes of the labeled text. From the selected models speech parameters, spectral and excitation parameters were generated. To select the appropriate models and to extract speech parameters from the models, HMGenS was used, a speech parameter generating tool, which is provided by HTS-2.0.1 open source tool kit. Finally, speech waveform is synthesized directly from the generated parameters by mel log spectral approximation filter using the MLSAF tool which is one of the SPTK tools.

CHAPTER FIVE

EVALUATION OF THE NEW SYSTEM

5.1 Methods

In this research work one technique were used to evaluate the system. That is, mean opinion score (MOS). MOS is an evaluation technique where evaluators indicate their assessments on a scale of bad (1) to excellent (5). Then the average of the opinion will be taken as the performance of the system. The performance of the new system is also measured in relation to the unit selection technique (concatenate). The reason that unit-selection technique is chosen is because it is a well known speech synthesis technique in generating an intelligible and natural sounding speech, regardless of high memory usage, discontinuity nature and high computation effort due to large run-time data [13]. To have the same ground for the comparison, the five hundred sentences used for training HMM were also used as speech database for unit-selection technique. In both cases a questionnaire was used to collect the evaluator's opinion.

5.2 Preparing a Questionnaire

To proceed with the evaluation, a questionnaire is designed that comprises questions that focus on the intelligibility and naturalness of the synthesized speech. The questionnaire is given in Appendix I. The first question is targeting in measuring the intelligibility of the synthesized speech and the second question is aimed at measuring whether the synthesized speech is human like or not. For both testing methods the same questions are used.

5.3 Test Data Preparation

For the purpose of evaluating the performance of the system, twenty sentences were collected randomly from the corpus with manual intervention. The manual intervention is needed to avoid semantically predictive sentences. For instance, if a sentence like "**የኢትዮጵያ ኤሌትሪክ ኃይል ኮርፖሬሽን ዋና ስራ አስኪያጅ ማብራሪያ ሰጡ**" is included in the test data, then the respondents can easily predict what the next word will be after hearing some of the previous words. Hence, with such sentences, the respondents will not give due attention to some part of the synthesized speech to understand what it says. This will eventually affect the evaluation process. Therefore, in this thesis work a maximum effort is done not to include semantically predictable sentences in the test dataset. Out of the total test data, twenty sentences were used to evaluate the system using the technique of MOS. The sentences are synthesized using the two techniques, which are, using HTS-FA and concatnative method.

All the preprocessing activities discussed in chapter four of section 4.7 are done before the text is given to the synthesizer. These preprocessing activities are done manually; that includes, changing numerical values into their corresponding pronunciations, expanding abbreviated words, and transcription.

5.4 Evaluation Results and Analysis

As it is explained in the above discussion Mean Opinion Score (MOS) techniques were used to evaluate the quality of the speech synthesis systems. The MOS is designed to measure the naturalness and intelligibility of the synthesized speech. The MOS scales the quality of speech into 5 levels as shown bellow in the Table 5.1.

Table 5.1 Mean Opinion Score level

MOS	Quality	Impairment
5	Excellent	Imperceptible (No difference with the natural speech)
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

A total of twenty five native speaker of the language were selected to evaluate the system. Nine of them are female and sixteen of them are male. The evaluators are asked to assign the MOS value to each of the twenty sentences synthesized using HMM based approach and data driven approach (concatenative method). They are asked to measure both intelligibility and naturalness of the synthesized speech.

Table5.2 and Table5.3 show analysis of the HMM based speech synthesizer both in its intelligibility and naturalness. Table 5.2 is for female and Table5.3 is for male evaluators. As it can be seen from the result, the synthesizer provides good quality speech in terms of its intelligibility and near good in terms of its naturalness. Moreover, the variability of the evaluators is minimal compared to the data driven approach.

Table5.2 Female HTS Analysis

	Naturalness	Intelligibility
Average	3.78	4.10
Standard Deviation	0.27	0.2

Table5.3 Male HTS Analysis

	Naturalness	Intelligibility
Average	3.43	4.13
Standard Deviation	0.24	0.25

Table5.4 and Table5.5 show the analysis of the data driven speech synthesizer evaluated by female and male evaluators respectively. The analysis result shows near good quality in terms of intelligibility and fair quality in its naturalness. We can also observe high degree of

variation among the evaluators during evaluating naturalness and intelligibility of data driven speech synthesizer.

Table5.4. Female Data driven Analysis

	Naturalness	Intelligibility
Average	3.32	3.53
Standard Deviation	0.54	0.49

Table5.5 Male Data Driven Analysis

	Naturalness	Intelligibility
Average	3.17	3.54
Standard Deviation	0.54	0.51

Table 5.6 and Table 5.7 show the analysis of both the HMM based approach and data driven approach speech synthesis system obtained from combining the results taken from both the female and male evaluators. The over all result shows that both synthesizers provide high quality speech in terms of intelligibility than naturalness. Moreover, variability of measuring quality of HMM based speech synthesis system is very less compared with the data driven approach. In addition to these, the HMM based speech synthesis system provide better quality speech waveform for Amharic than the data driven speech synthesizer both in its naturalness and intelligibility. Moreover, when we analyze the value of the standard deviation, table 5.6 and 5.7, for HTS-FA is smaller then that of data driven approach. This means that the performance of HTS-FA is some what consistent from sentence to sentence than that of data driven method.

Table5.6 HTS Analysis (both female and male combined)

	Naturalness	Intelligibility
Average	3.6	4.12
Standard Deviation	0.21	0.16

Table5.7 Data Driven Analysis (both female and male combined)

	Naturalness	Intelligibility
Average	3.25	3.54
Standard Deviation	0.52	0.48

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

Different speech synthesis techniques are being developed and implemented to generate a natural and intelligible speech as speech synthesis systems are becoming more applicable. Moreover, the technologies have also advanced to incorporate speech synthesis systems in different computing devices.

In this thesis work, a first attempt is done to develop a speech synthesizer for Amharic language using Hidden Markov Model. To come up with the new synthesizer, first every component of HMM based speech synthesis system was studied to identify those components that are dependent on the characteristics of a language. Having those components in mind, the Amharic language was studied. However, every feature of the Amharic language was not considered since it needs a lot of time and deep linguistic knowledge. Hence, only the characteristics and way of creation of Amharic phonemes are considered.

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-cepstrum 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 pre-process 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.

One evaluation 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 (concatenative 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. Especially when we see the performance of HTS-FA for the criterion naturalness, it is good. As it is discussed in chapter two, parameter based speech synthesis systems, like formant based approach, are known in generating a speech with low naturalness. In contrast to this fact HTS-FA has generated a speech which sounds natural. Therefore, this is the benefit of using HMM based speech synthesis system over other parameter based speech synthesis techniques. Moreover, the performance of the system in generating intelligible speech is also good as per the result of MOS test.

6.2 Recommendation

In this study, a speech synthesis system for Amharic language using Hidden Markov Model is developed. Once the model is well built, one can synthesize a phoneme, a word, a phrase, a sentence or even a paragraph. However, there are language dependent features that are not being incorporated in this thesis work. Hence, the new system (HTS-FA) is not the last ever text to speech synthesis system, rather it is a one step in a continual development of speech synthesis system using HMM. Therefore, to have a speech synthesizer that considers all speech features, the following points are recommended as a future work either to extend the work or to increase the quality of the synthesized speech

- Factor of intonation is not considered in developing the system. It would be better if this factor is considered to handle contextual meanings of words.

- The HMM based speech synthesis technique has the capability to produce a synthetic speech with different melody/prosody once it is model trained. It would be very much advantageous if this capability is also applied to Amharic too.

- In this study, different prosodic effects that affect the sound of the speech were not considered. It would be better if they are considered to get full fledged speech synthesizer; since the technique has shown good result for other languages like English and German in modeling these prosodic effects.

- The texts that are provided to the HTS-FA 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.

References:

[1] Kroger B., "Minimal Rules for Articulatory Speech Synthesis". Proceedings of EUSIPCO92 (1): 331-334, 1992.

[2] Allen J., Hunnicutt S., Klatt D., "From Text to Speech: The MITalk System". Cambridge University Press, Inc. 1987.

[3] Redmond, Xuedong Huang, Alejandro Acero , Hsiao-Wuen Hon, "Spoken language processing: A guide to Theory, Algorithm, and System Developmnet ", 2001.

[4] O'Saughnessy D., "Speech Communication - Human and Machine", Addison-Wesley., 1987.

[5] Witten I., "Principles of Computer Speech", Academic Press Inc., 1982.

[6] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM Based Speech Synthesis," Proc. of EUROSPEECH, vol.5, pp.2347– 2350, 1999.

[7] Hidden Markov model, <http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev >, last updated 03/09/2007, accessed date 10/10/2007.

[8] Laine Berhane, "Text To speech Synthesis of the Amharic Language", Master's Thesis, Addis Ababa University, 1998.

[9] Henok Lulseged, "Concatinative Text-to-Speech (TTS) synthesis for Amharic language", Masters Thesis, Addis Ababa University, 2003.

[10] Habtamu Taye, "Diphone based Text-to-Speech Synthesis System for Amharic", Master's Project, Addis Ababa University, 2007.

[11] Nadew Tademe, "Formant based speech synthesis for Amharic vowels", Master's thesis ,Addis Ababa University, 2008.

[12] Oliveira L., Viana M., Trancoso I., “A Rule Based Text-to-Speech System for Portuguese”, Proceedings of ICASSP 92 (2): 73-76, 1992.

[13] K. Tokuda, Heiga Zen, Alan W. Black, “An HMM based speech synthesis system applied to English“, Nagoya institute of technology, 2002.

[14] K. Tokuda, T. Yoshimura, T. Kobayashi, “Speech parameter Generation Algorithm for HMM Based Speech Synthesis”, Proc. of ICASSP 2000, vol.3, pp.1315-1318, 2000.

[15] Ronald M. Kaplan, “A Method for Tokenizing Text”, Palo Alto, California 94304 USA, 1997.

[16] Sacha K., Anna H., Marc S., “An HMM-Based Speech Synthesis System applied to German and its Adaptation to a limited Set of Expressive Football Announcements”, Germany, 2002.

[17] S. Young, J. Odell, P.Woodland, “Tree-based state tying for high accuracy acoustic modeling,” in Proc. ARPA Workshop on Human Language Technology, 1994.

[18] K. Tokuda, T. Masuko, T. Yamada, “An algorithm for speech parameter generation from continuous mixture HMM with dynamic features”. In: Proc. Euro-speech, Madrid, Spain, pp. 757-760, 1995.

[19] M. Tamura, T. Masuko, K. Tokuda, T. Kobayashi, “Speaker adaptation for HMM-based speech synthesis system using MLLR”, In The third ESCA/COCOSDAWorkshop on Speech Synthesis, pages 273–276, November 1998.

[20] M. Tamura, T. Masuko, K. Tokuda, T. Kobayashi, “Text-to-speech Synthesis with arbitrary speaker’s voice from average voice”, In Proc. EUROSPEECH 2001, pages 345–348, September 2001.

[21] Steven K Smith,” Digital Signal Processing, A Practical Guide for Engineers and Scientists”, ISBN 0-75067444-X ,USA, 2003.

[22] Klatt D., “Review of Text-to-Speech Conversion for English,” Journal of Acoustical Society of America, pp. 737-793, 1987.

[23] Kleijn K., Paliwal K., “Speech Coding and Synthesis” Elsevier Science B.V., the Netherlands, 1998.

[24] Santen J., Sproat R., Olive J., Hirschberg J., “Progress in Speech Synthesis”, Springer-Verlag New York Inc, 1997.

[25] Sami Lemmetty, “Review of Speech Synthesis Technology”, Master’s Thesis, Helsinki University of Technology, 1999.

[26] Bickley, C.A., K.N. Stevens, and D.R. Williams, "A Framework for Synthesis of Segments Based on Pseudoarticulatory Parameters", New York, pp. 211-220, Springer-Verlag, 1997.

[27] Sebsibe H/Mariam, S P Kishore, Alan W Black, Rohit Kumar, ”Unit-selection voice for Amharic using Festvox”, Language Technologies Research Center, Hyderabad.

[28] ጌታሁን አማረ ፣ “የአማርኛ ሰዋስው በቀላል አቀራረብ”፣ አዲስ አበባ፣ (1989).

[29] Balentine, B., and D. Morgan, “How to Build a Speech Recognition Application”, Enterprise Integration Group, 1999.

[30] K. Tokuda, T. Masuko, N. Miyazaki, T. Kobayashi, “Hidden Markov Models Based on Multi-Space Probability Distribution for Pitch Pattern Modeling,” Proc. of ICASSP, 1999.

[31] Million Meshesha, C. V. Jawahar, ” Recognition of printed Amharic Documents”, International Institute of information Technology, Hyderabad.

[32] "What is phonetics", http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsPhonetics.htm , last updated 5 January 2004, last accessed on 3/16/2008.

[33] Takayoshi Yoshimura, “Simultaneous modeling of phonetic and prosodic parameters, and characteristics conversion for HMM-Based text-to-speech system”, Doctorial dissertation, Nagoya Institute of Technology, 2002.

[34] Junichi Yamagishi, “Average Voice based Speech Synthesis”, March 2006.

[35] Alan W Black, Paul Taylor and Richard Caley, “The Festival Speech Synthesis System”, system documentation, 2002.

[36] Kleijn B., Schroeter J., Sondhi M., “On the Use of Neural Networks in Articulatory Speech Synthesis”. Journal of the Acoustical Society of America, JASA vol. 93 (2): 1109-1121, 1998.

[37] Coralie Hemptinne, “Integration of the Harmonic Plus Noise Model into the HMM speech Synthesis System (HTS)”, Master’s thesis, IDIAP research group, 2006.

[38] Worku Alemu, “The Application of OCR Technique to the Amharic Script”, Masters Thesis, Addis Ababa University, 1997.

[39] Paul Taylor, “Text-to-Speech Synthesis”, University of Cambridge, 1999.

[40] Heiga Zen, Keiichi Tokuda, Tadashi Kitamura, **“**An Introduction of Trajectory Model into HMM-Based Speech Synthesis”, Nagoya Institute of Technology, Japan, 2002.

[41] S. Young, G. Everman, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. "The HTK Book Version 3.2.1”, December 2002.

[42] Ge'ez script, http://www.omniglot.com/writing/ethiopic.htm , modified on February 2008, accessed on March 22, 2008.

[43] English language phoneme set, http://www.antimoon.com/forum/posts/6356.htm, last updated date 1/17/ 2005, accessed date 5/8/2008.

Appendix A: HMM models (prototype of HMM)

```
~o <VecSize> 78 <USER> <DIAGC><MSDInfo> 4  0  1  1  1 <StreamInfo> 4 75 1 1 1
<BeginHMM>
  <NumStates> 7
  <State> 2
  <SWeights> 4 1.0 1.0 1.0 1.0
  <Stream> 1
    <Mean> 75
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
      0.0 0.0 0.0 0.0 0.0
    <Variance> 75
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
      1.0 1.0 1.0 1.0 1.0
  <Stream> 2
  <NumMixes> 2
  <Mixture> 1 0.5000
    <Mean> 1
      0.0
    <Variance> 1
      1.0
  <Mixture> 2 0.5000
    <Mean> 0
    <Variance> 0
  <Stream> 3
  <NumMixes> 2
  <Mixture> 1 0.5000
    <Mean> 1
      0.0
    <Variance> 1
      1.0
  <Mixture> 2 0.5000
    <Mean> 0
    <Variance> 0
```

<Stream> 4
<NumMixes> 2
<Mixture> 1 0.5000
<Mean> 1
0.0
<Variance> 1
1.0
<Mixture> 2 0.5000
<Mean> 0
<Variance> 0

•
•
•
•

<TransP> 7
0.000e+0 1.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0
0.000e+0 6.000e-1 4.000e-1 0.000e+0 0.000e+0 0.000e+0 0.000e+0
0.000e+0 0.000e+0 6.000e-1 4.000e-1 0.000e+0 0.000e+0 0.000e+0
0.000e+0 0.000e+0 0.000e+0 6.000e-1 4.000e-1 0.000e+0 0.000e+0
0.000e+0 0.000e+0 0.000e+0 0.000e+0 6.000e-1 4.000e-1 0.000e+0
0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0 6.000e-1 4.000e-1
0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0 0.000e+0
<EndHMM>

Appendix B: HMM parameter values after training

```
~o
<STREAMINFO> 4 75 1 1 1
<MSDINFO> 4 0 1 1 1
<VECSIZE> 78<NULLD><USER><DIAGC>
~h "a"
<BEGINHMM>
<NUMSTATES> 7
<STATE> 2
<STREAM> 1
<MEAN> 75
 6.693517e+00 1.695134e+00 7.244078e-01 5.384697e-01 7.522134e-02
8.403338e-02 -1.566391e-02 -1.374851e-01 -1.505398e-01 -3.716591e-02 -
6.227630e-02 4.887889e-02 -6.244192e-02 9.926887e-02 2.049330e-03
7.941793e-03 3.289547e-02 -6.532258e-02 1.999849e-02 -2.319381e-02 -
2.057564e-02 -1.529014e-02 -3.408495e-02 2.282176e-02 -2.859924e-03 -
4.782430e-02 5.161230e-03 4.958896e-03 5.663353e-03 1.367485e-02 8.486941e-
03 4.525048e-03 8.609721e-03 9.544555e-04 -1.191931e-03 -1.626729e-03 -
3.728639e-03 -1.462157e-03 1.408241e-03 1.893488e-03 2.349570e-03
1.487760e-03 -2.970539e-03 1.089684e-03 -1.727801e-03 2.366696e-03 -
6.327085e-04 1.136619e-03 -7.520532e-04 3.699983e-04 -9.092009e-03 -
1.475505e-03 -1.041418e-03 -2.278277e-03 1.669429e-03 4.025013e-04 -
8.336814e-04 4.353221e-03 3.100295e-03 2.854861e-03 3.028554e-03 -
8.901970e-05 1.305229e-03 2.239861e-04 1.359549e-03 -2.652593e-04 -
2.337949e-03 1.071327e-04 -1.953208e-03 1.145419e-03 -4.733365e-04
1.279232e-03 1.036349e-03 -5.045644e-04 4.920392e-05
<VARIANCE> 75
 5.085668e-01 1.778035e-01 6.548125e-02 1.210372e-01 7.276987e-02
4.075677e-02 5.007447e-02 4.956583e-02 2.987399e-02 3.155578e-02 4.623993e-
02 2.166080e-02 2.038163e-02 1.775854e-02 1.511216e-02 1.524771e-02
1.709748e-02 1.953967e-02 1.633199e-02 1.389173e-02 9.491248e-03 1.071938e-
02 1.010362e-02 9.133078e-03 7.602289e-03 2.095623e-02 5.418280e-03
4.276302e-03 3.893057e-03 3.588392e-03 3.277723e-03 3.291521e-03 3.489447e-
03 2.369432e-03 2.638450e-03 2.594000e-03 2.642473e-03 2.203780e-03
2.347692e-03 1.705867e-03 1.976260e-03 2.075311e-03 2.022423e-03 2.053814e-
03 2.077378e-03 1.600737e-03 1.547676e-03 1.771087e-03 1.785172e-03
1.660243e-03 1.237689e-02 5.855670e-03 6.009626e-03 5.191595e-03 5.171665e-
03 5.418999e-03 5.005392e-03 5.022834e-03 4.178424e-03 3.713517e-03
4.016446e-03 4.573830e-03 4.009293e-03 3.966196e-03 3.460601e-03 3.745027e-
03 3.892833e-03 3.599634e-03 4.251110e-03 3.758225e-03 3.657265e-03
3.409992e-03 3.779311e-03 3.699149e-03 3.686395e-03
<STREAM> 2
<NUMMIXES> 2
<MIXTURE> 1 9.204411e-01
<MEAN> 1
 5.161759e+00
<VARIANCE> 1
 6.033209e-02
<MIXTURE> 2 7.955896e-02
<MEAN> 0
<VARIANCE> 0
<STREAM> 3
<NUMMIXES> 2
<MIXTURE> 1 9.059848e-01
<MEAN> 1
 -3.181223e-03
<VARIANCE> 1
```

```
 3.141171e-04
<MIXTURE> 2 9.401518e-02
<MEAN> 0
<VARIANCE> 0
<STREAM> 4
<NUMMIXES> 2
<MIXTURE> 1 9.059848e-01
<MEAN> 1
 -2.531956e-04
<VARIANCE> 1
 6.909184e-04
<MIXTURE> 2 9.401520e-02
<MEAN> 0
<VARIANCE> 0
<STATE> 3
<STREAM> 1
<MEAN> 75
 6.278678e+00 1.498606e+00 6.780912e-01 4.821650e-01 1.125758e-01
1.835528e-01 1.968839e-02 -2.779491e-02 -7.119231e-02 3.040618e-02
4.401606e-02 5.933332e-02 5.388062e-03 1.146679e-01 1.385820e-02 2.196358e-
02 3.411189e-02 -3.171091e-02 9.266269e-03 -3.679173e-03 -1.356745e-02
1.667987e-03 -1.094922e-02 2.359273e-02 1.575763e-02 1.788221e-02
3.807760e-03 6.919238e-03 -5.671646e-03 -8.560908e-03 -1.018170e-02 -
2.838571e-03 -1.272653e-03 4.356237e-04 2.571471e-03 -4.927918e-04
4.292504e-03 4.761840e-05 1.516184e-03 5.483477e-03 -4.462001e-03 -
5.539907e-03 -1.583294e-03 -3.733432e-03 1.563170e-03 1.017021e-03
5.005164e-04 -9.506997e-04 -2.338832e-03 -4.682454e-04 3.068242e-02 -
5.096005e-03 -5.723048e-03 -4.562632e-03 -7.744453e-03 -8.667957e-03 -
2.210114e-03 -9.732557e-03 -5.086367e-03 -6.122919e-04 -2.324401e-03
8.842708e-04 3.011604e-04 6.803177e-04 -1.563109e-03 -1.811696e-03 -
5.006208e-04 4.724166e-05 3.149019e-03 2.040967e-03 1.726708e-03 -
1.181275e-03 -1.496310e-03 4.511388e-04 -2.496061e-04
<VARIANCE> 75
 6.055524e-01 3.231325e-01 7.579714e-02 9.569706e-02 6.516957e-02
4.499516e-02 3.494694e-02 3.424603e-02 2.920639e-02 3.067860e-02 3.128058e-
02 1.798806e-02 2.240982e-02 1.448179e-02 1.301102e-02 1.380515e-02
1.267742e-02 1.577971e-02 1.211608e-02 1.286749e-02 9.910688e-03 9.942727e-
03 9.876522e-03 9.964806e-03 7.574502e-03 5.952126e-02 2.962779e-02
1.002396e-02 8.456872e-03 5.910177e-03 5.624079e-03 4.395100e-03 5.626231e-
03 4.360421e-03 4.136642e-03 3.795546e-03 3.083191e-03 3.591360e-03
2.832577e-03 2.522090e-03 2.824532e-03 2.823251e-03 2.626681e-03 2.647764e-
03 2.534005e-03 2.312233e-03 2.203378e-03 2.243207e-03 2.093729e-03
1.908508e-03 3.055662e-02 2.330890e-02 1.399525e-02 1.148357e-02 9.384221e-
03 9.364595e-03 7.499142e-03 8.050506e-03 7.550687e-03 7.724090e-03
6.967897e-03 6.458165e-03 6.364993e-03 6.918008e-03 5.735351e-03 6.758419e-
03 6.842634e-03 5.917564e-03 6.230772e-03 5.891582e-03 5.874767e-03
6.148609e-03 5.646124e-03 5.705611e-03 4.694458e-03
<STREAM> 2
<NUMMIXES> 2
<MIXTURE> 1 6.239973e-01
<MEAN> 1
 5.156621e+00
<VARIANCE> 1
 6.033209e-02
<MIXTURE> 2 3.760027e-01
<MEAN> 0
<VARIANCE> 0
```

<STREAM> 3
<NUMMIXES> 2
<MIXTURE> 1 5.477383e-01
<MEAN> 1
 4.774486e-04
<VARIANCE> 1
 7.712571e-04
<MIXTURE> 2 4.522617e-01
<MEAN> 0
<VARIANCE> 0
<STREAM> 4
<NUMMIXES> 2
<MIXTURE> 1 5.477383e-01
<MEAN> 1
 -1.007120e-03
<VARIANCE> 1
 4.882841e-03
<MIXTURE> 2 4.522617e-01
<MEAN> 0
<VARIANCE> 0
<STATE> 4
<STREAM> 1
<MEAN> 75
 7.043646e+00 1.796505e+00 6.861151e-01 3.403526e-01 -1.284222e-01 -
1.416741e-02 -7.240297e-03 -1.336713e-01 -1.434562e-01 2.452471e-02 -
5.351168e-03 5.502967e-02 -1.550328e-02 8.947996e-02 -1.446152e-02 -
9.512611e-03 -3.163300e-02 -4.709482e-02 -3.533467e-02 2.268473e-02 -
4.433338e-02 8.548136e-03 -5.328271e-02 2.994988e-02 1.481973e-03
6.060326e-03 -1.132896e-03 -2.738448e-03 -3.415728e-03 -2.538713e-03 -
2.287537e-03 -2.361546e-03 -4.105459e-03 2.668310e-04 3.928750e-03
2.308729e-03 4.929170e-04 -5.672817e-04 2.487819e-05 -6.533007e-04 -
1.091174e-03 -8.932960e-04 2.996902e-04 1.971249e-03 2.758265e-03 -
8.859632e-04 6.881410e-05 -1.301605e-03 9.253227e-04 1.696174e-03 -
1.697731e-02 5.994376e-04 2.117001e-03 4.073712e-03 3.922684e-03 3.717485e-
03 1.313156e-03 1.203849e-03 2.827008e-04 -7.773394e-04 8.705141e-04
4.231636e-04 8.508265e-05 -2.043032e-04 -7.122443e-06 2.627003e-04
2.354637e-03 1.340303e-03 2.911325e-04 -1.235957e-03 5.418149e-05 -
5.626043e-04 1.054618e-03 1.472600e-03 -8.889937e-04
<VARIANCE> 75
 5.604838e-01 1.235196e-01 7.008188e-02 1.157278e-01 1.050474e-01
3.863570e-02 6.307460e-02 5.546012e-02 2.434561e-02 3.083648e-02 5.203298e-
02 2.034275e-02 2.439404e-02 2.074945e-02 1.609071e-02 1.309721e-02
1.715907e-02 2.122143e-02 1.588939e-02 1.457415e-02 9.298984e-03 9.420286e-
03 9.714190e-03 9.632085e-03 7.183479e-03 6.875387e-03 1.854546e-03
2.386807e-03 2.722363e-03 2.820272e-03 2.090669e-03 2.157948e-03 1.811561e-
03 1.807893e-03 1.628090e-03 1.915122e-03 1.694244e-03 1.694892e-03
1.654026e-03 1.381307e-03 1.403324e-03 1.442034e-03 1.643748e-03 1.511924e-
03 1.664349e-03 1.362969e-03 1.225923e-03 1.331290e-03 1.483254e-03
1.286929e-03 4.059764e-03 2.930453e-03 3.699102e-03 3.153789e-03 3.899152e-
03 3.397994e-03 3.693111e-03 2.933645e-03 3.057291e-03 2.790062e-03
3.156682e-03 3.085183e-03 2.859406e-03 2.793697e-03 2.710057e-03 2.549420e-
03 2.649027e-03 2.779346e-03 3.075911e-03 2.908221e-03 2.498118e-03
2.648853e-03 2.739348e-03 2.895948e-03 2.689846e-03
<STREAM> 2
<NUMMIXES> 2
<MIXTURE> 1 9.389691e-01
<MEAN> 1

```
 5.162688e+00
<VARIANCE> 1
 6.033209e-02
<MIXTURE> 2 6.103094e-02
<MEAN> 0
<VARIANCE> 0
<STREAM> 3
<NUMMIXES> 2
<MIXTURE> 1 9.347528e-01
<MEAN> 1
 1.916319e-04
<VARIANCE> 1
 1.161532e-04
<MIXTURE> 2 6.524722e-02
<MEAN> 0
<VARIANCE> 0
<STREAM> 4
<NUMMIXES> 2
<MIXTURE> 1 9.347528e-01
<MEAN> 1
 -4.350170e-04
<VARIANCE> 1
 1.246891e-04
<MIXTURE> 2 6.524722e-02
<MEAN> 0
<VARIANCE> 0
<STATE> 5
<STREAM> 1
<MEAN> 75
 5.727677e+00 1.354520e+00 5.986654e-01 3.460865e-01 9.908057e-02
1.689584e-01 6.330473e-02 3.990283e-02 -5.297328e-03 1.208424e-01
1.077406e-01 1.338060e-01 9.513786e-02 1.312576e-01 7.377761e-02 5.629250e-
02 5.166267e-02 3.101170e-02 4.390126e-02 5.924209e-02 4.093626e-02
4.490296e-02 3.273711e-02 5.521361e-02 4.398669e-02 -3.503108e-02 -
6.678469e-03 -5.590024e-03 3.201214e-03 6.389000e-03 4.654499e-03
2.752214e-03 1.316455e-03 2.925453e-03 1.535571e-03 3.695855e-03 2.944939e-
03 2.684326e-03 1.253083e-03 2.286199e-03 1.328538e-03 4.164519e-03
3.230717e-03 1.244622e-03 1.171826e-03 2.052583e-03 4.087024e-04 3.759873e-
03 3.418970e-03 3.454153e-04 -1.890451e-03 2.303253e-03 7.973950e-04
2.077967e-03 -4.833097e-04 1.800695e-03 7.994809e-04 2.864533e-03
1.278403e-03 5.487060e-04 -4.889463e-04 -1.365041e-03 4.955665e-05
1.055051e-03 4.378113e-05 2.023660e-03 1.107753e-03 4.150323e-04 -
9.400024e-04 -4.100590e-04 -1.044390e-03 -1.772776e-04 -1.216896e-03 -
2.543709e-03 -3.784621e-04
<VARIANCE> 75
 1.415342e+00 5.290374e-01 5.765979e-02 6.284116e-02 4.647157e-02
3.013191e-02 3.868406e-02 3.553712e-02 2.207143e-02 1.965774e-02 2.756054e-
02 1.717115e-02 2.135768e-02 1.388726e-02 1.265049e-02 1.102166e-02
1.211101e-02 1.290794e-02 1.240424e-02 1.083455e-02 9.748545e-03 9.932678e-
03 9.750019e-03 9.463318e-03 7.275990e-03 7.420799e-03 3.682088e-03
3.154644e-03 2.775823e-03 2.575679e-03 2.445902e-03 2.876446e-03 2.460808e-
03 2.272227e-03 2.011186e-03 2.214902e-03 2.422387e-03 2.265791e-03
2.210488e-03 2.050046e-03 1.984063e-03 2.027036e-03 2.142995e-03 2.009678e-
03 1.953331e-03 1.981975e-03 1.731600e-03 1.988962e-03 2.067169e-03
1.835720e-03 6.630040e-03 6.112707e-03 6.295932e-03 5.862611e-03 6.680402e-
03 5.395688e-03 6.358483e-03 5.685916e-03 5.159392e-03 4.998310e-03
5.349897e-03 5.649905e-03 5.641629e-03 5.248017e-03 5.153224e-03 4.491934e-
```

03 5.349625e-03 5.545142e-03 4.996600e-03 5.119096e-03 5.229029e-03
4.604074e-03 4.883014e-03 5.890934e-03 4.887441e-03
<STREAM> 2
<NUMMIXES> 2
<MIXTURE> 1 2.414400e-01
<MEAN> 1
 5.151020e+00
<VARIANCE> 1
 6.033209e-02
<MIXTURE> 2 7.585600e-01
<MEAN> 0
<VARIANCE> 0
<STREAM> 3
<NUMMIXES> 2
<MIXTURE> 1 2.267198e-01
<MEAN> 1
 -1.828525e-03
<VARIANCE> 1
 2.233700e-04
<MIXTURE> 2 7.732802e-01
<MEAN> 0
<VARIANCE> 0
<STREAM> 4
<NUMMIXES> 2
<MIXTURE> 1 2.267198e-01
<MEAN> 1
 6.920523e-05
<VARIANCE> 1
 3.928166e-04
<MIXTURE> 2 7.732802e-01
<MEAN> 0
<VARIANCE> 0
<STATE> 6
<STREAM> 1
<MEAN> 75
 6.375810e+00 1.416337e+00 6.120916e-01 4.347261e-01 7.111742e-02
1.507254e-01 -4.918654e-03 -6.239764e-02 -5.474234e-02 5.798263e-02
5.937327e-02 6.571139e-02 1.661509e-02 1.146361e-01 2.988519e-02 3.667788e-
02 4.187375e-02 -1.912386e-02 2.440465e-02 2.678607e-02 -1.203114e-04
2.031328e-02 2.529554e-03 3.998952e-02 1.812448e-02 6.737453e-03 -
4.210028e-02 -6.366177e-03 -2.483399e-03 -4.788878e-03 3.222343e-03 -
4.983085e-03 2.855417e-03 5.490100e-05 -3.382825e-04 1.986564e-03 -
3.340242e-03 3.118172e-03 7.679943e-04 -8.518401e-04 1.998676e-03
2.455553e-03 2.027737e-03 -2.279908e-03 6.652349e-04 2.911296e-04 -
1.774070e-03 -1.876487e-03 -1.046925e-03 -2.226060e-03 -2.870052e-02 -
2.104702e-02 -8.014372e-03 -1.074686e-02 -2.850820e-03 -1.982158e-03 -
3.058532e-03 -1.486173e-03 -2.457478e-03 -2.711211e-04 -4.576296e-04 -
2.568571e-03 -2.050402e-03 -6.896429e-04 2.465964e-03 1.682394e-04 -
3.946231e-03 -3.433756e-03 -1.505385e-03 -4.592078e-04 -1.288647e-03 -
3.106231e-04 -1.290761e-03 3.887672e-04 1.167614e-03
<VARIANCE> 75
 9.554256e-01 3.831314e-01 9.248934e-02 9.235999e-02 6.450356e-02
4.491321e-02 4.084324e-02 4.327960e-02 3.157597e-02 2.377941e-02 3.833020e-
02 2.121047e-02 2.573639e-02 1.603843e-02 1.556201e-02 1.460111e-02
1.620272e-02 1.627732e-02 1.268966e-02 1.448991e-02 1.291825e-02 1.232922e-
02 1.336503e-02 1.216926e-02 1.004357e-02 1.525052e-01 4.094885e-02
1.123526e-02 7.556801e-03 5.550422e-03 4.643850e-03 4.598497e-03 4.829783e-

```
03 3.148189e-03 2.894968e-03 3.354277e-03 3.189505e-03 3.510293e-03
2.952863e-03 2.444158e-03 2.397527e-03 2.568377e-03 2.692831e-03 2.354760e-
03 2.480307e-03 2.368008e-03 2.190285e-03 1.991881e-03 2.376270e-03
1.881668e-03 3.267947e-01 7.167968e-02 1.938664e-02 1.450819e-02 9.123934e-
03 9.951485e-03 8.090367e-03 7.993930e-03 6.484723e-03 6.806205e-03
6.768832e-03 7.955289e-03 6.671183e-03 6.575807e-03 6.055509e-03 6.488629e-
03 5.925934e-03 5.131976e-03 5.896883e-03 5.423291e-03 6.176044e-03
5.523460e-03 5.831941e-03 6.407597e-03 4.968999e-03
<STREAM> 2
<NUMMIXES> 2
<MIXTURE> 1 5.279024e-01
<MEAN> 1
 5.173549e+00
<VARIANCE> 1
 6.033209e-02
<MIXTURE> 2 4.720976e-01
<MEAN> 0
<VARIANCE> 0
<STREAM> 3
<NUMMIXES> 2
<MIXTURE> 1 4.912901e-01
<MEAN> 1
 1.540814e-03
<VARIANCE> 1
 4.174368e-04
<MIXTURE> 2 5.087100e-01
<MEAN> 0
<VARIANCE> 0
<STREAM> 4
<NUMMIXES> 2
<MIXTURE> 1 4.912901e-01
<MEAN> 1
 -1.933297e-03
<VARIANCE> 1
 1.952052e-03
<MIXTURE> 2 5.087100e-01
<MEAN> 0
<VARIANCE> 0
<TRANSP> 7
0.000000e+00 1.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00
0.000000e+00 7.205893e-01 2.794107e-01 0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00 6.474091e-01 3.525909e-01 0.000000e+00
0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 7.895981e-01 2.104018e-01
0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 6.841003e-01 3.158997e-
01 0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 4.577166e-
01 5.422834e-01
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00
<ENDHMM>
```

Appendix C: Amharic Phonetic List, IPA Equivalence and its ASCII Transliteration

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

Appendix D: Amharic alphabets with their seven orders

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

Appendix E: labialized Letter

ቈ	ቍ	ቋ	ቌ	ቍ
ኈ	ኍ	ኋ	ኌ	ኍ
ጐ	ጕ	ጓ	ጔ	ጕ
ኰ	ኵ	ካ	ኴ	ኵ
ጐ	ጕ	ጓ	ጔ	ጕ

Appendix F: Amharic phoneme set and their corresponding features, including silence.

phonemes & silence	features
SIL	- 0 0 0 - 0 0 -
h	- 0 0 0 + f v -
l	- 0 0 0 + l a +
m	- 0 0 0 + n l +
s	- 0 0 0 + f a -
r	- 0 0 0 + l a +
xx	- 0 0 0 + f a -
sx	- 0 0 0 + f p -
q	- 0 0 0 + s v -
b	- 0 0 0 + s l +
t	- 0 0 0 + s a -
c	- 0 0 0 + a p -
n	- 0 0 0 + n a +
nx	- 0 0 0 + n p +
e	+ s 2 2 - 0 0 0
u	+ s 3 3 + 0 0 0
ii	+ s 3 1 - 0 0 0
a	+ s 1 2 - 0 0 0
ie	+ s 2 1 - 0 0 0
ix	+ s 3 2 - 0 0 0
o	+ s 2 3 - 0 0 0
k	- 0 0 0 + s v -
w	- 0 0 0 + f l +
z	- 0 0 0 + f a +
zx	- 0 0 0 + f p +
y	- 0 0 0 + l p +
d	- 0 0 0 + s a +
g	- 0 0 0 + s v +
j	- 0 0 0 + a p +
tx	- 0 0 0 + s a -
cx	- 0 0 0 + a p -
px	- 0 0 0 + s l -
f	- 0 0 0 + f b -
p	- 0 0 0 + s l -
v	- 0 0 0 + f b +

Appendix G: Utterance Structure

```
EST_File utterance
DataType ascii
version 2
EST_Header_End
Features max_id 97 ; type Text ; iform "\"betxam amesegixnalew .\"" ;
filename prompt-utt/amharic042.utt ; fileid amharic042 ;
Stream_Items
1 id _1 ; name betxam ; whitespace "" ; prepunctuation "" ;
2 id _2 ; name amesegixnalew ; whitespace " " ; prepunctuation "" ;
3 id _3 ; name . ; whitespace " " ; prepunctuation "" ;
4 id _6 ; name . ; pbreak B ; pos punc ;
5 id _5 ; name amesegixnalew ; pbreak B ; pos nil ;
6 id _4 ; name betxam ; pbreak NB ; pos nil ;
7 id _7 ; name B ;
8 id _8 ; name syl ; stress 1 ;
9 id _10 ; name syl ; stress 0 ;
10 id _12 ; name syl ; stress 0 ;
11 id _14 ; name syl ; stress 0 ;
12 id _16 ; name syl ; stress 0 ;
13 id _18 ; name syl ; stress 1 ;
14 id _20 ; name syl ; stress 0 ;
15 id _22 ; name syl ; stress 0 ;
16 id _24 ; name syl ; stress 0 ;
17 id _26 ; name syl ; stress 0 ;
18 id _28 ; name syl ; stress 0 ;
19 id _30 ; name syl ; stress 0 ;
20 id _32 ; name syl ; stress 0 ;
21 id _34 ; name syl ; stress 0 ;
22 id _36 ; name syl ; stress 0 ;
23 id _38 ; name syl ; stress 0 ;
24 id _40 ; name syl ; stress 0 ;
25 id _42 ; name SIL ; dur_factor 1 ; end 0.24 ; source_end 0.045125 ;
26 id _9 ; name b ; dur_factor 1 ; end 0.335 ; source_end 0.146187 ;
27 id _11 ; name e ; dur_factor 1 ; end 0.415 ; source_end 0.247249 ;
28 id _13 ; name tx ; dur_factor 1 ; end 0.475 ; source_end 0.348311 ;
29 id _15 ; name a ; dur_factor 1 ; end 0.61 ; source_end 0.449373 ;
30 id _17 ; name m ; dur_factor 1 ; end 0.68 ; source_end 0.550435 ;
31 id _19 ; name a ; dur_factor 1 ; end 0.82 ; source_end 0.651497 ;
32 id _21 ; name m ; dur_factor 1 ; end 0.88 ; source_end 0.752559 ;
33 id _23 ; name e ; dur_factor 1 ; end 0.93 ; source_end 0.853621 ;
34 id _25 ; name s ; dur_factor 1 ; end 1.025 ; source_end 0.954683 ;
35 id _27 ; name e ; dur_factor 1 ; end 1.09 ; source_end 1.05575 ;
36 id _29 ; name g ; dur_factor 1 ; end 1.2 ; source_end 1.15681 ;
37 id _31 ; name ix ; dur_factor 1 ; end 1.245 ; source_end 1.25787 ;
38 id _33 ; name n ; dur_factor 1 ; end 1.285 ; source_end 1.35893 ;
39 id _35 ; name a ; dur_factor 1 ; end 1.385 ; source_end 1.45999 ;
40 id _37 ; name l ; dur_factor 1 ; end 1.5 ; source_end 1.56106 ;
41 id _39 ; name e ; dur_factor 1 ; end 1.535 ; source_end 1.66212 ;
42 id _41 ; name w ; dur_factor 1.5 ; end 1.565 ; source_end 1.76318 ;
43 id _43 ; name SIL ; dur_factor 1 ; end 1.75 ; source_end 1.87505 ;
44 id _44 ; name Accented ;
45 id _45 ; name Accented ;
46 id _54 ; f0 110 ; pos 3.72098 ;
47 id _51 ; f0 131.95 ; pos 2.01933 ;
```

```
48 id _52 ; f0 129.95 ; pos 2.07539 ;
49 id _53 ; f0 121.645 ; pos 2.08539 ;
50 id _50 ; f0 121.95 ; pos 2.00933 ;
51 id _47 ; f0 134.907 ; pos 1.28123 ;
52 id _48 ; f0 132.907 ; pos 1.44108 ;
53 id _49 ; f0 124.187 ; pos 1.45108 ;
54 id _46 ; f0 124.907 ; pos 1.27123 ;
55 id _55 ; name SIL-b ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.101062 ; num_frames 9 ;
56 id _56 ; name b-e ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.202124 ; num_frames 9 ;
57 id _57 ; name e-tx ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.303186 ; num_frames 9 ;
58 id _58 ; name tx-a ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.404248 ; num_frames 9 ;
59 id _59 ; name a-m ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.50531 ; num_frames 9 ;
60 id _60 ; name m-a ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.606372 ; num_frames 9 ;
61 id _61 ; name a-m ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.707434 ; num_frames 9 ;
62 id _62 ; name m-e ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.808496 ; num_frames 9 ;
63 id _63 ; name e-s ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 0.909558 ; num_frames 9 ;
64 id _64 ; name s-e ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.01062 ; num_frames 9 ;
65 id _65 ; name e-g ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.11168 ; num_frames 9 ;
66 id _66 ; name g-ix ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.21274 ; num_frames 9 ;
67 id _67 ; name ix-n ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.31381 ; num_frames 9 ;
68 id _68 ; name n-a ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.41487 ; num_frames 9 ;
69 id _69 ; name a-l ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.51593 ; num_frames 9 ;
70 id _70 ; name l-e ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.61699 ; num_frames 9 ;
71 id _71 ; name e-w ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.71805 ; num_frames 9 ;
72 id _72 ; name w-SIL ; sig "[Val wave]" ; coefs "[Val track]" ;
middle_frame 3 ; end 1.81912 ; num_frames 9 ;
73 id _73 ; name coef ; coefs "[Val track]" ; frame "[Val wavevector]" ;
74 id _74 ; name f0 ; f0 "[Val track]" ;
75 id _75 ; coefs "[Val track]" ;
76 id _76 ;
77 id _77 ; map "[Val ivector]" ;
78 id _78 ; wave "[Val wave]" ;
End_of_Stream_Items
Relations
Relation Token ; "(" ")" ;
4 4 3 0 0 0
3 3 0 4 0 2
5 5 2 0 0 0
2 2 0 5 3 1
6 6 1 0 0 0
```

```
1 1 0 6 2 0
End_of_Relation
Relation Word ; "(" ")" ;
2 5 0 0 0 1
1 6 0 0 2 0
End_of_Relation
Relation Phrase ; "(" ")" ;
3 5 0 0 0 2
2 6 1 0 3 0
1 7 0 2 0 0
End_of_Relation
Relation Syllable ; "(" ")" ;
17 24 0 0 0 16
16 23 0 0 17 15
15 22 0 0 16 14
14 21 0 0 15 13
13 20 0 0 14 12
12 19 0 0 13 11
11 18 0 0 12 10
10 17 0 0 11 9
9 16 0 0 10 8
8 15 0 0 9 7
7 14 0 0 8 6
6 13 0 0 7 5
5 12 0 0 6 4
4 11 0 0 5 3
3 10 0 0 4 2
2 9 0 0 3 1
1 8 0 0 2 0
End_of_Relation
Relation Segment ; "(" ")" ;
19 43 0 0 0 18
18 42 0 0 19 17
17 41 0 0 18 16
16 40 0 0 17 15
15 39 0 0 16 14
14 38 0 0 15 13
13 37 0 0 14 12
12 36 0 0 13 11
11 35 0 0 12 10
10 34 0 0 11 9
9 33 0 0 10 8
8 32 0 0 9 7
7 31 0 0 8 6
6 30 0 0 7 5
5 29 0 0 6 4
4 28 0 0 5 3
3 27 0 0 4 2
2 26 0 0 3 1
1 25 0 0 2 0
End_of_Relation
Relation SylStructure ; "(" ")" ;
3 4 0 0 0 2
16 42 15 0 0 0
15 24 0 16 0 14
17 41 14 0 0 0
14 23 0 17 15 13
```

```
18 40 13 0 0 0
13 22 0 18 14 12
19 39 12 0 0 0
12 21 0 19 13 11
20 38 11 0 0 0
11 20 0 20 12 10
21 37 10 0 0 0
10 19 0 21 11 9
22 36 9 0 0 0
9 18 0 22 10 8
23 35 8 0 0 0
8 17 0 23 9 7
24 34 7 0 0 0
7 16 0 24 8 6
25 33 6 0 0 0
6 15 0 25 7 5
26 32 5 0 0 0
5 14 0 26 6 4
27 31 4 0 0 0
4 13 2 27 5 0
2 5 0 4 3 1
33 30 32 0 0 0
32 12 0 33 0 31
34 29 31 0 0 0
31 11 0 34 32 30
35 28 30 0 0 0
30 10 0 35 31 29
36 27 29 0 0 0
29 9 0 36 30 28
37 26 28 0 0 0
28 8 1 37 29 0
1 6 0 28 2 0
End_of_Relation
Relation IntEvent ; "(" ")" ;
2 45 0 0 0 1
1 44 0 0 2 0
End_of_Relation
Relation Intonation ; "(" ")" ;
3 45 2 0 0 0
2 13 0 3 0 1
4 44 1 0 0 0
1 8 0 4 2 0
End_of_Relation
Relation Target ; "(" ")" ;
6 46 5 0 0 0
5 42 0 6 0 4
9 49 0 0 0 8
8 48 0 0 9 7
7 47 4 0 8 0
4 31 0 7 5 3
10 50 3 0 0 0
3 30 0 10 4 2
13 53 0 0 0 12
12 52 0 0 13 11
11 51 2 0 12 0
2 26 0 11 3 1
14 54 1 0 0 0
```

```
1 25 0 14 2 0
End_of_Relation
Relation Unit; grouped 1 ;
18 72 0 0 0 17
17 71 0 0 18 16
16 70 0 0 17 15
15 69 0 0 16 14
14 68 0 0 15 13
13 67 0 0 14 12
12 66 0 0 13 11
11 65 0 0 12 10
10 64 0 0 11 9
9 63 0 0 10 8
8 62 0 0 9 7
7 61 0 0 8 6
6 60 0 0 7 5
5 59 0 0 6 4
4 58 0 0 5 3
3 57 0 0 4 2
2 56 0 0 3 1
1 55 0 0 2 0
End_of_Relation
Relation SourceCoef ; "(" ")" ;
1 73 0 0 0 0
End_of_Relation
Relation f0 ; "(" ")" ;
1 74 0 0 0 0
End_of_Relation
Relation TargetCoef ; "(" ")" ;
2 76 0 0 0 1
1 75 0 0 2 0
End_of_Relation
Relation US_map ; "(" ")" ;
1 77 0 0 0 0
End_of_Relation
Relation Wave ; "(" ")" ;
1 78 0 0 0 0
End_of_Relation
End_of_Relations
End_of_Utterance
```

Appendix H: Labeled text for the transcribed text “betxam amesegixnalehu”

```
x^x-SIL+b=e@1_0/A:0_0_0/B:0-0-0@1-0&1-1#1-1$1-1!0-0;0-
0|0/C:0+0+0/D:0_0/E:0+0@1+0&1+0#0+0/F:0_0/G:0_0/H:0=0@1=1|0/I:0=0/J:17+2-1
x^SIL-b+e=tx@1_1/A:0_0_0/B:1-1-1@1-5&1-17#1-2$1-2!0-5;0-
5|novowel/C:0+0+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
SIL^b-e+tx=a@1_1/A:1_1_1/B:0-0-1@2-4&2-16#1-2$1-2!1-4;1-
4|e/C:0+0+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
b^e-tx+a=m@1_1/A:0_0_1/B:0-0-1@3-3&3-15#1-2$1-2!2-3;2-
3|novowel/C:0+0+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
e^tx-a+m=a@1_1/A:0_0_1/B:0-0-1@4-2&4-14#1-2$1-2!3-2;3-
2|a/C:0+0+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
tx^a-m+a=m@1_1/A:0_0_1/B:0-0-1@5-1&5-13#1-2$1-2!4-1;4-
1|novowel/C:1+1+1/D:0_0/E:content+5@1+2&1+1#0+1/F:content_12/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
a^m-a+m=e@1_1/A:0_0_1/B:1-1-1@1-12&6-12#1-1$1-1!5-0;5-
0|a/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
m^a-m+e=s@1_1/A:1_1_1/B:0-0-1@2-11&7-11#2-1$2-1!1-0;1-
0|novowel/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
a^m-e+s=e@1_1/A:0_0_1/B:0-0-1@3-10&8-10#2-1$2-1!2-0;2-
0|e/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
m^e-s+e=g@1_1/A:0_0_1/B:0-0-1@4-9&9-9#2-1$2-1!3-0;3-
0|novowel/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
e^s-e+g=ix@1_1/A:0_0_1/B:0-0-1@5-8&10-8#2-1$2-1!4-0;4-
0|e/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
s^e-g+ix=n@1_1/A:0_0_1/B:0-0-1@6-7&11-7#2-1$2-1!5-0;5-
0|novowel/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
e^g-ix+n=a@1_1/A:0_0_1/B:0-0-1@7-6&12-6#2-1$2-1!6-0;6-
0|ix/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NO
NE/I:0=0/J:17+2-1
g^ix-n+a=l@1_1/A:0_0_1/B:0-0-1@8-5&13-5#2-1$2-1!7-0;7-
0|novowel/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
ix^n-a+l=e@1_1/A:0_0_1/B:0-0-1@9-4&14-4#2-1$2-1!8-0;8-
0|a/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
n^a-l+e=w@1_1/A:0_0_1/B:0-0-1@10-3&15-3#2-1$2-1!9-0;9-
0|novowel/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
a^l-e+w=SIL@1_1/A:0_0_1/B:0-0-1@11-2&16-2#2-1$2-1!10-0;10-
0|e/C:0+0+1/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1=1|NON
E/I:0=0/J:17+2-1
l^e-w+SIL=x@1_1/A:0_0_1/B:0-0-1@12-1&17-1#2-1$2-1!11-0;11-
0|novowel/C:0+0+0/D:content_5/E:content+12@2+1&2+0#1+0/F:0_0/G:0_0/H:17=2@1
=1|NONE/I:0=0/J:17+2-1
e^w-SIL+x=x@1_0/A:0_0_0/B:0-0-0@1-0&1-1#1-1$1-1!0-0;0-
0|0/C:0+0+0/D:0_0/E:0+0@1+0&1+0#0+0/F:0_0/G:0_0/H:0=0@1=1|0/I:0=0/J:17+2-1
```

Appendix I: Questionnaire

Addis Ababa University

Department of Computer Science

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

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

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

1. How do you judge the understandability of the synthesized speech?
    a. Excellent
    b. Very Good
    c. Good
    d. Fair
    f. Poor
    g. Very poor

2. How do you judge the naturalness of the synthesized speech?
    a. Excellent
    b. Very Good
    c. Good
    d. Fair
    f. Poor
    g. Very poor

Thank you

Appendix J: Short Description of the open source software used in this thesis work

- **HTK Ver 3.4** (Hidden Markov Model Tool Kit):- A set of tools for building and manipulating Hidden Markov Models provided by the Cambridge University Engineering Department.
- **HTS Ver 2.0.1**:- A HMM- based speech synthesis system developed by the Nagoya Institute of Technology.
- **Festival Speech synthesis Systems Ver 1.96**:- A set of tools for building speech synthesis systems provided by the University of Edinburgh and Carneggie Mellon University including a full TTS Systems.
- **SPTK Ver 3.4**:- speech signal processing toolkit from Nagoya Institute of technology.
- **Wavesurfer Ver 1.8.5**:- A tool for waveform visualization and manipulation developed by the Centre for speech Technology at KTH in Stockholm, Sweden.
- **Sox-14.0.1**:- A tool for changing speech data encodings from one form to another.

Appendix K: Question set

```
QS "L-Vowel" {*^a-*,*^e-*,*^ii-*,*^ix-*,*^ie-*,*^o-*,*^u-*}
QS "L-Consonant" {*^b-*,*^c-*,*^cx-*,*^d-*,*^f-*,*^g-*,*^h-*,*^j-*,*^k-
*,*^l-*,*^m-*,*^n-*,*^nx-*,*^q-*,*^px-*,*^p-*,*^r-*,*^s-*,*^sx-*,*^t-
*,*^tx-*,*^v-*,*^w-*,*^y-*,*^z-*,*^zx-*,*^xx-*}
QS "L-Stop" {*^b-*,*^d-*,*^px-*,*^g-*,*^k-*,*^p-*,*^t-*,*^tx-*,*^q-*}
QS "L-Nasal"      {*^m-*,*^n-*,*^nx-*}
QS "L-Fricative"  {*^f-*,*^xx-*,*^v-*,*^s-*,*^sx-*,*^z-*,*^zx-*}
QS "L-Liquid"     {*^l-*,*^r-*}
QS "L-Front"      {*^ii-*,*^ie-*,*^px-*,*^f-*,*^m-*,*^p-*,*^v-*,*^w-*}
QS "L-Central"    {*^a-*,*^e-*,*^ix-*,*^d-*,*^tx-*,*^sx-*,*^l-*,*^n-*,*^nx-
*,*^r-*,*^s-*,*^t-*,*^xx-*,*^z-*,*^zx-*,*^c-*,*^j-*,*^cx-*}
QS "L-Back" {*^aa-*,*^ax-*,*^ch-*,*^g-*,*^q-*,*^jh-*,*^k-*,*^ng-*,*^o-
*,*^sh-*,*^u-*,*^uw-*,*^y-*}
QS "L-Front_Vowel"            {*^ii-*,*^ie-*}
QS "L-Central_Vowel"          {*^a-*,*^ix-*,*^e-*}
QS "L-Back_Vowel"       {*^o-*,*^u-*}
QS "L-Long_Vowel"       {*^ii-*,*^ie-*}
QS "L-Short_Vowel" {*^a-*,*^ah-*,*^ax-*,*^ay-*,*^ey-*,*^ix-*,*^oy-*,*^u-*}
QS "L-Front_Start_Vowel"      {*^aw-*,*^axr-*,*^ie-*,*^ii-*}
QS "L-Fronting_Vowel"         {*^ay-*,*^ey-*,*^o-*}
QS "L-High_Vowel"       {*^ii-*,*^ix-*,*^iy-*,*^u-*,*^uw-*}
QS "L-Medium_Vowel"           {*^ie-*,*^e-*,*^o-*}
QS "L-Low_Vowel"        {*^a-*}
QS "L-Rounded_Vowel"          {*^o-*,*^u-*}
QS "L-Unrounded_Vowel"        {*^a-*,*^e-*,*^ie-*,*^ii-*,*^ix-*}
QS "L-Reduced_Vowel"          {*^ax-*,*^axr-*,*^ix-*}
QS "L-IVowel"                 {*^ix-*,*^ii-*,*^ie-*}
QS "L-EVowel"                 {*^e-*}
QS "L-AVowel"                 {*^a-*}
QS "L-OVowel"                 {*^o-*}
QS "L-UVowel"                 {*^u-*}
QS "L-Unvoiced_Consonant"     {*^c-*,*^f-*,*^k-*,*^p-*,*^s-*,*^sx-*,*^t-
*,*^tx-*}
QS "L-Voiced_Consonant"       {*^b-*,*^d-*,*^g-*,*^l-*,*^m-*,*^n-*,*^nx-
*,*^r-*,*^v-*,*^w-*,*^y-*,*^z-*,*^zx-*}
QS "L-Front_Consonant"        {*^b-*,*^f-*,*^m-*,*^p-*,*^v-*,*^w-*,*^px-*}
QS "L-Central_Consonant"      {*^d-*,*^c-*,*^cx-*,*^j-*,*^l-*,*^n-*,*^nx-
*,*^r-*,*^s-*,*^t-*,*^tx-*,*^z-*,*^zx-*,*^xx-*,*^y-*}
QS "L-Back_Consonant"         {*^ch-*,*^g-*,*^hh-*,*^jh-*,*^k-*,*^ng-*,*^sh-
*,*^q-*}
QS "L-Fortis_Consonant"       {*^ch-*,*^f-*,*^k-*,*^p-*,*^s-*,*^sh-*,*^t-
*,*^th-*}
QS "L-Voiced_Stop"            {*^b-*,*^d-*,*^g-*}
QS "L-Unvoiced_Stop"          {*^p-*,*^t-*,*^k-*}
QS "L-Front_Stop"       {*^b-*,*^p-*,*^px-*}
QS "L-Central_Stop"           {*^d-*,*^t-*,*^tx-*}
QS "L-Back_Stop"        {*^g-*,*^k-*,*^q-*}
QS "L-Voiced_Fricative"       {*^v-*,*^z-*,*^xx-*,*^zx-*}
QS "L-Unvoiced_Fricative"     {*^f-*,*^s-*,*^sx-*,*^tx-*}
QS "L-Front_Fricative"        {*^f-*,*^v-*}
QS "L-Central_Fricative"      {*^s-*,*^z-**^xx-*}
QS "L-Back_Fricative"         {*^sx-*,*^zx-*}
QS "L-Affricate_Consonant"    {*^c-*,*^j-*,*^cx-*}
QS "L-Not_Affricate"          {*^f-*,*^s-*,*^sx-*,*^tx-*,*^v-*,*^z-*,*^zx-
*,*^xx-*}
```

```
QS "L-silences"                {*^SIL-*,*^pau-*,*^h#-*,*^brth-*}
QS "L-p"                             {*^p-*}
QS "L-t"                             {*^t-*}
QS "L-k"                             {*^k-*}
QS "L-b"                             {*^b-*}
QS "L-d"                             {*^d-*}
QS "L-g"                             {*^g-*}
QS "L-px"                            {*^px-*}
QS "L-tx"                            {*^tx-*}
QS "L-cx"                            {*^cx-*}
QS "L-q"                             {*^q-*}
QS "L-f"                             {*^f-*}
QS "L-s"                             {*^s-*}
QS "L-sx"                            {*^sx-*}
QS "L-h"                             {*^h-*}
QS "L-xx"                            {*^xx-*}
QS "L-c"                             {*^c-*}
QS "L-j"                             {*^j-*}
QS "L-m"                             {*^m-*}
QS "L-n"                             {*^n-*}
QS "L-nx"                            {*^nx-*}
QS "L-l"                             {*^l-*}
QS "L-r"                             {*^r-*}
QS "L-y"                             {*^y-*}
QS "L-w"                             {*^w-*}
QS "L-v"                             {*^v-*}
QS "L-z"                             {*^z-*}
QS "L-zx"                            {*^zx-*}
QS "L-e"                             {*^e-*}
QS "L-u"                             {*^u-*}
QS "L-ii"                            {*^ii-*}
QS "L-a"                             {*^a-*}
QS "L-ie"                            {*^ie-*}
QS "L-ix"                            {*^ix-*}
QS "L-o"                             {*^o-*}
QS "L-pau"                           {*^pau-*}
QS "L-SIL"                           {*^SIL-*}
QS "L-h#"                            {*^h#-*}
QS "L-brth"                          {*^brth-*}
```

Declaration

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

Declared by:

Bereket Kasaye Tikui Date: Jun 5, 2009.

_____

Signature:

Confirmed by advisor:

Sebsibie H/Mariam Date: Jun 5, 2009.

_____

Signature:

Place and date of submission: Addis Ababa, October 2008.