ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

PHONEME LEVEL AUTOMATIC SPEECH SEGMENTATION FOR AMHARIC LANGUAGE USING HMM APPROACH

BY

ESHETE DERB EMIRU

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OFADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

NOVEMBER, 2011

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

PHONEME LEVEL AUTOMATIC SPEECH SEGMENTATION FOR AMHARIC LANGUAGE USING HMM APPROACH

BY

ESHETE DERB EMIRU

APPROVED BY

EXAMINING BOARD:

1. Sebsibe H/Mariam, Advisor (PhD) _____
2. _____ _____
3. _____ _____
4. _____ _____

Acknowledgement

I would like to thank my advisor, Dr. Sebsbie Hailmariam, for initiating the research idea, giving me critical and constructive comments throughout this thesis work. Dr. Sebsbie has also taught me how I can overcome difficulties and hardship. Thus, he is my role model to the rest of my life.

I would also like to thank Mr. Teshome Tefera for his continuous support and encouragement to finalize the thesis. My thank goes to my friends Abraham Wobie and Nirayo Haylu for their support during corpus preparation and epithetic vowel insertion algorithm development respectively. In addition to my friends, my thank goes to Mrs. Adina Berie for her endless support and patience during speech recording for experimentation. She has recorded properly all 1000 Amharic sentences collected from various sources.

There is no way I would be where I am now without the immeasurable love, support, and encouragement of my parents. I couldn‟t find terms to express their indispensable assistance throughout my life. Finally and most importantly, I would like to thank God who has blessed my work.

Table of Contents

List of Tables

List of Figures

List of Appendixes

List of Acronyms

ANN	Artificial Neural Networks
ASAM	Automatically Segmented data trained Acoustic Models
ASCII	American Standard Code for Information Exchange
ASR	Automatic Speech Recognition
C	Consonant
CART	Classification and Regression Tree
CC	Consonant Consonant
CV	Consonant Vowel
DARPA	Defense Advanced research Project Agency
DB	Data Base
EBNF	Extended Backus-Naur Form
G2P	Grapheme to Phoneme
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
LPC	Linear Prediction Coding
LSP	Line Spectral Pairs
MFC	Mel Frequency Coefficients
MFCC	Mel Frequency Cepstral Coefficient
MFCC- EDA	Mel Frequency Cepstral Coefficients- Energy Delta Acceleration
MLF	Master Label Files
NN	Neural Network
NSAM	Non Segmented data trained Acoustic Models
PSD	Power Spectral Density
RMS	Root Mean Squared
SLF	Standard Lattice Format
TTS	Text To Speech Synthesis
V	Vowel
ZCR	Zero Crossing Rate

Abstract

Speech segmentation is the process of identifying the boundaries between meaningful units like phonemes in a continuous speech. A need exists for reliable, automatic determination of phonemes boundaries in different speech research areas such as ASR to improve the performance of the recognizer, to improve the quality of speech synthesis system through segmented database, and to improve performance of language identification and speaker verification system.

In this study unsupervised method of automatic speech segmentation is proposed as a solution. Text corpus with size of 1000 Amharic sentences was collected from political news, economy news, sport news, health news, fictions, Bible, penal code and Federal Negarit Gazzeta. These Amharic texts are recorded by one female and one male speaker in order to have parallel speech corpus. Both text and speech corpuses are split into training (90%) and test (10%) data sets.

Phoneme based speaker dependent Hidden Markov Model is preferred, being the one that is most widely used, and also due to the availability of the HTK software suite. HMM approach is used to model Amharic phonemes in individual HMM with 3 emitting and 2 non emitting states without skipping left to right HMM. MFCC feature vectors together with their first and second derivatives are selected for individual HMM models.

Letter and phoneme were used as a basic unit to model the HMM in context independent, context dependent with single Gaussian mixture and context dependent with Multiple Gaussian mixtures. The system is also evaluated in terms of percentage of boundary deviations with 5ms, 10ms, 15ms and 20ms tolerance values with reference to manual segmentation results.

The evaluation of the experiments shows that best performance with minimum percentage of time boundary deviations are achieved using phoneme based approach in context dependent environment with two Gaussian mixture and four Gaussian mixture for male speaker and female speaker respectively.

Keywords: *phonemes, unsupervised method, Automatic speech segmentation, Hidden Markov Model.*

CHAPTER ONE

INTRODUCTION

1.1 Background

Speech segmentation is the process of identifying the boundaries between words, syllables, or phonemes in spoken natural languages. In speech segmentation, the basic idea of segmentation is to divide a continuous speech signal into smaller parts, where each of these segments has phonetic or acoustic properties that distinguishes it from neighboring segments. Speech segmentation involves dividing speech utterances into different chunks which are recognizable and meaningful. This includes segmentation at language level, accent level, sentence level, word level, pause or silence level and phoneme level[1]. Segments can also be thought of as patterns, each segment differing from total randomness in a coherent and perceivable manner[2].

In this study, speech segmentation at phoneme level is taken into consideration. A phone is a sound heard or articulated in actual speech, and as such it is a physical entity which can be measured and recorded by various devices. By contrast, a phoneme is what is perceived to be a particular phonetic entity, and thus by definition it is an abstraction, something like the common denominator of countless phones, namely actual sounds which share certain essential features. Even one and the same speaker and different speakers of a given language pronounces a given phoneme in numerous variations, which however are normally perceived as one phoneme, without creating any serious problem of communication[3]. Phonemes are also defined as the smallest unit that distinguishes a minimal word pair in a particular language. For example the minimal word pair bin [b iy n] vs. pin [p iy n] defines the units /b/ and /p/ to be phonemes in English[4].

The most commonly proposed phoneme level speech segmentations are using either manual segmentation or automatic segmentation techniques. In manual speech segmentation expert/phonetician is required and its segmentation is based on listening and visual judgment on required boundaries. However, manual phonetic segmentation is tedious, expensive, inconsistent,

prone to errors and time-consuming task[5]. There is also a disagreement between phoneticians and there is no clear, common and coherent strategies in order to segment speech waveforms [6]. Considering these and other disadvantages the development of automatic speech data is becoming increasingly important[7, 8].

Automatic speech segmentation is the partitioning of a continuous speech signal into discrete, non-overlapping units through algorithms developed for this purpose[9]. Generally, automatic speech segmentation methods are divided into two types, namely supervised and unsupervised segmentation methods. Supervised methods require *a priori* knowledge about phoneme boundaries [10,11,12]. These boundaries of phonemes exist in the form of their pre-segmented. It also requires pre-defined models of phoneme set of a specific language[13].

On the other hand, unsupervised methods don‟t require pre-defined model and knowledge about phoneme sets and their boundaries respectively. It is most commonly used in automatic speech segmentation through new modeling and training data sets[13]. Thus unsupervised method yields a desirable and more flexible framework for automatic segmentation of a speech at phoneme level[14].

Hidden Markov Model is the most commonly used model for automatic speech segmentation in unsupervised method[15,16,17,18]. This model can handle new data robustly within different working environments and enables to predict similar patterns efficiently[19]. It is also language independent and computationally efficient to develop and evaluate due to the existence of established algorithms[20,21,22]. Generally all the above benefits of both HMM model and unsupervised methods are the driving forces to conduct a research on unsupervised method of automatic speech segmentation using HMM model at phoneme level.

1.2 Motivation of thesis work

The task of speech segmentation is accomplished as part of preprocessing in various speech research areas like ASR, TTS, speaker verification, language identification and others. Segmentation results like phonemes are indispensable for the initial training of acoustic ASR models, the development of TTS systems and speech research in general. The experimental

results of these speech research areas are highly dependent on segmentation performance. In order to get the required segmented results, segmentation should be performed without any prior information about the speaker of the utterance in question. It should not rely on any type of prior learning, and should be able to process unknown utterances in a fully unsupervised manner.

There is no research conducted on automatic speech segmentation for Amharic language. This language has a large number of published and unpublished documents for training and these documents are helpful to get all phonetically balanced Amharic phoneme sets. Identifying the boundaries between phonemes in a continuous speech requires language model and acoustic model[13]. These models can be achieved through unsupervised method since there are no predefined acoustic model and language model for Amharic phoneme sets with their time boundaries.

1.3 Statement of the problem

Identifying the phoneme boundaries in a continuous speech is still a challenging task since it affects by co-articulation effect, the speaker accent and speaker speed. Even though these problems exist there is a need of phonemes with correctly identified boundaries in the following speech research areas mainly.

In ASR, the incorporation of information from phoneme boundary locations could potentially improve the recognition performance[16, 23]. The recognition improvement is achieved by segmenting the prerecorded speech signal and incorporating the phoneme boundary information into a phoneme recognizer.

In TTS, correctly segmented phonemes of automatic speech segmentation results are also very important to improve the quality of speech synthesis[24]. Speech segmentation is an essential task in speech synthesis where accuracy and consistency of segmentation are firmly connected to the quality of speech. For example, unit-selection speech synthesis requires a speech database built in segmented phonemes[21].

In language identification and speaker verification, correctly segmented phoneme boundaries also improve detection of speakers change and languages change[1, 25]. These phoneme boundaries

could be obtained through robust system to the dynamic change of acoustic signals and unsupervised method of automatic speech segmentation by developing language and acoustic models for a specific language.

There is also a need of correctly segmented Amharic phonemes with their time boundaries for speech research area such as ASR, TTS, speaker verification and language identification that can be potentially conducted for Amharic language. These correctly segmented Amharic phonemes with their time boundaries are vital to improve speech recognizer, speech synthesizer, and language and speaker detector results entirely as practiced in other foreign languages like English, French, Arabic, Mandarin, Dutch and Spanish. In order to get more improved results in Amharic speech research areas in terms of their performance and accuracy, automatic speech segmentation at phoneme level for Amharic language is required.

1.4 Objective

1.4.1 General objective

The general objective of this study is to design and develop phoneme level automatic speech segmenter for a given Amharic continuous speech.

1.4.2 Specific objective

The specific objectives of the research work are:

- To explore the phonetic variation of sentence units of Amharic language in speech segmentation.
- To build parallel corpus for Amharic sentences.
- To determine appropriate pronunciation dictionary for phoneme level Amharic speech segmentation.
- To determine the appropriate features for speech segmentation.
- To explore the Amharic phonetic variation of sentences that help in modeling Amharic

speech segmentation.

- To determine the appropriate acoustic modeling technique for phoneme level speech segmentation.
- To develop a prototype of the speech segmentation model of Amharic language.
- To evaluate Amharic speech segmentation model.

1.5 Scope and Limitation of the Study

The study would have been very interesting, if it compared all approaches, segmented in all levels (like word level, syllable and phoneme level) and used more number of speakers during speech recording. However, due to time constraint this study focused on speech segmentation for Amharic language using HMM approach at phoneme level.

1.6 Methodology

1.6.1 Review of related literatures

Literatures have been reviewed on different approaches and models. These literatures give an overview and basic understanding for every stages of the thesis work. Research conducted on automatic speech segmentation for various languages are also reviewed as part of related works. In these related works, the script of languages, the phonetics of the language, the features used for parameterization, the corpus size used, the models, algorithms, the number of states used to represent a phoneme and the experimental results are presented in detail.

1.6.2 Data collection

Optimal text selection technique is used to prepare Amharic text corpus from various Amharic document. These Amharic documents are used as data sources to get phonetically rich and balanced collections of sentences. Amharic bibles, health news, political news, sport news, economy news, penal code, federal Negarit Gazeta and Amharic fictions named as “Fiker eskemekaber” are data sources used for text corpus preparation. This text corpus contains 1000

Amharic sentences.

1.6.3 Design approaches

Two approaches are used during design of Amharic automatic speech segmentation system. These approaches are grapheme based approach and phoneme based approach and they differ in basic units of pronunciation dictionary preparation. Grapheme based approach uses letter sequences as pronunciation dictionary. This approach is the bench mark for our system and it is the approach followed in many Amharic speech recognition systems.

The second approach, phoneme based approach uses phoneme sequences as pronunciation dictionary. Since this pronunciation dictionary considers Amharic epithetic vowel /ix/ [ɨ], it is nearly phonetic.

1.6.4 Modeling

The most commonly and successfully used statistical approach for the task of explicit segmentation is Hidden Markov Models (HMMs)[26]. Hidden Markov Model is an extension of discrete Markov Model in which states are hidden, in the sense that an underlying stochastic process is not directly observable, but can only be observed through another set of stochastic processes. HMM models also support multiple pronunciation, language and acoustic models.

1.6.5 Tools

Software used in our study are Cygwin, payton, audacity, HTK, praat and Thomson Reuters EndNote. All these software are open sources and they are available online to use them for various applications. Cygwin software provides a Linux look and feel environment for Windows. Payton version 3.0 is used for automatic transcription of Amharic sentences to their corresponding Latin representations as per ASCII translation table attached in Appendix B. Audacity software is used for recording and editing sounds during speech corpus preparation. HTK toolkit is used for preprocessing of wave files, for development of HMM model, to train the developed HMM model and to get segmented letter/phoneme with their time boundaries. It consists of a set of library modules and tools available in C source code. Praat software is used for

manual labeling/segmentation during test data preparation. Thomson Reuters EndNote is used to generate references as IEEE format from already prepared reference database.

1.6.6 Evaluation

The evaluation of automatic speech segmentation results are carried out in terms of time boundary deviation with reference to manually segmented phonemes. Since the boundary deviations of automatically segmented phonemes are within 20ms tolerance values[12, 15, 27] and time boundaries of phonemes below 5ms tolerance values are not considered as errors[28, 29], evaluation is expressed in terms of percentage deviation in tolerance values of 5ms, 10ms, 15ms and 20ms and it is calculated using equation1[30].

$$\%\text{deviation} = \frac{\text{No of phonemes exceeding the tolerance value}}{\text{total number of boundaries tested}} \text{ x } 100 \qquad (1)$$

1.7 Application of results

Automatic speech segmentation is used to determine the boundaries of phonemes from continuous speech. Its results are very important to other speech based technologies like ASR, TTS, speech database, and language identification and speaker verification. Their benefits with respect to some speech based technologies are as follow:

For ASR: Segmented phonemes with their time boundaries are easier for automatic speech recognizer and they are useful to improve its performance[31]. In the context of stochastic approaches to speech recognition (ASR), accurate speech segmentation ensures appropriate initialization of model parameters during the bootstrapping process, so that the model parameters are close enough to the global maxima at initialization.

For TTS: Most successful speech synthesis systems today typically employ the use of segment concatenation of speech units like phonemes from input training speech corpora[32]. More natural speech synthesis is possible if effective segmentation can be performed to extract reliable synthesis units i.e. phonemes.

For speech database: Phoneme segmentation is used in building speech database for unit-

selection speech synthesis[21]. These databases are also helpful for other computational applications like speech based information retrieval in digital libraries systems by identifying the speakers as per the speech segment of them.

For language identification: Automatic segmented phonemes are also used for language tutor for children and handicaps through correctly identified phoneme boundaries[1]. The phoneme boundaries used to control them during tutorial.

For speaker verification: Speaker change can be detected through segmented phoneme boundaries of a speaker. Automatic segmented phonemes with their time boundaries are also applicable in telephone speech applications by identifying the speakers[25].

For pronunciation dictionary: correctly segmented phonemes are also useful to prepare a pronunciation dictionary at phoneme level. This pronunciation dictionary matches exactly with phonemes exist on their acoustic signals of a given continuous speech.

1.8 Organization of the Thesis

This thesis is organized into Seven Chapters including the current chapter. Chapter Two gives an overview of the literature review part of the thesis work. The literature review includes the general overview of the possible approaches or models used in the last two decades. The detail presentations of these approaches help us to select the appropriate model or approach for this thesis work.

Chapter Three gives an overview of the related works used in the thesis work based on HMM model and HTK toolkit. In the related work, the script of languages, phonetics of the language, the features used for parameterization and feature vectors, the corpus size used, models, techniques and the experimental results are presented in detail.

Chapter Four gives an overview of sound production system, and Amharic language writing system and its phonetics. In this section, consonant and vowel phonemes of Amharic language are identified and presented. The transcription of Amharic characters and their phonetic representation are also explained.

Chapter Five covers the overall explanation and design of the new Amharic speech segmentation system. This chapter includes HMM phonetic modeling, epithetic vowel insertion rules and algorithm, manual phonetic segmentation, and performance measurement techniques of the system.

Chapter Six presents experimentation and results of the thesis work. It includes the procedures of the experiment. Training and testing procedures are also given due attention. The results section presents in tabular form as per the result found in various tests. Finally, conclusions and recommendations for future work are presented in the seventh Chapter.

CHAPTER TWO

LITERATURE REVIEW

This chapter covers the overall concepts of speech segmentation. It also covers the most common speech segmentation approaches such as rule based, statistical (HMM) and Artificial Neural Network (ANN) approaches. During the last two decades various researches are conducted to identify phoneme boundaries using the above approaches and their contributions are reviewed in this literature part.

2.1 Speech segmentation

A speech signal is composed of several variables including the dialect of the speaker, style of speaking and language dependent information which in itself contains variables phonemic rules, phoneme inventory etc. Segmentation of speech signal at phoneme level has to take into account all of this information to obtain quality segment boundaries.

A phoneme is the smallest structural unit that distinguishes meaning in a language but phones refer to the instances of phonemes in the actual utterances - i.e. the physical segments. The wave-forms of a speech is segmented into its phoneme level as shown Figure 2.1, the speech of the string “my speech” [m a i s p ee t sh] segmented into /m/, /a/, /i/, / s/, / p/, /ee/, /t/, and /sh/.

Figure 2.1 Speech waveform of the string “my speech”

In addition to the physical observation of utterances, the speech waveform near a phoneme boundary is determined by the context. The same phone boundary might have different signal

characteristics for different contexts. Thus, for the boundary between any two phones *a* and *b* a simple *a* –end *b* -start kind of detector will not work very accurately. Also the signal features that change sharply near the boundary depend on both the phonemes[24].

2.2 Approaches of speech segmentation

2.1.1 Rule based approach

Rule-based systems are based on explicit representation of facts about language through well-understood knowledge representation schemes and associated algorithms. These rule-based systems usually consist of a set of rules which comes from different sources of knowledge. Knowledge is represented as facts or rules in the rule-based approach. The inference engine repeatedly selects a rule whose condition is satisfied and executes the rule. The primary source of evidence in rule-based systems comes from human-developed rules (like grammatical rules) and lexicons.

The rule based speech segmentation can be also based on acoustic characteristics of a speech like pitch, energy, zero crossing rate (ZCR), power spectral density (PSD), formant transitions, rhythm of consonants and vowels, fundamental frequency, duration, rate of speech, manner of articulation and place of articulation explicitly.

In[33], speech segmentation was exploited using the prior knowledge of glottal pulse locations for the estimation of adjacent broad phonemic class boundaries. The approach‟s validity was tested on the DARPA-TIMIT American-English language corpus and NOISEX-92 database. The performance was evaluated with an implicit approach used for the automatic detection of broad phonemic class boundaries from continuous speech signals in different additive noise environments.

In[1], Automatic Arabic Speech Segmentation System was done based on zero crossing rate (ZCR), power spectral density (PSD), formant transitions, rhythm of consonants and vowels, intonation pattern also called fundamental frequency and vowel duration. Pattern showed a typical shape in the start and the end which could provide information about specific consonants.

PSD and ZCR played a major role in segmenting phonemes. Spectrogram[1] and spectrograph[2] were very important in identifying PSD and ZCR respectively[1]. The spectrograph which is the wave representation of a cue helps in determining the ZCR by evaluating the frequency at which the zero line is crossed by the signal under consideration[1]. On the other hand, dotted lines in the spectrogram represent the formant trends as shown in Figure 2.2. They have a typical shape in the start and end which was helpful in broad segmentation of phonemes. The spectrogram with the energy bursts in vowel and consonant region which revealed that the bursts were very powerful in the vowel region as compared to the consonant region as shown in Figure 2.2.

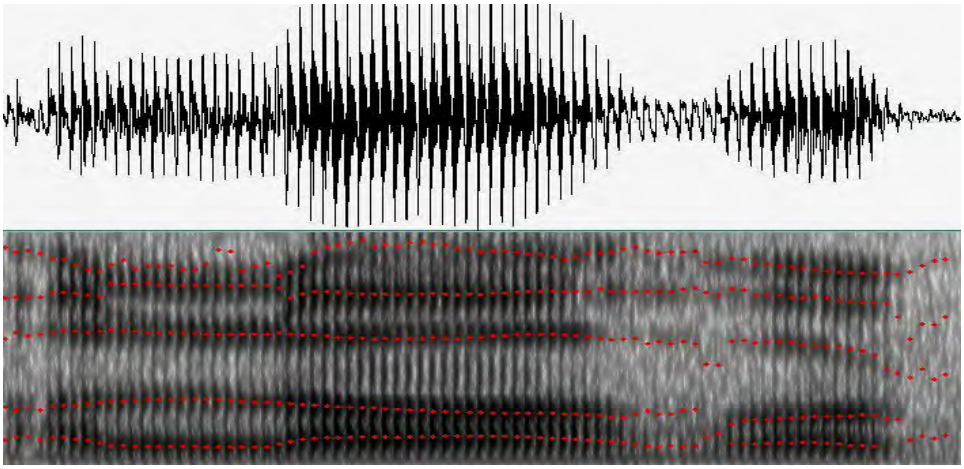

Figure 2.2 Spectrogram (below) and spectrograph (above) illustrating different cues.

In[1], a special algorithm was developed to accomplish segmentation for the Arabic speech based on energy level of the uttered words or sentences. In Arabic language, phonemes can be divided into two energy regions: unvoiced phonemes are categorized as low energy; vowels and semi-vowels are categorized as high energy.

As explained in[34], most of the methods rely heavily on a series of rules derived from acoustic phonetic knowledge. However, these rule-based methods are very complex and hard to optimize their parameters efficiently; the performances degrade severely in the real application. In order to

[1] *Spectrogram is formants of a cue to identify burst region of phonemes during segmentation.*

[2] *Spectrograph is wave representation of a cue to know the ZCR line crossed by the signal.*

overcome these drawbacks, other methods or approaches of phoneme segmentation are proposed.

2.1.2 Hidden Markov Model (HMM)

Statistical approaches employ various mathematical techniques and often use large text corpora to develop generalized models of linguistic phenomena. The model is based on actual phenomena provided by the text corpora without adding significant linguistic or world knowledge.

The most widely and successfully used statistical approach for the task of explicit speech segmentation is Hidden Markov Models (HMMs)[26]. Hidden Markov Model is an extension of discrete Markov model in which states are hidden, in the sense that an underlying stochastic process is not directly observable, but can only be observed through another set of stochastic processes. HMM is found a very powerful tool for various speech based application areas due to the following advantages:

1. It has strong statistical foundation and able to handle new data robustly[19].
2. It is computationally efficient to develop and evaluate (due to the existence of established algorithms)[19].
3. It enables to predict similar patterns efficiently[35].
4. It is language independent and does not assume complex linguistic knowledge such as duration of the phones during speech segmentation [20,21,22].
5. It can be used in different working environments/conditions such as speech segmentation under noise conditions[36].

Hidden Markov models (HMMs) provide a probabilistic technique for grouping observations of a process into states. An HMM state can be interpreted as a type of behavior exhibited by the process being modeled. The HMM represents the overall process behavior in terms of movement between states and describes the inherent variations of the observations within a state[37]. For each observation, the process being modeled occupies one of the HMM states. With each observation, the HMM either moves to another state or stays in the same state, based on a set of

state transition probabilities associated with the state. Thus, the state transition probability (A) distributions describe the underlying dynamic structure of the observations. The variety of the observations within a state is represented by the observation probability (B) distribution for that state, which may be either continuous or discrete. The HMMs discussed in this thesis work are continuous HMMs, where the observation sequence is a sequence of symbols drawn from a finite set of possible observations. With discrete HMMs, the continuous multivariate observations are mapped to a discrete set of observation symbols by a technique known as vector quantization [38].

HMM based automatic phonetic segmentation requires extensive training data even though it is useful to get very high degree of segmentation accuracy[39]. The standard HMM-based approach has been broadly used for automatic speech segmentation using a process called forced alignment [9]. It is a common practice to use context-independent HMMs for speech segmentation[40, 41]. Context-dependent HMMs can better model the spectral movements in phonetic transitions. However, the segmentations they produce tend to be less precise than the ones produced by context-independent HMMs[42].

The Hidden Markov Models are used to represent the sequence of sounds within a section of speech. Phoneme can be modeled by an individual HMM[43].The most common HMM architecture (topology) is the left-to-right (directional) HMM[44]. Standard left-right 3-emitting state HMMs with no skips over states is considered as a standard for phoneme segmentation[43, 45]. These standard HMMs are modeled for speech segmentation and each phoneme is represented by an individual HMM as shown in Figure 2.3.

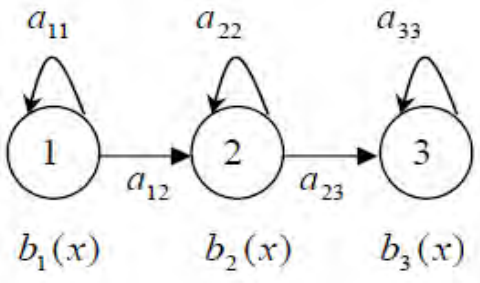

Figure 2.3 Representation of Left-to-right HMM

Where: 1,2 and 3 are states to represent a phoneme

a_{ij} is the transition probability from state i to state j and represented by the label on the edge from state i to state j

$a_{11,}$ a_{22} and a_{33} are self transition probabilities.

$b_i(x)$ is probability of observation x on state i

Moreover, the HMM architecture can be left-to-right 3 emitting and 2 non emitting states with skip in order to model every phonemes of a speech as shown in Figure 2.4 [42].

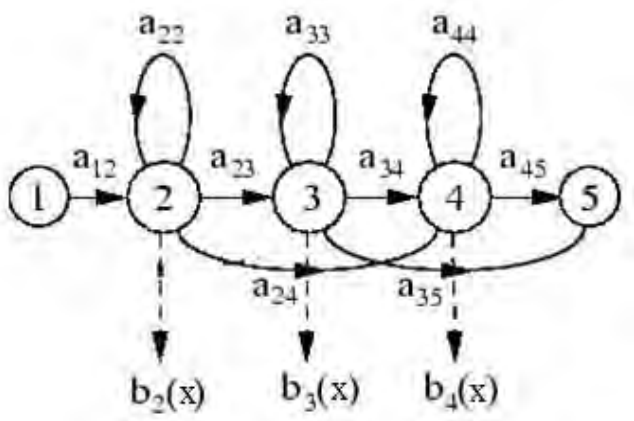

Figure 2.4 The 5-states HMM phoneme model.

Where: 1,2,3,4 and 5 are states to represent a phoneme

a_{ij} is the transition probability state i to state j and represented by the label on the edge from state i to state j

$a_{22,}$ a_{33} and a_{44} are self transition probabilities

$b_i(x)$ is probability of observation x on state i

An HMM λ model is described by three parameters:

A: transition probabilities defined for each speech unit

B: the observation probabilities defined for each state of the speech unit

Π: the initial state probabilities that define the probability of each state to be the 1^{st} state in a state sequence that define a given observation sequence.

Then $\lambda = (A, B, \pi)$

HMM-based systems perform Viterbi alignment on an existing acoustic model λ to determine the best state sequence q, that can be defined with the observation sequence O can be defined as[37]:

$$P(O, q \mid \lambda) = P(O \mid q, \lambda) P(q \mid \lambda) \qquad (1)$$

Where: $q = q_1\ q_2\ q_3\ \ldots . q_N$

$O = o_1\ o_2\ o_3\ \ldots o_N$

N is the number of observations in O

The solution to the decoding problem of HMM, which deals with finding the best state sequence *q* that best aligned with the observation sequence *O* given an HMM model λ[37], addresses the problem of automatic speech segmentation.

One of the methods used to automatically segment speech at the phone level is to force-align phone sequence on an existing acoustic model λ. The frames at the transitions from one phone to the other become the boundaries. In order to solve practical problems with HMMs, three basic problems have been identified[37].

The first deals with how to efficiently compute $P(O \mid \lambda)$, the probability of the observation sequence given the model parameters. The solution is either the forward or backward algorithms.

Furthermore, they are used for the computation of the probabilities of partial observations used in Baum-Welch algorithm for the training of model parameters. The forward variable α in equation 2 is the probability of the first *n* observations and being in state *i* given the model λ[37]. The backward variable β in equation 3 is the probability of the last $N - n$ observations given that the system was in state *i* at the *nth* observation given the model λ[37] . The probability of observing the 1st n observation and being at state i is given by equation 4[37].

$$\alpha_n(i) = P(o_1\ o_2 \cdots o_n, q_n = i|\lambda) \quad (2)$$

$$\beta_n(i) = P(O_{n+1}O_{n+2}\ldots O_N | q_n = i, \lambda) \quad (3)$$

$$P(O, q_n = i|\lambda) = \alpha_n(i)\beta_n(i) \quad (4)$$

The second problem deals with how to efficiently compute $P(O,q|\lambda)$, the joint probability of the observation O and the best state sequence q given the model λ, where $q \in Q$ and Q is the set of all possible state sequences. The solution to this is the Viterbi algorithm. The best state sequence through an HMM automatically gives segmentation because the frames at the transitions from one unit to the other become the boundary frames. This is one of the advantages of HMMs over other methods that require prior segmentation of speech.

The third problem is about adjusting the model parameter λ to locally maximize $P(O|\lambda)$, but unlike the previous two, there is no known closed form solution. However, Baum-Welch algorithm is used and to converge to local maxima after a finite number of iterations.

2.1.3 Artificial Neural Network (ANN)

A neural network (NN) uses a mathematical or computational model for information processing based on a connectionist approach to computation. Neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. However, the paradigm of neural networks is implicit; its learning more corresponds to some kind of natural intelligence than traditional artificial intelligence which is common in rule-based learning.

The most common type of artificial neural network consists of three or four groups or layers of units which are first/input layer, one or two intermediate/hidden layers and an output layer. A layer of input units is connected to a layer of hidden units which is connected to a layer of output units as shown in Figure 2.5.

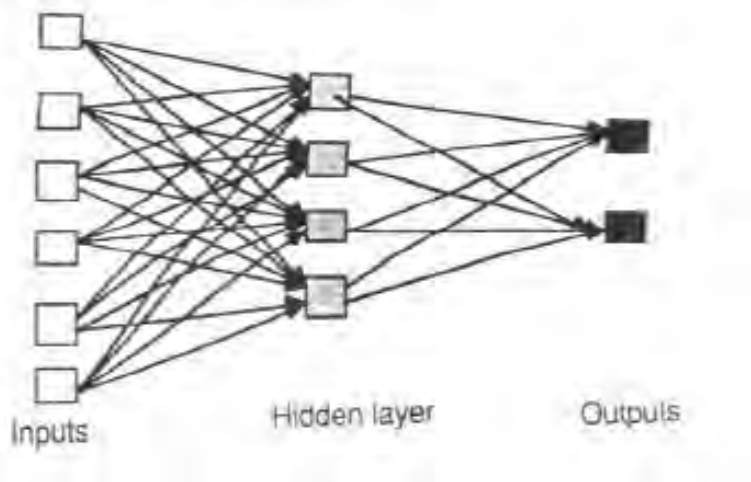

Figure 2.5 An example of a simple feed forward network

The activity of the input units represents information like features pitch and duration[46, 47] that is fed into the network. These features represent a phoneme that is to be segmented automatically. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units. Finally, the weighted results are classified into some group of features and represent phoneme classes.

Simple type of network like Figure 2.5 is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

Neural networks have a growing number of applications in audio, including speech recognition, classification, discrimination, and more artistic applications, such as composing, or finishing Bach‟s last fugue[48] but neural networks are often used for pattern recognition[49]. This indicates us to choose other best appropriate approaches/models like HMM in order to achieve high quality segmented phonemes.

CHAPTER THREE

RELATED WORKS

This chapter covers related works done on speech segmentation using Hidden Markov Model and HTK toolkit in various languages. It helps us to have a clear view on this thesis work and covers nature of the language script, the features used, corpus size, the number of HMM states, techniques and algorithms used in the HMM model, and the results of their experiment.

3.1 Speech segmentation for Tamil Language

Speech segmentation on Tamil language was introduced and built on a Tamil voice using HMM model for speech synthesis application[50]. Tamil is one of the Indian languages with fewer characters than other Indian languages. In Tamil there are 13 vowels and 18 consonants which are very small in number as compared to other Indian languages. During speech to text system pronunciation variations of some consonants are considered and in effect there are 41 phones. Letter to sound rules for Tamil are built either using rule-based or machine learning algorithms such as CART [22, 51].

The corpus size used in the system was 0.3 million sentences of the Tamil text from the news portal. A small set of sentences was selected in order to build a unit selection voice. This corpus has 2.7 million words and 4302 syllables. By using a greedy approach, 2394 sentences were selected which cover 25769 words and 2392 high frequency syllables.

The selected sentences were recorded by a female native Tamil speaker in a recording studio. The speaker uttered the sentences into a stand mounted microphone placed infront of her. The speech data was recorded at 44 KHz, mono channel at 16 bits per sample. After the recording, it was down sampled to 16 KHz for further processing.

MFCCs, MFCCs + deltas and MFCCs + delta + delta-delta were feature vectors used in speech segmentation. HMMs based approach was used. Two experiments were done in order to get the proper speech segmentation results. In both experiments, the segmented labels of their results

were compared with manually segmented labels in the context of Hindi database since Tamil language is a family of Hindi language and their main difference is in their structure only.

In the first experiment, speech segmentation experiment with SPHINX and HTK was done for Hindi database. The results of the performance of Sphinx-II and HTK were similar for segmentation purposes. However, there was an average difference of 2 ms between Sphinx-II and HTK, which could be attributed to semi-continuous models or context-dependent models used in Sphinx-II. The experimental results in terms of average deviation were 26.0 ms and 29.59 ms using Sphinix-II and HTK tools respectively. These results were evaluated in terms of boundary deviations with reference to manually segmented labels since manually segmented labels are considered as correct results. These results also indicate that the delta and delta-delta coefficients may not contribute to lessen the average deviation. Typical computation of delta coefficients was done by smoothing the difference parameters and hence these coefficients could be more relevant to speech recognition than to speech segmentation.

In the second experiment, speech segmentation with different parameters experiment was proposed with context-independent models and with written HMM code in order to get delta coefficients which are more relevant to speech segmentation. The use of this written HMM code also served the purpose of comparing its efficiency and performance with standard tool such as HTK and Sphinx-II. In HMM code two Gaussians per states and a frame rate of 5 ms was used. An average deviation of context independent monophones was reduced to 24.11 ms with reference to manual segmented Hindi database. This result showed that context-independent models may be useful and relevant for speech segmentation in the context of speech synthesis as they avoid computation time and resources required to build context dependent state models.

3.2 Speech segmentation for Maltese Language

Building an Automatic Speech Annotation System was conducted on a Maltese language[52]**.** Maltese is the only Semitic language written in the Latin alphabet. Speech annotation was the task of partitioning a speech signal into basic speech sounds of a language known as phonemes. It was a process of locating the phoneme boundaries as precisely as possible from a priori sequence of phonemes. For this purpose, two automated methods were proposed.

The first method was using the English-American TIMIT database. HTK toolkit and 23 utterances from TIMIT were used for the training phase of the phoneme models. The topology used was a continuous-density Gaussian 5-state HMM. A sequence of parameterized feature vectors was extracted from the utterances. The system made use of MFCC using Energy, Delta, Acceleration and Cepstral Mean Normalization. Each analysis made use of a 25 ms Hamming window with a 10 ms overlap. The training of all the phonemes was carried out by a system based on HTK toolkit.

For the purpose of system evaluation, it was required to map the set of TIMIT phonemes into the equivalent Maltese phonemes. In practice, some Maltese phonemes used the same TIMIT phonemes while the rest made use of the nearest equivalent. System evaluation was performed by a set of Maltese utterances. Phoneme segmentation found to be within tolerance, which for this experiment measured 17.2 ms, was marked as correct. System performance was not so satisfactory because only 73% of the segmentation were within tolerance range. However, these were only preliminary results on which further research was developed.

The second method was using manually segmented Maltese utterances tailor-made for this Maltese speech segmentation system this second method was based on a speaker-dependent system utilizing a continuous-density Gaussian, 5-state, left-right HMM with mixture topology. To build the Maltese speech database, 100 Maltese utterances were recorded by one speaker at a professional studio. By utilizing HTK toolkit and Cool Edit, all the utterances were manually labeled so that for each phoneme there was an associated start and end timing.

For the training of HMMs, 70 of these utterances were used while the other 30 were used for the evaluation phase. For the training of HMMs, the 70 training speech waveforms were parameterized into MFCC coefficients, using Energy, Delta, Acceleration and Cepstral Mean coefficients. A 39-element acoustic vector was extracted from each frame. The source waveforms were sampled at 16000 Hz, utilizing a 25ms Hamming window and a frame period of 10 msec.

The second set of 30 Maltese utterances was used for system evaluation. These utterances were parameterized into a series of feature vectors similar to the training utterances. Automatic

Annotation was performed by the HTK HVite tool in forced alignment mode. The annotation process will then attach the appropriate HMM definition to each phoneme instance. The decoder, by applying the Viterbi algorithm, then selects the optimal path through the network which leads to the best matching pronunciations.

Due to the fact that the annotation system was utilizing a frame of 25 ms duration, at a period of 10 ms, and the location of phoneme boundaries was frame dependent. It was established that label markings within the 17.2 ms tolerance were correct. System performance was established by comparing the annotated phoneme label markings with the manually marked ones. Detection accuracy was worked out by the formula shown below and the boundary detection accuracy reached the 88.11% mark.

3.3 Speech segmentation for Mandarin Language

Mandarin is the main language of government, the media and education in China and Taiwan. Syllable Boundaries based Speech Segmentation in Demi-Syllable Level for Mandarin was done using HTK and its result can be used directly for application of Text To Speech (TTS) system[15] **.** It was easy to acquire the syllable boundaries in much high accuracy with the help of pitch contours, energy contours, zero-across rate contours.

The segmentation system offered an optimized method in speech segmentation of Mandarin speech database by using Hidden Markov Model (HMM). It also analyzed that the influence of the amount of HMM states and the amount of the training corpus. Normally the size of the corpus used for speech synthesis or speech recognition was extremely large.

Mandarin is a tonal language, and the classical minimum speech elements of Mandarin are demi-syllables, which are initials and finals. The initials contain plosives, fricatives and nasals, and the finals contain single-vowels and compound-vowels. The spectrums of compound-vowels are more complicated than the phonemes.

To make sure, the number of HMM states was essentially important for Mandarin speech segmentation, the various numbers of HMM states were tested. It was found that the accuracy of speech segmentation was sensitive to the amount of HMM state too much and it was proved that

increased number of HMM state can improve the final results of speech segmentation a little, especially when the state number of compound-vowel was increased. Nevertheless, good context coverage and consistent segmentation by HMMs can partly overcomes the drawback of an imperfect automatic segmentation when compared to manual segmentation. To acquire the balance of computing cost and good results, the speech segmentation used 3, 3, and 5 as the state number of initials, single-vowels and compound-vowels respectively.

Generally, the system described a new method to segment the speech in demi-syllable level for Mandarin with HTK, which was based on syllable boundaries. The speech segmentation accuracy improved from 88% to 95% with 20 ms tolerance compared to hand-labeled corpus. The testing result also showed that the accuracy of the boundaries between plosive and vowel or vowel and vowel was also improved from 81% to 92%.

3.4 Speech segmentation for French Language

Automatic Phoneme Segmentation system done on French speech for speech synthesis application[53]. Speech synthesis by unit selection requires the segmentation of a large single speaker high quality recording. Automatic speech recognition techniques like Hidden Markov Models (HMM) can be optimized for maximum segmentation accuracy.

The recorded text was a set of 3994 sentences in French, chosen in[54] for good phonetic and contextual covering. It was read by a male French speaker in anechoic room and recorded with a high quality microphone and a 16 bits 44.1 kHz analog to digital converter.

The acoustic features used in all experiments are Mel-Frequency Cepstral Coefficients (MFCC) together with their first and second smoothed time difference features were calculated on 25 ms with frame rate of frame size 5ms.

Optimal HMM architecture and parameter values had been determined for a high quality mono speaker recording. An automatic speech recognition technique in HMM model was used to get maximum segmentation accuracy. The segmentation system presented was based on the Hidden Markov Models Toolkit (HTK) [55]. It had been designed to perform a Viterbi decoding based on a phoneme bigram language model when the text transcription was unknown, or to make use

of the textual information modeled by a multi-pronunciation phonetic graph when the text was at least approximately known.

Firstly, using no text transcription, the design of an HMM phoneme recognizer was optimized subject to a phoneme bigram language model. Optimal performance was obtained with triphone models, 7 states per phoneme and 5 Gaussian mixtures per state, reaching 94.4% phoneme recognition accuracy with 95.2% of phoneme boundaries within 70 ms of hand labeled boundaries.

Secondly, using the textual information modeled by a multi-pronunciation phonetic graph built according to errors found in the first step, the reported phoneme recognition accuracy increases to 96.8% with 96.1% of phoneme boundaries within 70 ms of hand labeled boundaries. Finally, the results from these two segmentation methods based on different phonetic graphs, the evaluation set, the hand labeling and the test procedures were discussed and possible improvements were proposed.

3.5 Speech segmentation for Arabic Language

Automatic Segmentation System was done on Arabic language for Speech Recognition Application[16]. The speech database used in the training consists of 10 hours of recorded speech collected from 100 speakers. These speech data were recorded at 16 KHz using a microphone connected to a desktop PC equipped with sound card in a silent room. A software tool (Validator) also developed to collect the spoken utterances and labeling them.

First the utterances were automatically transcribed using the orthographic transcription as an input and using an automatic transcription engine specially designed for this task. The designed transcription engine used to apply a set of letter to phoneme rules (also known as grapheme to phoneme or G2P rules)[56].

The automatic segmentation system was used to label a speech database system that was used in the training of continuous, phoneme based speaker independent Hidden Markov Models. The training of the acoustic model was based on HTK version 3.1 training tools. The system was trained with different sets of acoustic models. It also used Mel scale cepstral coefficients with

appending the 0 th cepstral parameter Co plus their first and second derivatives (MFCC_0_D_A). The acoustic modeling used cross-word tri-phones Hidden Markov Models trained using conventional maximum likelihood estimation.

The HTK's tool HVite was used in forced alignment mode to perform automatic segmentation of speech corpus. The system was evaluated for automatic speech segmentation accuracy in two ways. Firstly, the evaluation was carried out with reference to manual segmented phonemes. Comparison of automatic segmentation to manual segmentation and the deviation is measured in 4 utterances. The lengths of the 4 sentences were 45, 42, 56,115 phonemes making a total of 258 phonemes. The evaluation was done by counting the number of boundaries for which the deviation between automatic and manual deviations exceeded tolerance thresholds of 35, 70 and 100 ms[57]. The experiments showed that 7.8 % of the phoneme boundaries of automatic segmented data deviate from those that were manually segmented more than 30 milliseconds while 0.78 % deviates more than 70 milliseconds.

Secondly, the evaluation was carried out with reference to non segmented data trained acoustic model (ASAM). In this evaluation method, automatically segmented data trained acoustic models (ASAM) were trained using the automatic phoneme segmented data and exactly the same training techniques that were used in training NSAM models. The impact of using automatic segmentation on the improvement of speech recognition accuracy was measured by comparing ASAM recognition accuracy with NSAM models. The results showed clearly a significant improvement between training using automatically segmented (labeled) phonemes speech data (ASAM) and the models that are trained using (NSAM).

In general, the experiments showed that automatic segmentation led to improvement in speech recognition accuracy of 0.49 % for a 306 words bigram language model test and 0.14% for 1340 words bigram model test.

CHAPTER FOUR

AMHARIC LANGUAGE AND ITS PHONETICS

This chapter presents the origin of Amharic writing system and how to transcribe to its phoneme sound representation. The general sound production system with especial reference to Amharic language is also discussed. The third part is about phonetics which refers to the study of speech sounds used in the language. Phonetics is concerned with the sounds of the language, how these sounds are articulated and how the listener perceives them.

4.1 Amharic writing system

According to Bender et al. [58], three writing systems are in use in Ethiopia, the Ethiopic (Ge‟ez) syllabary, the Roman alphabet, and Arabic script. The widely used is syllabary which is derived from the writing system of ancient South Arabian inscriptions and it is used for Ge‟ez, Amharic, Tigrigna and other semantic languages. The writing system has a similarity with some Semitic languages like Arabic in having vowel marks added to basically consonant letters. The present writing system of Amharic is taken from Ge‟ez. Ge‟ez in turn took its script from the South Arabian mainly attested in inscriptions in the Sabean dialect[58]. The original Sabaean alphabet is said to have had 29 symbols. When Ge‟ez became the spoken and written language in common uses in northern Ethiopia, it took only 24 of the 29 Sabaean symbols, modified most of them and added two new symbols to represent sounds of Greek and Latin loan from words not found in Ge‟ez, these symbols are ጰ and ፐ. The style of the writing was also modified from left to right. By the time Ge‟ez ceased to be a living spoken and written language and replaced by Amharic and other languages, further changes took place. Amharic did not discriminate in adopting the Ge‟ez fidel; it took all of the symbols[59] and added some new ones that represent sounds not found in Ge‟ez. These added alphabetic characters are ቸ, ጨ, ጀ, ኘ, ቨ, ሸ, ኸ, and ዠ.

Currently, the language‟s writing system contains 34 base characters each of which occurs in a basic form and six other forms known as orders (Appendix A)[58]. The seven orders represent syllable combinations consisting of a consonant following vowel as shown in Table 4.1. Out of the seven derivatives six of them are CV (Consonant vowel) combinations while the sixth is the

consonant itself[60]. Other symbols representing labialization, numerals, and punctuation marks are also available. This is why the Amharic writing system is often called syllabic rather than alphabetic, even if there is some opposition[61]. The 34 basic characters and their orders give 239 distinct symbols. In addition, there are forty others that contain a special feature usually representing labialization e.g. ቯ, ቈ. In Amharic there is no Capital-Lower case distinction.

4.1.1 Transcription System

The script of Amharic language is phonetic in nature. In order to represent Amharic characters to their corresponding sounds, ASCII transliteration system is used throughout this thesis work (attached in Appendix B)[62] and this transliteration is accomplished automatically with Python code developed for this purpose which takes ASCII translation table as input (attached in Appendix C). This transliteration scheme is designed based on the orthographic ordering of the script and the acoustic similarity of the letters. The translator also normalized to some common sounds since in Amharic language there are some characters which have different orthography but the sound is the same like ሀ, ሐ,ኀ, ኃ, ሃ and ሰ ,ሠ. Table 4.1 shows, the ASCII transcription of three consonants with their 7 orders.

Table 4.1 The Amharic consonants with the vowels (using ASCII translation)

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

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

4.2 Amharic Phonetics

Humans can produce an infinite number of sounds; each language has a set of abstract linguistic units, called phonemes, to describe its sounds. Amharic language is primarily comprised of 39 phonemes – 7 vowels and 31 consonants[63]. One additional consonant /ቨ/ [v] is inherited and included summing up to a total of 39 phonemes. These phonemes are categorized into vowels and consonants.

4.2.1 Vowels

Vowels have different categories based on the position and height of the tongue and their shapes during speech production. Based on the tongue position in the oral cavity, vowels are classified into three sub categories that are front, central and back[64]. Based on the height of the tongue, these vowels are also classified into high, middle and low. Based on their shapes during speech production, vowels are classified into two sub classes that are rounded and unrounded. The vowel classifications are indicated in Figure 4.1 i.e. rows represent vowels classification based on the position of the tongue and columns represent vowels classification based on the height of the tongue. In addition to these classifications, the right represents a rounded vowel (in which the lips are rounded) while the left is its unrounded counterpart. The central vowels are also considered to be unrounded.

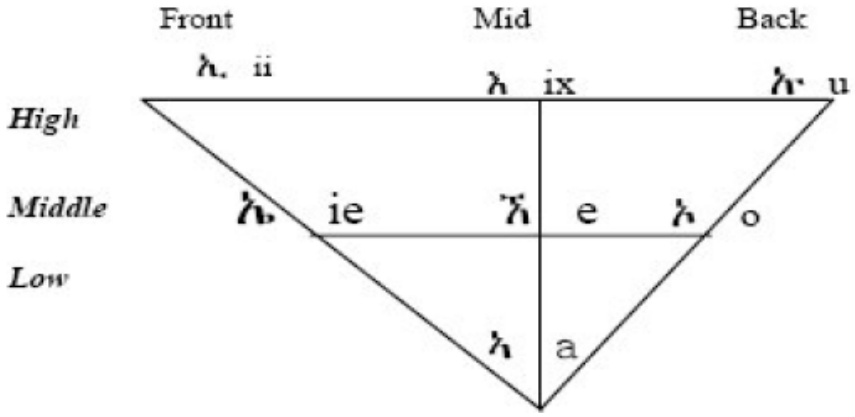

Figure 4.1 IPA maps of the Amharic vowels[60]

4.2.2 Consonants

Consonant phonemes are described in the following categories: voiced vs. unvoiced; manner of articulation; place of articulation as shown in Table 4.2. The place of articulation means where the constriction is located in the vocal tract. Based on manner of articulation, consonants are categorized into stops, fricatives, affricatives, nasals, liquids and semivowels[65].

Table 4.2 Phonetic representation and characterization of Amharic consonants (adopted from[62]).

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

CHAPTER FIVE

DESIGN OF AMHARIC AUTOMATIC SPEECH SEGMENTATION ARCHITECTURE

5.1 Introduction

This chapter covers design approaches used, design goals and automatic speech segmentation system for Amharic language. The design approaches, namely grapheme based and phoneme based approaches are used to develop automatic speech segmentation system in order to meet the objective of thesis work. In addition to these approaches, two techniques of speech segmentation are also discussed. These techniques are automatic speech segmentation and manual labeling of phonemes with their time boundaries.

In automatic speech segmentation system, data preparation, language modeling, HMM acoustic modeling and speech segmentation are discussed in detail as components. Designing automatic speech segmentation in a single model becomes complex in terms of its diagrammatic representation and it is not easy to understand the flow of processes because of these reasons splitting into its components is required. Since data preparation is the indispensable parts of automatic speech segmentation system, the ways of data collection, text and speech corpus preparation, pronunciation dictionary preparation, sampling technique and feature extraction process are presented under it.

In addition to the automatic speech segmentation, manual labeling technique used to prepare test data sets is also discussed since its results are used as references to give analysis on the performance of automatic speech segmentation system. At the end, the result verification method used to measure the performance of the automatic speech segmentation system with reference to manual segmentation results is also presented in detail.

5.2 Design Goals

The goal of designing automatic speech segmentation system for Amharic language is to get correctly segmented Amharic phonemes with their time boundaries by developing appropriate pronunciation dictionary, language model and acoustic models for them. The design of automatic speech segmentation is, therefore, to achieve a better performance segmented results of Amharic language units known as phonemes with minimum percentage of time boundary deviation in reference to manually labeled phonemes.

5.3 Automatic Speech Segmentation Architecture

The general model of automatic speech segmentation system includes data preparation, manual labeling, language modeling, HMM model building, HMM segmenter and data verification components. All components are included in both grapheme and phoneme based approaches which can be implemented individually as phase1 and phase2 respectively during experimentation. Including all components in a single model or architecture is complex and difficult to understand therefore the models of each component are given separately and integrated by taking the output of the previous component as input to the next component. The descriptions of components are also given in the following sub-sections.

5.3.1 Data Preparation

Data preparation is the main part of automatic speech segmentation system since it has high contribution to the performance of automatic speech segmenter. It includes core processes like corpus preparation which encompasses both text and speech corpuses, lexicon preparation as pronunciation dictionary, sampling techniques used to split both text and speech corpuses into test and training sets and feature extraction process to prepare speech corpus usable format to the HMM acoustic modeling and HMM segmenter as input. The overall data preparation of automatic speech segmentation system is as shown in Figure 5.1.

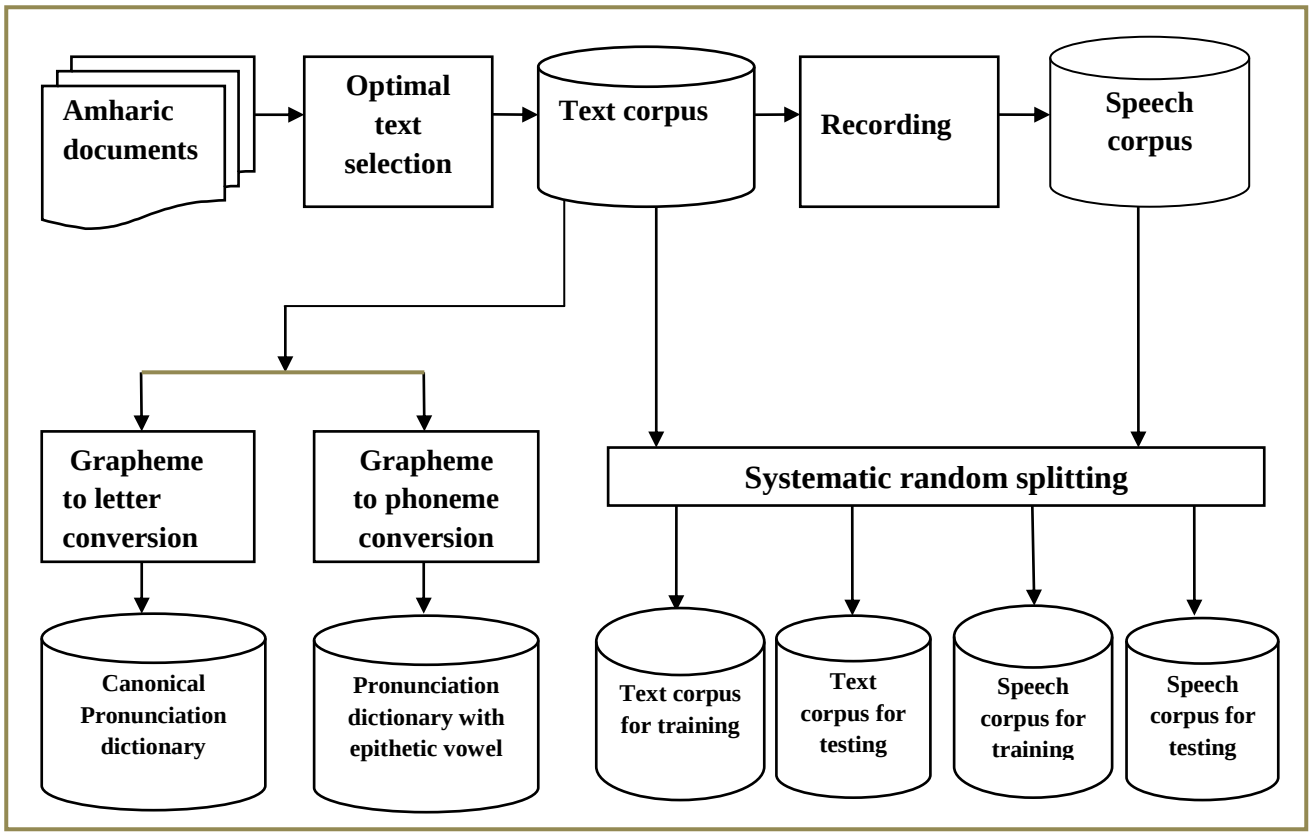

Figure 5.1 Data preparation components of automatic speech segmentation

5.3.1.1 Data collection and Corpus preparation

Data collection is the primary process of data preparation during automatic speech segmentation system. The selection of sentences from various text sources aims at both a phonetically rich and balanced collection of sentences. Optimal text selection technique is used to take sentences from various Amharic documents. Selection of Amharic sentences from eight different Amharic sources (political news, economy news, sport news, health news, fictions, Bible, penal code and Federal Negarit Gazzeta) is required. On the text corpus, problems like spelling and grammar errors are corrected, abbreviations are expanded and numbers are textually transcribed. To avoid elongated sentences that create difficulty for the readers, sentences with a maximum of twenty words in length are chosen from the available text sources. In over all, the text corpus covers all Amharic phonemes to get phonetically balanced results during speech segmentation.

Having the text corpus, the next important step in corpus preparation is recording the speech in order to get speech corpus of the corresponding text corpus. In the recording of the selected sentences, the two speakers read exactly what is presented to them. Speaker invariants like gender, age accent are also taken into consideration during speaker selection. After speaker selection is completed, the text to be recorded is printed on paper. To avoid sound interferences, the recording process of speech corpus is carried out in a quiet environment.

The prepared Amharic sentences are recorded to create speech corpus using the software called Audacity with the sample frequency of 48 kHz using a Mono channel. Each of the training utterances is recorded by two speakers. The recorded files are saved in the *.wav format. Even though the recording process was carried out in a quiet environment, it was not 100% free from the surrounding noise. Sound interference can be treated as negligible since both training and testing data are recorded in the same environment as the noise affects both data sets equally.

5.3.1.2 Pronunciation Dictionary Preparation

Lexicons are prepared from text corpus and used as pronunciation dictionary. Since pronunciation dictionary for Amharic language is not built yet, preparing pronunciation dictionary plays a major role in the performance of automatic speech segmentation. Two methods are proposed to build the pronunciation dictionary:

5.3.1.2.1 Letter based pronunciation dictionary

Grapheme based speech segmentation approach is direct implementation of the transcribed words found in ASCII transliteration system. The segmentation results are achieved through letter based pronunciation dictionary. This pronunciation dictionary is developed by direct transliteration of Amharic words to their corresponding Latin alphabet/letters representations. These words contain letter sequences as pronunciation dictionary. This pronunciation dictionary is also known as *canonical pronunciation dictionary*. For each word, a canonical pronunciation dictionary includes only the grapheme transliteration sequences and assumed that the sequences are pronounced in read speech. It does not consider pronunciation variations such as speaker variability, dialect, or co-articulation in conversational speech. The automatic speech

segmentation is carried out through these letter sequences as pronunciations dictionary and it is common in Amharic speech research areas like speech recognition.

5.3.1.2.2 Phoneme based pronunciation dictionary

Grapheme based speech segmentation approach uses the second method of preparing pronunciation dictionary i.e. phoneme based method. Phoneme based pronunciation dictionary is built in direct transliteration of Amharic words to their Latin representation like letter based method and Amharic epithetic vowel insertion algorithm. This pronunciation dictionary contains phoneme sequences as pronunciation and it is nearly phonetic.

In order to get phoneme sequences as pronunciation dictionary developing grapheme to phoneme converter is required. Nirayo Hailu[66] developed epithetic vowel insertion algorithm to syllabify Amharic words and he has got 98.1% accuracy using rule based syllabification approach. By adopting his algorithm with some modification in case of germination of consonants; phoneme based pronunciation dictionary is developed as attached in Appendix F. The main inputs to develop epithetic vowel insertion algorithm are the concept of epenthesis in Amharic words, sonority scale of phonemes, epithetic vowel insertion rules.

Epenthesis in Amharic words: The process of epenthesis is common in Amharic. It can occur word-initially and medially. Hudson[67] states that epenthesis is extensive in word-formation in the Ethiopian Semitic languages since many morphemes, both roots and affixes, consist only consonants. Amharic epenthesis vowel may be said to provide almost all occurrences of the high central vowel /**ix/** [ɨ].

Sonority scale of phonemes: In addition to rules of epenthesis in Amharic words, assigning sonority scale for each Amharic consonant is required. Sonority scale is almost universal for all languages or it is language independent[68, 69]. Therefore, as in state-of-the-art systems, grouping of phonemes into classes and assigning sonority scale for each class have been done and the sonority scale assigned for each class is attached in Appendix D. It is indicated that stops have got the least sonority scale (number) and glides are more sonorous.

Epithetic vowel insertion rules: By summarizing epenthesis vowel insertions of Amharic language discussed by Hudson [67] and Mulugeta[70], six main epenthesis vowel insertions are developed in order to build the epenthesis vowel insertion algorithm. These six epenthesis vowel insertion rules are presented in Appendix E.

Epithetic vowel insertion algorithm (Appendix F)**:** Based on the summary of epithetic vowel insertion rules, epithetic vowel insertion algorithm is developed and implemented using Microsoft Visual Basic C#. Epithetic insertion process takes place by plug-in this implementation to our automatic speech segmentation system in order to prepare the phoneme based pronunciation dictionary.

5.3.1.3 Sampling

To be free from bias during data selection, systematic random sampling technique is used for splitting both text corpus and speech corpuses into two, namely training and testing sets. It implies that data are arranged in ten columns and ten rows and randomly a single column is taken as test data sets. The training set consists of 900 Amharic texts of text corpus with their corresponding speech corpuses. The rest of 100 Amharic texts with their corresponding speech corpuses are test data sets of the automatic speech segmentation system. The end results of systematic random sampling technique are two text corpuses and two speech corpus. It implies that training data sets and test data sets of both corpuses are obtained. The training data sets are used for building an HMM model where as test data sets are used as inputs to the HMM segmenter as far as the models are built by the training data sets.

5.3.1.4 Feature extraction

Digital signal processing techniques are applied to convert analogue wave form into a digital signal wave form. Since the speeches what we have recorded are continuous or analog; they have to be converted into discrete or digital by sampling technique and quantization of the wave form. On these samples of speech wave form, feature extraction process takes place for preprocessing of the input speech signal. This feature extraction is the process of spectral parameter extraction (Parameterization) which involves conversion of speech samples into feature vectors to provide

spectral patterns of the speech. Feature extraction process usually assumes that the characteristics of the speech signal are stationary over a short time period, typically of the order of 25 milliseconds.

MFCCs are widely used feature vectors in speech segmentation and they provide a compact representation of log-spectral envelope of speech signals[16]. These feature vectors are also used in this new automatic speech segmentation system together with their first and second derivatives. The first and second derivatives of MFCC are included to make the model sensitive to the dynamic behavior of the signal[5].

The feature vector that represents the distinctive properties of the phoneme is designed to be of length **39**, consisting of **12** mel-cepstrum coefficients and energy component, and additionally their delta and acceleration coefficients. The structure of the feature vector is as shown in Figure 5.2.

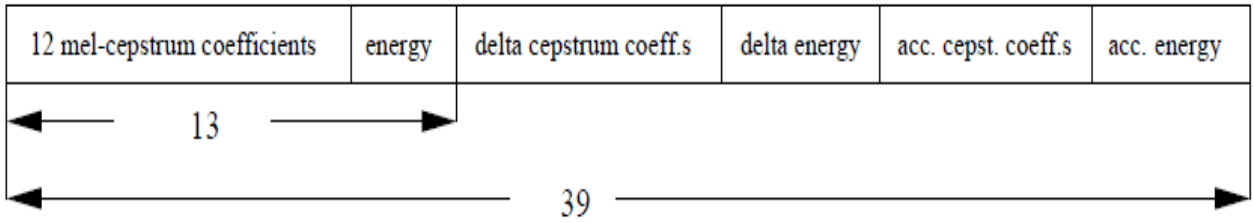

Figure 5.2 The structure of the feature vectors

5.3.2 Manual Labeling

Manual labeling is required to verify the automatic speech segmentation system. It is carried out before the completion of automatic speech segmentation to be free of bias from automatic results. The overall manual labeling system is as shown in Figure 5.3. The inputs to manual labeling are pronunciation dictionaries (can be either canonical or phoneme label pronunciation dictionary as indicated in section 5.4.1.2.1 and section 5.4.1.2.2 respectively), speech corpus and text corpuses of test data (as discussed in section 5.4.1.1). The output of this system has been used as reference for automatic speech segmenter performance evaluation.

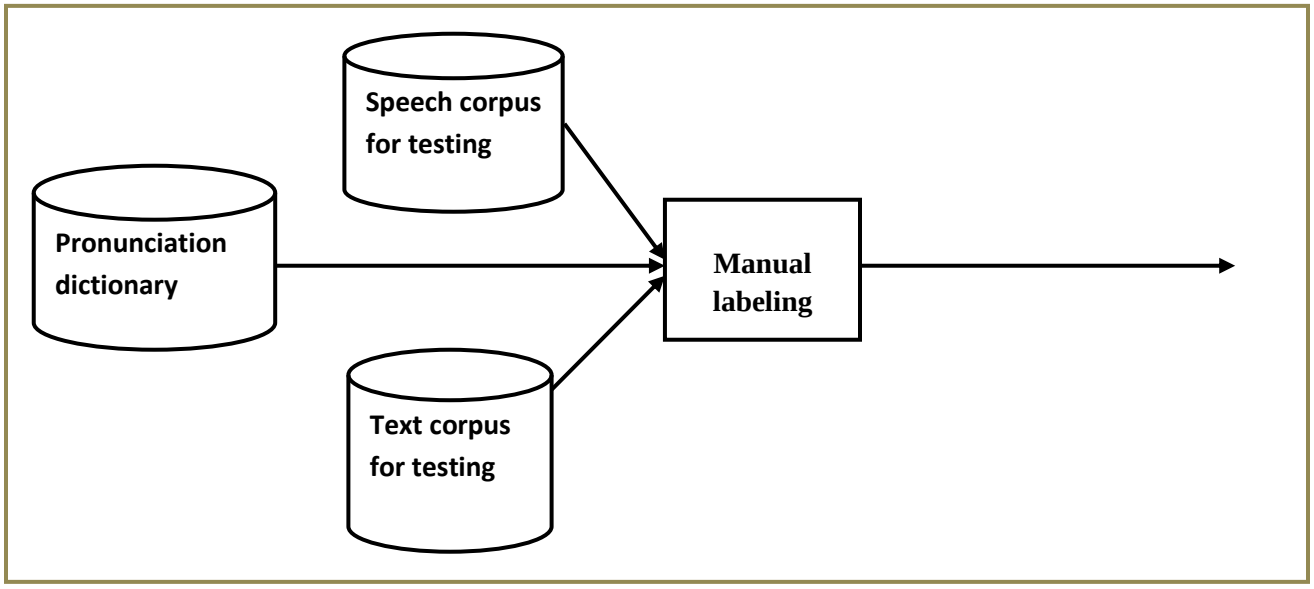

Figure 5.3 Manual labeling components

Even though Manual approaches to speech segmentation are not feasible for large corpora and in a multilingual domain, it is considered by some to be more accurate [71]. To be able to evaluate the results of automatic segmentation found in the above approaches (Grapheme based and phoneme based approaches), a small portion of the speech data (100 sentences used for testing in each speaker) is segmented by hand. This segmentation is carried out via praat. Praat shows the main characteristics of acoustic data both in its spectrogram and wave formats. The segmentation is also performed without any a priori information about test results found during automatic speech segmentation by a single labeler but takes the split test text corpus during manual labeling. In manual segmentation, acoustic signal properties like pitch, formant, duration, intensity, energy, power spectral density and zero crossing rates are very important in order to carry out the segmentation. The wave form of a speech and the spectrogram which includes pitch and formants are selected for our manual Amharic speech segmentation as shown in Figure 5.4. Pitch or sometimes we call it fundamental frequency represents the vibration made on vocal cords during speech production. It helps to identify which phonemes have high pitch value and which are with low value. Formats also tell us the frequency variation at vocal tract. Formant pattern show a typical shape in the start and the end which provides us with information about

specific phoneme. In each acoustic signal of phonemes, there are 4 formants mainly which represent the height of the tongue(F1), length of the tongue (F2) , roundness/un roundness of the tongue(F3) and combination of them (F4) during speech production. The other property used for manual segmentation is the wave form representation of speeches and it varies as per the phoneme characteristics exist in a speech. There is a variation of wave and spectrogram representations as per individual phonemes as shown in Figure 5.4.

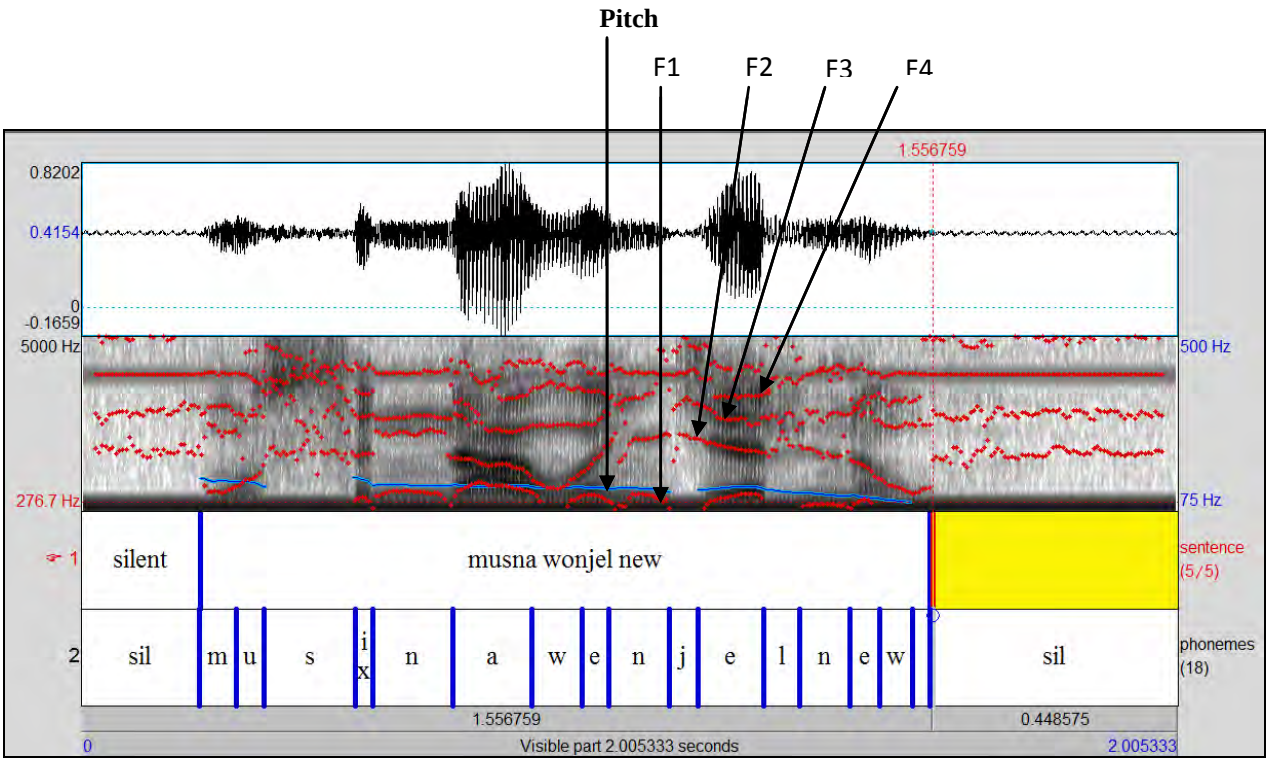

Figure 5.4 Manual labeling on Amharic string "ሙስና ወንጀል ነው" / corruption is crime /

In the acoustic signal, vowels usually show stronger amplitude than consonants, and when viewed in detail, they show a characteristic fundamental frequency pattern which are represented in dot lines in Figure 5.4. Vowels are voiced, and have usually darker region among phonemes. In vowels, the time domain signal is not entirely periodic but it is quasi-periodic due to the repeated excitations of the vocal tract by vocal fold closures.

Based on manner of articulation Amharic consonants are classified as stops, fricatives, affricatives, nasals, liquids and semivowels as indicated in section 4.3.2. Stops are complex sounds that are made up of two distinct initial phases, followed by a transition phase[72].

Initially, while the port is closed, the signal shows silence. At the opening of the port, there is usually a fairly sharp release of the air pressure that has built up in the oral tract during the closure phase. This gives rise to a so-called "burst" and to a short period of frication. In the last stage, the frication merges into the characteristics of the succeeding sound (often a vowel). Acoustically, the closure phase is associated with a minimum of radiated energy. Because the vocal tract is obstructed, little or no acoustic energy is produced. Upon the release, a burst of energy is created as the impounded air escapes. Stop releases are further classified into aspirated and unaspirated. Aspiration is a breathy noise generated as air passes through the partially closed vocal folds into the pharynx. In Amharic stops, the phoneme / h/ is aspirated.

Acoustically, nasal sounds can be identified by prominent vocal cord vibrations, a lessened amplitude with respect to that of adjoining vowels, and a reduced presence of higher formants.

In addition to our segmentation task, we have observed that manual segmentation is not an easy task as we thought, some of the challenges that we have faced during segmentation are:

- Some speech sounds (such as vowels) have much greater acoustic impact ("salience") than do other sounds (such as aspirated or nasal consonants).
- For some other sounds, such as stops, fairly complicated transition patterns exist.
- Some sound like stops are affected by co-articulation. This effect is due to the context of the phoneme that exists before and after it.

5.3.3 Language Modeling

Language model is responsible for detecting connections between letters in a word and words in a sentence during grapheme based approaches and for detecting connections between phonemes in a word and words in a sentence during phoneme based approach. Of course building a bigram, trigram and the like improves the performance automatic speech segmentation but in our text corpus the possibility of getting such kinds of words is low. So unigram language modeling is enough to build the new automatic speech segmentation system. The language model is built by assuming that the next letter or phonemes in the sequence depends only upon previous letter or phonemes. The general block diagram of language model is as shown in Figure 5.5 and it has a great contribution to keep the sequence of letters or phonemes in the given transcribed Amharic

sentences of text corpus. It takes text corpuses and pronunciation dictionary which includes one of dictionaries as the main inputs. It implies that the pronunciation dictionary can be either canonical or phoneme label pronunciation dictionary as indicated in section 5.4.1.2.1 and section 5.4.1.2.2 respectively. The output of this language model is one of the inputs to the next process which are acoustic modeling and HMM segmenter parts of automatic speech segmentation.

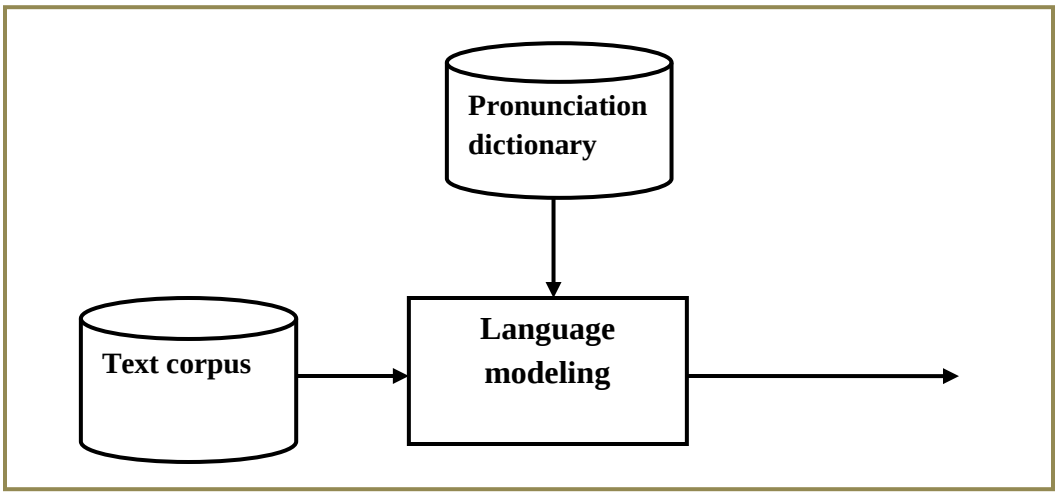

Figure 5.5 Language modeling components

5.3.4 HMM Acoustic Modeling

Acoustic modeling is used to represent distinct sound that makes up a spoken word used in the language model. Each distinct sound corresponds to a phoneme. Acoustic modeling process takes four main inputs; pronunciation dictionary (can be canonical or phoneme label as indicated in section 5.4.1.2.1 and section 5.4.1.2.2 respectively), training text corpus, feature vectors of the training speech corpus and language models (can be either letter sequences or phoneme sequence as indicated in section 5.4.3). It trains the automatic speech segmentation system by making models to individual Amharic phonemes. These models represent individual phonemes with 5 states, three emitting states and two non-emitting states (the first and the last states are non-emitting) without skipping in left to right HMM.

Phoneme based acoustic modeling is the most commonly used HMM model to represent each phonemes with individual HMM. According to Markova rule in order to represent a phoneme with these states some features are taken from the state before and after it.

The acoustic models are statistical models which capture the correspondence between a short sequence of acoustic vectors and letters or phonemes. It is created using audio recordings of speech and their text scripts and compiling them into a statistical representation of sounds which makeup words. This is done through HMM modeling. The overall component of acoustic modeling is shown in Figure 5.6.

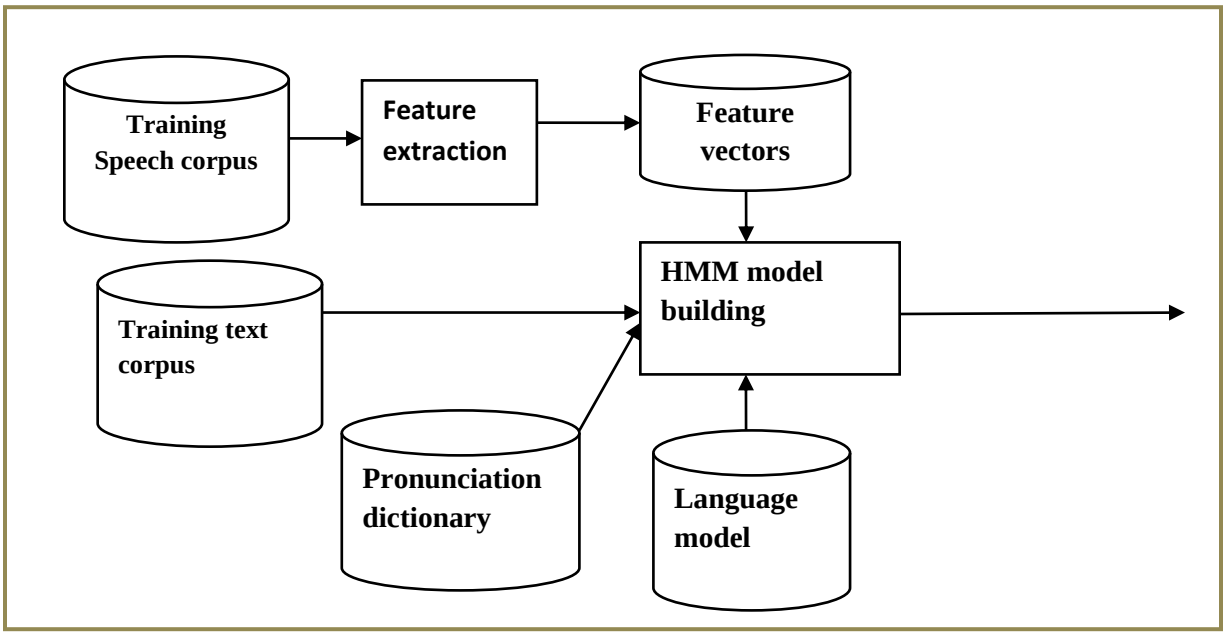

Figure 5.6 Acoustic modeling components

The acoustic modeling system is also carried out in three modeling techniques, namely acoustic modeling without tied state, with tied state and tied state with various Gaussian mixture values as shown in Figure 5.7. The acoustic data used for training are the main inputs for each modeling stages and letter or phoneme sequences of a string are another inputs for HMM modeling. These modeling are considered to get the most accurate results so as to increase the performance of automatic segmenter.

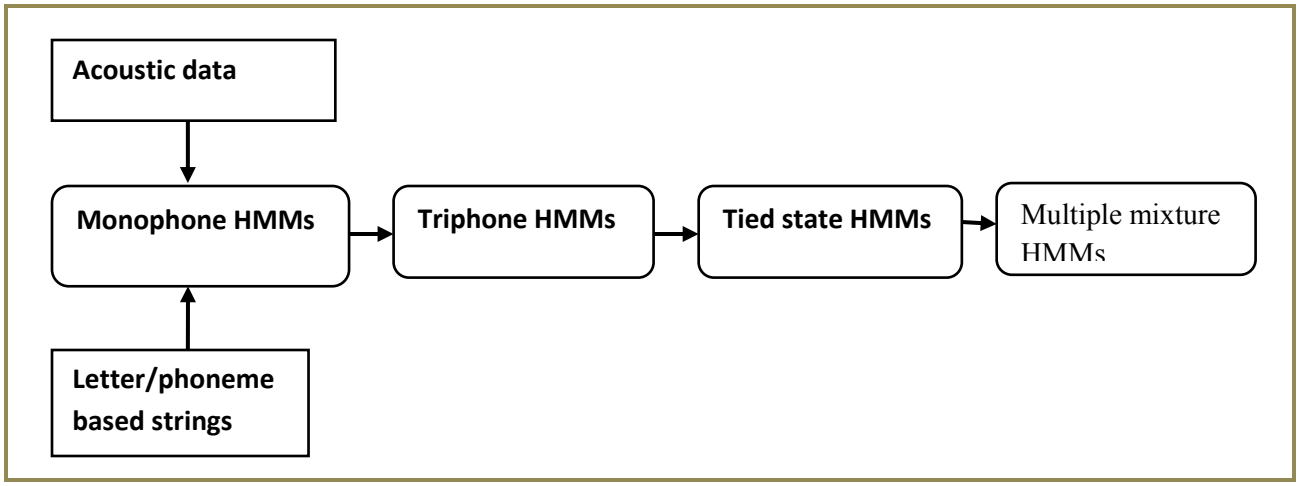

Figure 5.7 The block diagram of acoustic modeling techniques

As shown in Figure 5.7, primarily letters/phonemes are modeled with monophone HMMs without considering its context. This is also termed as acoustic modeling without tied state. It is a common practice to use context-independent HMMs for speech segmentation[20, 73]. Context-dependent HMMs can better model the spectral movements in phonetic transitions[74]. It implies that acoustic modeling which takes its context into consideration is also required as second technique of acoustic modeling.

Having a set of monophone HMMs, triphone HMM model building is required to create context-dependent triphone HMMS. This is done in two steps. Firstly, conversions of monophone transcriptions into triphone transcriptions takes place and create a set of triphone models by copying the monophones and re-estimating them. Secondly, similar acoustic states of triphones are tied to ensure that all state distributions are robustly estimated.

Context-dependent triphones are made by simply cloning monophones and then re-estimating using triphone transcriptions. Then states are tied within triphone sets in order to share data and thus be able to make robust parameter estimates. Although some of them showed acceptable performances, most of the HMM (Hidden Markov Model)-based method uses context sensitive tri-phone model to resolve co-articulation effect[75].

As third technique, Acoustic modeling with Multiple Gaussian mixtures values are used to improve automatic speech segmentation results considerably, because they help avoid the

problem resulting from the usage of the same type of probability density distribution for different models and states. So that context dependent HMMs with different probability distributions are used for further improvement of automatic speech segmentation. If an HMM state is made to contain multiple Gaussian mixture components, then the training vectors would be associated with highest likelihood mixture component. The number of vectors associated with each component within a state can then be used to estimate the mixture weights.

5.3.5 HMM Segmenter

HMM segmenter takes four main inputs which are mainly used to complete the automatic segmentation as per already trained during acoustic modeling as shown in Figure 5.8. These inputs are pronunciation dictionary, language model, feature vectors and acoustic models. Pronunciation dictionary can be either canonical or phoneme label as indicated in section 5.4.1.2.1 and section 5.4.1.2.2 respectively. As indicated in section 5.4.3, language models can be either letter sequences or phoneme sequences based on the type of pronunciation dictionary used. Feature vectors are found from test speech corpus via feature extraction process as indicated in section 5.4.1.4. Acoustic models can be monophone HMMs, tied state HMMs or multiple mixture HMMs as indicated in section 5.4.4.

The HMM segmenter gives automatically segmented results of the test data set. Forced alignment using the HMM-based segmenter relies on knowledge of the underlying phonetic transcription of a given utterance. In general, once the orthographic transcription and speech data are available, it is possible to employ forced alignment for automatic phonetic transcription.

After getting the three main inputs, the HMM segmenter assigns the corresponding letter or phoneme to acoustic signal as per the training and best selected pronunciation of a word. The Viterbi algorithm selects the best way and the correct pronunciation of a word. Finally, the recognized sequence of letters or phonemes aligned with their time boundaries found on acoustic signal.

HMM-based aligner requires the phoneme sequence as an input since their alignment is accomplished based on a sequence of orthographic words, a pronunciation dictionary (including variations of pronunciation per word) and stochastic models of phonemes. At the end, phonemes with their time boundaries are obtained as an output of automatic HMM based speech segmenter.

This output can be either letter sequence or phoneme sequences with their time boundaries in context independent, context dependent with single Gaussian mixture or context dependent with multiple Gaussian mixture segmentation contexts. These outputs are the last stage of automatic speech segmentation system.

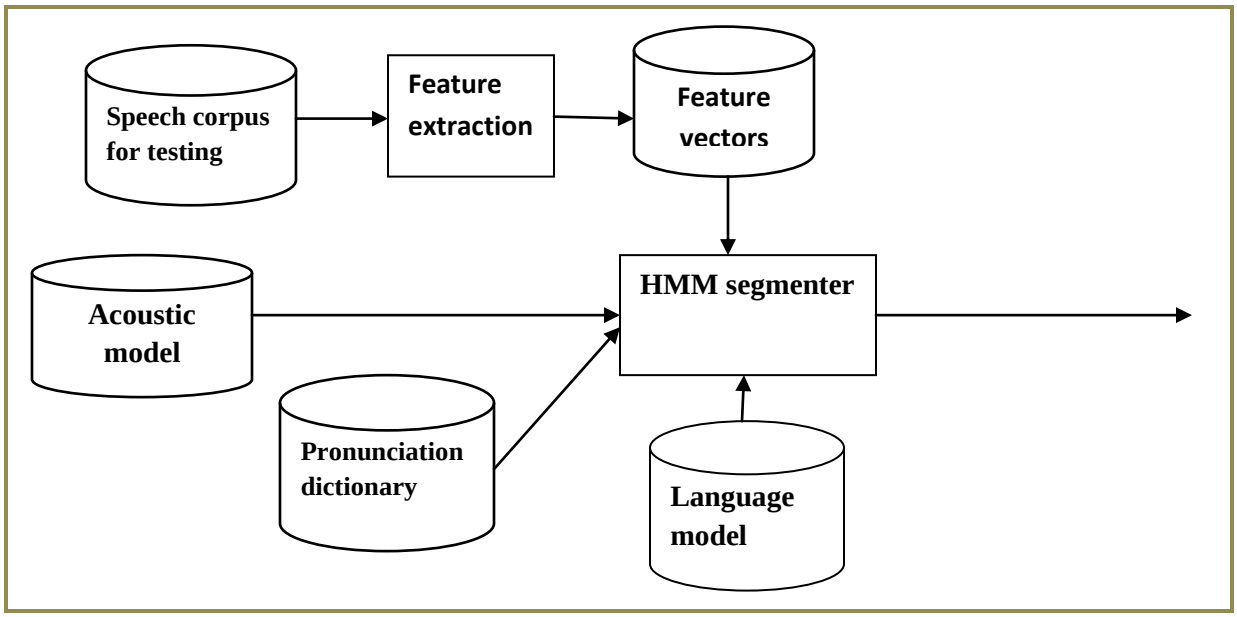

Figure 5.8 Phoneme based HMM segmenter components

5.3.6 Verifications

The HMM segmenter results can be phonemes found in context independent HMMs, context dependent with tied state HMMs or context dependent with multiple mixture HMMs. The time boundaries of each automatically segmented result are evaluated with manual segmented phonemes as references. This is termed as verification and it is used to evaluate the performance of automatic speech system with reference to manually identified phoneme boundaries. This evaluation is given based on the tolerance values like 5ms, 10ms, 15ms and 20ms. The verification components takes tolerance values, HMM segmenter results which consists of automatically segmented phonemes with time boundaries and manually segmented phonemes time boundaries as main inputs as shown in Figure 5.9.

In order to measure the performance of automatic speech segmentation, mapping concept is used[76]. During mapping; every phonemes found during automatic speech segmentation is

compared to their corresponding phonemes found in manual speech segmentation and epithetic vowel is considered as part of letters during letter based approach. In overall, no letter is passed without mapping during verification components. As explained in 5.3, 10% of the collected data is used for testing purpose by applying manual speech segmentation system on them.

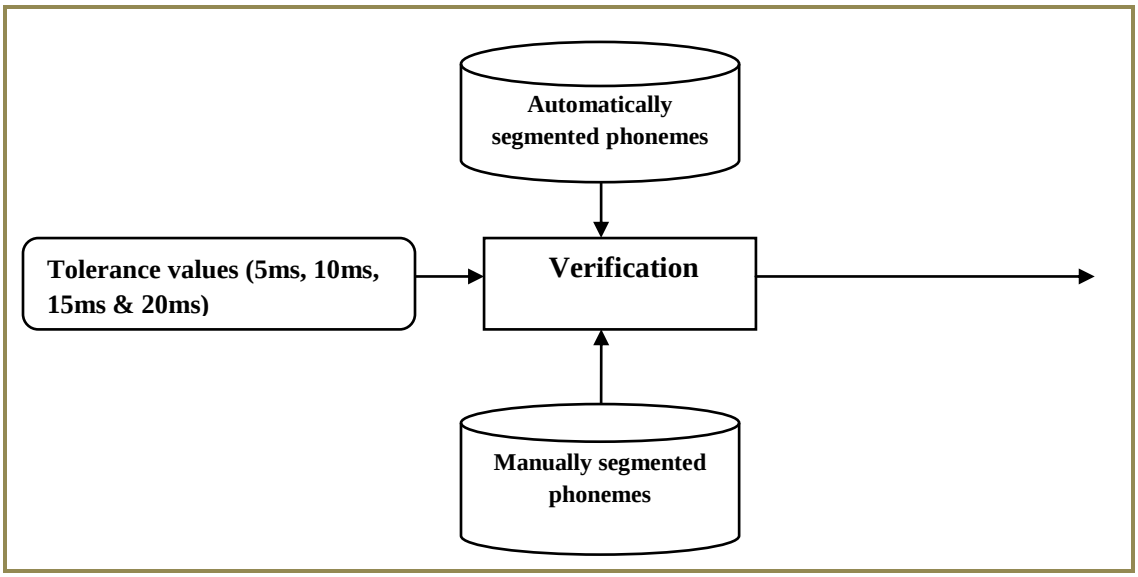

Figure 5.9 Verification components for performance evaluation

These manually segmented phoneme sequences are compared with each letter and phoneme sequences found during automatic speech segmentation in both Grapheme and phoneme based approaches. In one to one mapping of letters and phoneme; 4 main cases are taken into consideration.

Case 1: When both the initial and final time boundaries of manually segmented phonemes are greater than the initial and final time boundaries of automatically segmented Phonemes.

Case 2: When the initial time boundaries of manually segmented phonemes are less than the initial time boundaries of automatically segmented phonemes where as the final time boundaries of manually segmented phonemes are greater than the initial time boundaries of automatically segmented phonemes.

Case 3: When the initial time boundaries of manually segmented phonemes are greater than the initial time boundaries of automatically segmented phoneme where as the final time boundaries

of manually segmented phonemes are less than the initial time boundaries of automatically segmented phonemes.

Case 4: When both the initial and final time boundaries of manually segmented phonemes are less than the initial and final time boundaries of automatically segmented Phonemes.

The general formula used to calculate the deviation of initial time boundaries of a phoneme or a letter and the deviation of final time boundaries of a phoneme or a letter are indicated in (1) and (2) respectively.

$$t_1=|t_i-t''_i| \quad (1)$$

$$t_2=|t_{i+1}-t''_{i+1}| \quad (2)$$

where: t_i is initial time boundary of manually segmented letters/phonemes.

t''_i is initial time boundary of automatically segmented letters/phonemes.

t_1 is time difference between initial time boundaries of manually and automatically segmented phonemes.

t_{i+1} is final time boundary of manually segmented letters/phonemes.

t''_{i+1} is final time boundary of automatically segmented letters/phonemes.

t_2 is time difference between final time boundaries of manually and automatically segmented phoneme

The time differences found in (1) and (2) are deviation between manually segmented and automatically segmented phonemes or letters. In our study, boundary deviations are evaluated in four different tolerance values such as 5ms, 10ms, 15ms and 20ms. It implies that the deviations exceeding these tolerances are considered as errors. Finally, the error is expressed in terms of percentage of deviation and it is calculated in quantitative method using (3).

$$\%\text{deviation} = \frac{\text{No of phonemes exceeding the tolerance value}}{\text{total number of boundaries tested}} \times 100 \quad (3)$$

CHAPTER SIX

EXPERIMENTAL RESULTS AND EVALUATION

6.1 Introduction

This experimentation covers two techniques of speech segmentation, namely automatic speech segmentation using HTK toolkit and manual segmentation/labeling using praat software. Firstly, automatic segmentation is implemented in two phases with HTK toolkit. These phases differ in basic units of lexicon preparation which is used as pronunciation dictionary. The basic units are sequence of letters and phonemes in phase1 and phase2 respectively. In each phase, HMM modeling techniques and HTK commands are used to complete the task of automatic speech segmentation. Data preparation, HMM modeling and segmentation are the main stages of it.

Data preparation includes Amharic text corpus preparation, lexicon preparation, speech corpus preparation to the corresponding text corpus, data transcription to HTK usable format and parameterization of speech signals. In corpus based speech segmentation preparing training and testing data sets is required and for this purpose systematic random sampling technique is used to split both text and speech corpuses into training data sets (90%) and testing data sets (10%). Training data sets are used for language and acoustic modeling purpose where as testing data sets are used for evaluation of automatic HMM segmenter. Since HTK doesn‟t use speech data directly transcribing them into phone level and word level, and parameterization of them is also required as part of data preparation. Parameterization of speech data takes place through feature extraction process.

After data preparation, acoustic/ HMM modeling and HMM segmentation are taking place in context independent and in context dependent with single Gaussian mixture and context dependent with multiple Gaussian mixtures environments.

Secondly, manual segmentation also takes place on test data sets in addition to automatic speech segmentation applied on them. Manually segmented phonemes are its results and they are achieved by hand labeling of phonemes with their time boundaries. These manual segmented

phonemes are used to evaluate the performance of automatic speech segmentation system. This evaluation is carried out in terms of percentage of deviation. It implies that the deviation of time boundaries of automatic segmented phonemes with reference to hand segmented phonemes since manual segmentation are considered as accurate results.

The deviation of time boundaries are calculated in tolerance values of 5ms, 10ms, 15ms and 20ms. Finally, the performance of automatic segmenter in both phases with reference to manual segmentation is evaluated in terms of percentage of boundary deviation. The test results are also presented in both tabular and textual forms.

6.2 Data preparation

In order to build phonetically balanced speech corpus, text data should be collected from various sources of Amharic documents as indicated in section 5.4.1.1. The samples of collected Amharic texts is attached in Appendix G. Transliteration of the Amharic texts into their corresponding ASCII representation is time taking and prone to error if it is done manually. In order to precede the automatic speech segmentation with error free transcribed texts, Payton software with version 3.1 is used and a Payton code which takes ASCII translation table as input is developed for this purpose as attached in Appendix C. This software transcribes automatically Amharic texts to their corresponding Latin representation as per ASCII transliteration table attached in Appendix B.

The transcribed texts split into two. These are training and testing set with 900 sentences and 100 sentences respectively. These split set of texts are known as prompt files and considered as text corpuses which need to be recorded. Prompt files are used in both phases of experimentation by categorizing words into grammar file and making sentences in standard lattice format. HTK provides a grammar definition language for specifying the task of grammars. It consists of a set of variable definitions followed by a regular expression describing the sentences that exist on the prompt file. The grammar used for new automatic speech segmentation system is attached in Appendix H. The HTK segmenter requires a sentence network to be defined using a low level notation called HTK Standard Lattice Format (SLF) in which each sentence is listed explicitly as attached in Appendix I. This network can be created automatically from the grammar using the

HParse tool of HTK toolkit. Since the grammar file contains set of rules governing words in a sentence, its execution creates an equivalent sentence network. Other main parts of data preparation, namely lexicon preparation, data recording, creating transcription files and coding the acoustic data are discussed in the next subsections.

6.2.1 Lexicon preparation

Lexicons are prepared as pronunciation dictionaries for our automatic speech segmentation system. The HTK Perl script **prompts2wlist** can take the prompts file and produce words in sorted order (wlist). After we have sorted word list, lexicon preparation is required for both phases. The first lexicon is prepared directly from wlist file but in the second lexicon epithetic vowels are incorporated by plug-in the epithetic vowel insertion algorithm to each words of the wlist file.

The overall lexicon preparation processes of both phase1 and phase2 are described diagrammatically as shown in Figure 6.1. In phase1, transcription of every word into their corresponding Latin alphabets takes place and these sequences of Latin alphabets are considered as pronunciation dictionary. Grapheme to phoneme conversion through epithetic insertion algorithm is considered in phase 2. These phonemes sequences are also considered as second pronunciation dictionary.

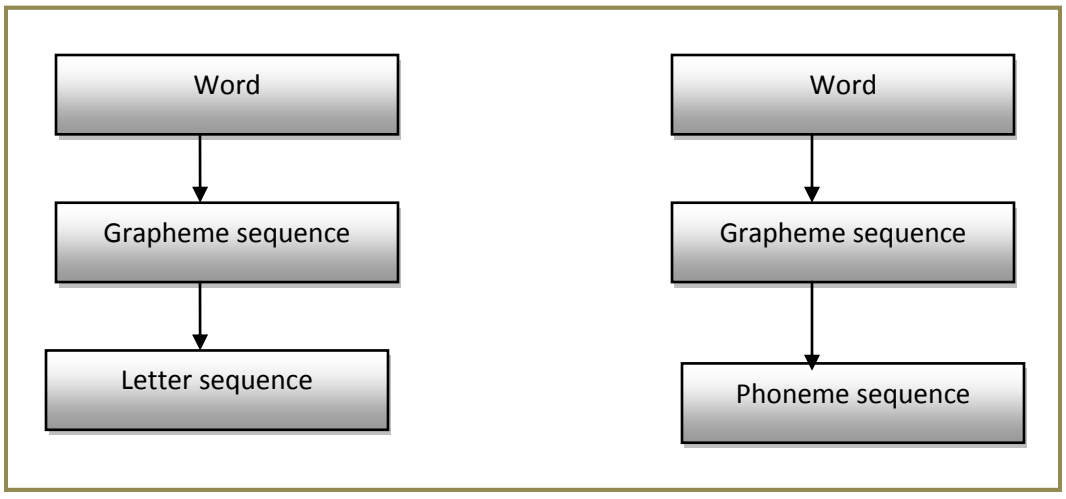

Figure 6.1 Lexicon preparation steps in both phase1 (left) and phase2 (right)

For example; the Amharic word “በላ” has two orthographies which are በ and ላ. Its corresponding ASCII transcription becomes ”bela”. According to ASCII translation, “በ“ is represented with “be” and „ላ“ represented with “la”, and their letter sequences together becomes “b e l a”. Unlike letter sequences, phoneme sequences include an epithetic vowel /ix/ [ɨ] and these phoneme sequences are nearly phonetic representation of Amharic words. Let us consider Amharic word “ትምህርት”, its letter sequence is “t m h r t” but its phoneme sequence becomes “t ix m h r ix t” which includes the epithetic vowel /ix/ [ɨ].

Based on the above idea, lexicons are prepared with letter sequences and phoneme sequences for phase1 and phase2 respectively. The general formats used in our lexicons preparation are like this;

WORD [output] p1 p2 p3.... p_n

Where WORD is the transcribed word found in wlist and p1 p2 p3....p_n are either letter sequences or phoneme sequences found in each word. Two lexicons are prepared for both phases and named them lexicon1 and lexicon2. Some of the examples used in both lexicon1 and lexicon2 are given below.

Lexicon 1			Lexicon 2		
kll	[kll]	k l l	kixll	[kixll]	k ix l l
klln	[klln]	k l l n	kixllixn	[kixllixn]	k ix l l ix n
kffl	[kffl]	k f f l	kixffixl	[kixffixl]	k ix f f ix l

HDMan command of HTK toolkit is used to prepare pronunciation dictionary by taking lexicons as input. The format of these pronunciation dictionaries of both lexicon 1 and lexicon 2 are the same except including short pause at the end of each word as attached in Appendix J.

6.2.2 Data recording and speech corpus preparation

Randomly two Amharic speakers who are not parents are selected from both genders and their profiles are as indicated in Table 6.1. One thousand sentences listed in the prompt file

(transcribed text corpuses) are recorded for both training and testing data in both speakers. Audacity software is the tool or software used during data recording.

Table 6.1 Speakers profile for speech recording

Speaker code	sex	Age	No of sentences recorded for training	No of sentences recorded for testing
Spk01	M	29	900	100
Spk02	F	25	900	100

After selecting the above speakers and microphone volume set, the audacity software also requires the following settings as preferences:

- set the microphone volume to **1.0**.
- set the default Sample Rate Format to **48KHz.**
- set the default Sample Format to **16-bit**.
- set the Channels to **1 (Mono)**.
- set the Uncompressed Export Format to **WAV (Microsoft 16 bit PCM)** or export the audio using **FLAC** format.

After getting the proper setting of audacity, data are recorded by each speaker properly with silent environment. The wave files of 900 sentences are used for training and the rest 100 sentence wave files are testing purpose.

6.2.3 Creating transcription files

To train a set of HMMs every file of training data should have an associated phone level transcription. To make this task easier, creating a word level transcription before creating the phone level transcription is required. The word level transcription is created by executing the Perl script prompts2mlf provided with the HTK toolkit. It uses the previously prepared prompt file as input. Using the created word level *MLF*, phone level transcription is created using the HTK label editor, *HLEd*. HLEd works by reading in a list of editing commands from an edit script file and then makes an edited copy of one or more label files. HLEd edit script is used to insert the silence model „sil" at the beginning and end of each utterance.

6.2.4 Coding the acoustic data

Recorded speech signals are parameterize into sequence of feature vectors and used in automatic speech segmentation process. This parameterization of speech signals known as feature extraction, feature vectors are its output as indicated in section 5.3. Feature extraction is performed using the *HCopy* tool by taking configuration script file attached in Appendix K in addition to wave files. This tool makes a copy of data and converts to MFCC format which are ready to use for HMM training. Here, Mel Frequency Cepstral Coefficients (MFCC_O_D_A - 12 melcepstral coefficients, 12 delta coefficients, 12 acceleration coefficients, log energy, delta energy, and acceleration energy) is used to parameterize the speech signals into feature vectors with 39 MFCC coefficients.

6.3 HMM model building and segmentation

The training phase of our experiment is done to build acoustic model for new segmentation system and it is helpful to segment Amharic phonemes through HMM segmenter. The segmentation process takes place with different techniques so as to obtain the required segmented results. These techniques are accomplishing phoneme automatic segmentation without tied state, with tied state and tied state with multiple Gaussian mixtures. Their detail presentation is given in the following HMM modeling and segmentation subsections.

6.3.1 Without tied state

The major process in unsupervised method of automatic speech segmentation is building and training the acoustic model because there is a predefined phoneme boundary. The first step of this process is to create a prototype HMM as attached in Appendix L. This prototype defines the structure and the overall form of the set of HMMs. Here a 3-state left-right topology is used to model the HMMs. The second step is initializing the monophone HMMs. For this purpose HTK uses the *HCompV* tool. Inputs for this tool are the prototype HMM definition and the training data. HCompV reads both inputs and gives an output with new definition in which every mean and covariance is equal to the global speech mean and covariance. So, every state of a monophone HMM gets the same global mean and covariance.

Next, a Master Macro File (MMF) called hmmdefs containing a copy for each of the required monophone HMMs is constructed. The next step is to re-estimate the stored monophones using the embedded re-estimation tool *HERest*. This process estimates the parameters of monophone HMMs from the training set that are intend to model. This is the process of training HMMs. The re-estimation procedure is repeated three times for each of the HMM and optimum training is obtained.

After we have a sufficient model to obtain time-aligned word and phone transcriptions. We'll use another instance of HVite to get output with the most likely alignments. The model works by adjusting alignments to maximize the degree to which phones cluster. So HTK will have computed the most likely location of every phoneme within the linear order of a sentence using the model we've built so far.

The HVite commands used above are helped us to transform the input word level transcription to new phone level transcription monophone using the pronunciations stored in the dictionary. The key difference between these operations and the original word-to-phone mapping performed by HLEd in previously executed pronunciation dictionary is that the segmenter considers all pronunciations for each word and outputs the pronunciation that best matches the acoustic data. The segmentation output of the HVite command executed by HTK toolkit is attached in Appendix M and the Units for time stamps is 100 nano seconds (1 unit = 100 ns (nanosecs)).

6.3.2 With tied state

Context dependent automatic speech segmentation is the continuation of context dependent automatic speech segmentation. It is understood that all previously steps of automatic speech segmentation without tied state are repeated in order to continue with tied state automatic speech segmentation.

After re-estimating the context independent monophone HMMs, we move onto context dependent triphone HMMs. These triphones are made simply by cloning the monophones and then re-estimating using triphone transcriptions.

The next step is to re-estimate the new triphone HMM set using the HERest tool. This is done in the same way as the monophone HMMs were estimated by replacing the monophone list and the monophone transcription with the corresponding triphone list and the triphone transcription.

This process is also repeated three times. The last step in model building process is to tie states within triphone sets in order to share data and thus be able to make robust parameter estimates. However, the choice of which states to tie requires a bit more subtlety since the performance of the segmenter depends crucially on how accurate the state output distributions capture the statistics of the speech data. In this thesis work it uses decision trees attached in Appendix N to tie the states within the triphone sets.

The final step of the acoustic modeling is the re-estimation of created tied state triphones and this process is also same as the earlier use of HERest command of HTK toolkit. This is also repeated for three times and the final output is trained acoustic model. The re-estimated HMMs are used with HVite to perform forced alignment on a speech signal given the phoneme sequence. The segmentation output of the HVite command executed by HTK toolkit is attached in Appendix O and the Units for time stamps is also 100 nano seconds.

6.3.3 With tied state and multiple Gaussian mixtures

In this experiment, Automatic speech segmentation is carried out in context dependent segmentation of phonemes with different probability distribution. These various probability distributions are obtained with various Gaussian mixture values.

In HTK, the conversion from single Gaussian HMMs to multiple mixture component HMMs is implemented using the HHEd MU command which will increase the number of components in a mixture by a process called mixture splitting. In this method the command works repeatedly by splitting the mixture with the largest mixture weight until the required number of components is obtained. This also allows recognition performance to be monitored to find the optimum. By way of example, one of the scripts used, containing all the MU commands is attached in Appendix P.

After phonemes are model in different Gaussian mixture values, three further re-estimation takes place through acoustic modeling process. This re-estimation is the continuation of previous tied

state or context dependent automatic speech segmentation. The re-estimated HMMs are used with HVite to perform forced alignment on a speech signal given the phoneme sequences. This HVite command also used to transform the input word level transcription to new phone level transcription of multiple Gaussian mixtures using the pronunciations stored in the dictionary. The segmentation output of the HVite command at Gaussian mixture 16 executed by HTK toolkit is attached in Appendix Q and the Units for time stamps is still in nano seconds.

6.4 Test data preparation and Result analysis

6.4.1 Test data preparation using manual segmentation

In each female and male recorded speech corpuses, 100 wave files are selected with systematic random sampling technique. These sentences are not included in the training phase of automatic speech segmentation.

The speech segmentation task is done manually using praat software/tool. By looking at the acoustic signal wave and spectrogram representations of a speech and listening to small parts of speech in order to decide where exactly to place the boundary of each phoneme. Manually segmented phoneme results are obtained and these results are still considered as most accurate speech segmentation method if the segmenter spends enough time and concentration on the corpus to segment, and it is also supposed to be perceptively validated. But processing time is a major drawback in addition to the experience required for more accurate segmentation.

In a similar technique of speech segmentation, we have identified the phoneme boundaries from the acoustic signal by observing the wave forms and spectrogram of speech utterances. The signal properties used to identify the boundaries are mainly formants and pitch representation of it. In all 100 sentences, various boundaries of a phoneme are found depending on the context used. These variations are not considered as a deviation or an error because the boundary deviations are considered with reference to automatic phonemes with the concept of phoneme mapping.

At the end, phoneme boundaries are identified through manual segmentation technique. These boundaries are expressed in milliseconds (ms) and the sample of manually labelled phonemes with their time boundaries are attached in Appendix R.

6.4.2 Test results and analysis

6.4.2.1 Experimental test results

In order to measure the performance of automatic speech segmentation, phoneme mapping concept is used. These manually segmented phoneme boundaries are compared with each phoneme sequences found during automatic speech segmentation where as epithetic vowel is considered as part of letters in case of grapheme based approach. The technique used to measure the performance of automatic segmentation is percentage of phoneme boundaries deviation with in tolerances values of 5ms, 10ms, 15ms and 20 ms[27]. Since the formula used to measure the performance of automatic speech segmentation have been discussed in the previous chapter, only the results achieved using it is presented here.

The phoneme boundary values beyond the tolerance values are considered as errors and these errors are expressed in terms of percentage through statistical technique. In this technique the number of phonemes occurred beyond the tolerance value and its percentage is given with reference to the total number of phonemes exist in each phases. The total numbers of phonemes exist in phase1 and phase2 are 5080 and 5404 respectively. Since epithetic vowels are inserted by plug in epithetic vowel insertion algorithm, the numbers of phonemes in phase2 are greater than phase1. These total phoneme sizes do not include short pauses exit in each speech utterances and this is expected to reduce the percentage of boundary deviation in each tolerance values. Having the performance evaluation technique, the test results of three automatic speech segmentation experiments, namely without tied state, with tied state and tied state with multiple Gaussian mixtures are presented individually.

Experiment I results: Segmentation performance without tied state

In this experiment, phonemes without context dependent are considered and their performances are also presented. The overall all results of the automatic segmentation in both phase1 and phase2 with single probability distribution are shown in Table 6.2 and Table 6.3.

Table 6.2 Letter based system (phase1) without tied state experimental results

Speaker code	**Test type**	**Time boundary**	**% of boundary deviation in tolerances**			
			5ms	10ms	15ms	20ms
Spk01	Without tied state	Initial (t1)	8.29	3.13	1.75	1.44
		End (t2)	5.61	2.22	1.63	1.40
		Both(t1+t2)	15.37	5.45	2.66	1.95
Spk02	Without tied state	Initial(t1)	6.72	2.60	2.83	2.12
		End (t2)	6.12	2.72	2.34	2.18
		Both(t1+t2)	12.16	4.37	3.03	2.46
Average			**9.05**	**3.42**	**2.37**	**1.93**

Table 6.3 Phoneme based system (phase2) without tied state experimental results

Speaker code	**Test type**	**Time boundary**	**% of boundary deviation in tolerances**			
			5ms	10ms	15ms	20ms
Spk01	Without tied state	Initial(t_1)	7.79	2.94	1.65	1.35
		End (t_2)	5.27	2.09	1.54	1.31
		Both(t_1+t_2)	14.45	5.13	2.50	1.83
Spk02	Without tied state	Initial(t_1)	4.98	1.81	1.48	1.42
		End (t_2)	4.56	1.87	1.52	1.44
		Both(t_1+t_2)	9.10	3.15	1.92	1.68
Average			**7.69**	**2.83**	**1.77**	**1.51**

As shown in both Table 6.2 and Table 6.3, the percentage deviation is minimized in phase2 for both speakers since the pronunciation dictionary developed in phase2 is more phonetic than phase1. The percentage of deviation presented with initial time boundary, end time boundary and both of them of phonemes. It is found that the percentage of boundary deviation is more in initial time boundaries of phonemes as compared to final time boundaries of them. On the other hand, the percentage of boundary deviation decreases rapidly when the boundary of deviation from 5ms to 20ms. The percentage of deviation beyond 20ms tolerance values are due to phoneme recognition errors and the result shows that all phoneme are almost within 20ms tolerance values. Generally, best result is achieved at phase2 and the average results obtained for both speakers in terms of percentage of deviation with the same probability distribution of context independent are 7.69%, 2.83%, 1.77% and 1.51% within 5ms, 10ms, 15 ms and 20ms tolerance values respectively.

Since this result is achieved automatic speech segmentation without considering contexts of phonemes, considering phonemes context is required. Phonemes context mainly depend on the co-articulation effect made by the phonemes before and after it. For this reason further experiment with tied state is required.

Experiment II results: Segmentation performance with tied state

In this experiment, phonemes with context dependent are considered and their performances are also presented. The overall all results of the automatic segmentation in both phase1 and phase2 with tied state are shown in Table 6.4 and Table 6.5.

Table 6.4 Letter based system (phase1) with tied state experimental results

Speaker code	**Test type**	**Time boundary**	**% of boundary deviation in tolerances**			
			5ms	10ms	15ms	20ms
Spk01	With tied state	Initial (t_1)	7.83	2.70	1.32	1.00
		End (t2)	5.06	1.77	1.12	0.94
		Both(t1+t2)	14.45	4.82	2.28	1.56
Spk02	With tied state	Initial (t1)	6.69	2.58	2.22	2.11
		End (t2)	6.12	2.72	2.34	2.18
		Both(t1+t2)	12.16	4.37	3.03	2.46
Average			**8.72**	**3.16**	**2.05**	**1.71**

Table 6.5 Phoneme based system (phase2) with tied state experimental results

Speaker code	**Test type**	**Time boundary**	**% of boundary deviation in tolerances**			
			5ms	10ms	15ms	20ms
Spk01	With tied state	Initial(t1)	7.36	2.54	1.24	0.94
		End (t2)	4.76	1.67	1.05	0.89
		Both(t1+t2)	13.58	4.53	2.15	1.46
Spk02	With tied state	Initial(t1)	4.90	2.04	1.61	1.50
		End (t2)	4.46	2.02	1.63	1.52
		Both(t1+t2)	9.04	3.55	2.165	1.795
Average			**7.35**	**2.73**	**1.64**	**1.35**

As shown in both Table 6.4 and Table 6.5, the result of context dependent (experiment II) showed that the percentage of boundary deviation become lesser than context independent

(experiment I) results. It implies that the performance of automatic speech segmenter result is improved in tied state HMMs. Like experiment I; the percentage of time boundary deviation differs in each speaker, minimum boundary deviation is achieved at phase2, the percentage of boundary deviation is more in initial time boundaries of phonemes as compared to final time boundaries of them and the percentage of boundary deviation decreases rapidly when the boundary of deviation from 5ms to 20ms. In over all, these results show the context dependent automatic speech segmentation improves the segmentation performance in comparison to the earlier context independent automatic speech segmentation.

At the end best result is achieved at phase2 and the average results obtained for both speakers in terms of percentage of deviation with the same probability distribution of context dependent are 7.35%, 2.73%, 1.64% and 1.35% within 5ms, 10ms, 15 ms and 20ms tolerance values respectively. Even though the result is improved during tied state automatic speech segmentation, this experiment does not consider the probability density function with different values. In order to consider the various probability density functions, further experiment with different Gaussian mixtures is also required.

Experiment III results: Segmentation performance with tied state and multiple Gaussian mixtures

Multiple Gaussian mixture systems are said to improve automatic speech segmentation results considerably because they help avoid the problem resulting from the usage of the same type of distribution for different models and different states. During mixture incrementing some component weights may become very small, resulting in defunct mixture components. Defunct mixtures often indicate that not enough training data is available to further increase the mixtures of a model. So, the best strategy employed in many systems is incrementing the mixture components in stages by a factor of N[50]. Thus taking the single Gaussian mixtures system during tied state, the mixtures are incremented by a factor of two until 16 mixture component HMMs are obtained. We have also checked these mixture increments with preliminary experiment. Then at each stage re-estimating and checking speech segmentation results is performed as shown in Table 6.6 and Table 6.7.

Table 6.6 Letter based (phae1) system with tied state and multiple Gaussian Mixture experimental results.

Speaker code	**Test type**	**No of Gaussian mixtures per state**	**Time boundaries**	**% of boundary deviation in tolerances**			
				5ms	10ms	15ms	20ms
Spk01	Tied state with multiple Gaussian mixtures	2	Initial (t_1)	8.03	3.05	1.71	1.38
			End (t_2)	5.33	2.09	1.52	1.36
			Both(t_1+t_2)	15.04	5.31	2.60	1.83
		4	Initial (t_1)	**7.80**	**2.91**	**1.40**	**1.06**
			End (t_2)	**5.08**	**1.81**	**1.22**	**1.00**
			Both(t_1+t_2)	**14.43**	**4.84**	**2.22**	**1.57**
		8	Initial (t_1)	9.45	4.43	2.95	2.54
			End (t_2)	6.59	3.39	2.72	2.48
			Both(t_1+t_2)	16.47	6.63	3.98	3.21
		16	Initial (t_1)	9.13	4.15	2.58	2.19
			Final(t_2)	6.38	3.07	2.42	2.19
			Both(t_1+t_2)	16.02	6.32	3.56	2.76
Spk02	Tied state with multiple Gaussian mixtures	2	Initial (t_1)	6.61	2.74	2.30	2.22
			End (t_2)	6.06	2.79	2.38	2.30
			Both(t_1+t_2)	12.72	4.59	2.93	2.60
		4	Initial (t_1)	**5.71**	**1.58**	**1.08**	**0.95**
			End (t_2)	**5.18**	**1.61**	**1.12**	**1.02**
			Both(t_1+t_2)	**12.83**	**3.60**	**1.93**	**1.32**
		8	Initial (t_1)	5.79	1.54	1.08	0.95
			End (t_2)	5.18	1.50	1.10	1.00
			Both(t_1+t_2)	12.89	3.66	1.83	1.26
		16	Initial (t_1)	5.73	1.58	1.12	1.00
			Final(t_2)	5.16	1.58	1.10	1.02
			Both(t_1+t_2)	12.85	3.64	1.91	1.34

Table 6.7 Phoneme based system (phase2) with tied state and multiple Gaussian Mixture experimental results.

Speaker code	**Test type**	**No of Gaussian mixtures per state**	**Time boundaries**	**% of boundary deviation in tolerances**			
				5ms	10ms	15ms	20ms
Spk01	Tied state with multiple Gaussian mixtures	2	Initial (t_1)	**6.29**	**2.09**	**0.81**	**0.44**
			End (t_2)	**3.78**	**1.11**	**0.61**	**0.39**
			Both(t_1+t_2)	**12.23**	**3.87**	**1.67**	**0.87**
		4	Initial (t_1)	7.05	2.68	1.33	0.98
			End (t_2)	4.40	1.63	1.13	0.91
			Both(t_1+t_2)	12.97	4.42	2.18	1.42
		8	Initial (t_1)	7.90	3.44	2.07	1.66
			End (t_2)	5.11	2.37	1.83	1.59
			Both(t_1+t_2)	13.95	5.16	2.98	2.20
		16	Initial (t_1)	7.79	3.42	2.00	1.65
			Final(t_2)	5.14	2.39	1.83	1.61
			Both(t_1+t_2)	13.66	5.14	2.89	2.15
Spk02	Tied state with multiple Gaussian mixtures	2	Initial (t_1)	5.09	1.70	1.30	1.280
			End (t_2)	4.85	1.65	1.31	1.30
			Both(t_1+t_2)	9.83	3.41	1.81	1.48
		4	Initial (t_1)	**3.78**	**0.50**	**0.07**	**0.06**
			End (t_2)	**3.29**	**0.46**	**0.07**	**0.06**
			Both(t_1+t_2)	**8.29**	**2.15**	**0.65**	**0.26**
		8	Initial (t_1)	3.81	0.70	2.78	2.04
			End (t_2)	3.39	0.61	0.28	0.204
			Both(t_1+t_2)	8.64	2.18	0.70	0.43
		16	Initial (t_1)	3.79	0.54	0.07	0.06
			Final(t_2)	3.31	0.50	0.07	0.06
			Both(t_1+t_2)	8.49	2.17	0.63	0.30

As shown in Table 6.6, the percentage of boundary deviation becomes less in tied state with multiple Gaussian mixtures as compared to the percentage deviation found in letter based with context dependent experiment (presented in Table 6.4). Even though the experiment is carried out with different Gaussian mixture values, best results with minimum percentage of boundary deviation are found at Gaussian mixture 4 in both male and female speakers. For male speaker at Gaussian mixture 4; the percentage of boundary deviation in initial time boundary are 7.80%, 2.91%, 1.40% and 1.06%, in end time boundary are 5.08%, 1.81%, 1.22% and 1.00% and in both initial and final time boundaries are 14.43%, 4.84%, 2.22% and 1.57% in 5ms, 10 ms, 15 ms and 20 ms respectively. For female speaker at Gaussian mixture; the percentage of boundary deviation in initial time boundary are 5.71%, 1.58%, 1.08% and 0.95%, in end time boundary are 5.18%, 1.61%, 1.12% and 1.02% and in both initial and final time boundaries are 12.83%, 3.60%, 1.93% and 1.32% in 5ms, 10 ms, 15 ms and 20 ms respectively.

As shown in Table 6.7, best results with minimum percentage boundary deviation are found in different probability distribution at phase2 with Gaussian mixture value 2 and with Gaussian mixture value 4 for male speaker and female speakers respectively. For male speaker; the percentage of boundary deviation in initial time boundary are 6.29%, 2.09%, 0.81% and 0.44%, in end time boundary are 3.78%, 1.11%, 0.61% and 0.39% and in both initial and final time boundaries are 12.23%, 3.87%, 1.67% and 0.87% in 5ms, 10 ms, 15 ms and 20 ms respectively. For female speaker; the percentage of boundary deviation in initial time boundary are 3.78%, 0.50%, 0.07% and 0.06%, in end time boundary are 3.29%, 0.46%, 0.07% and 0.06% and in both initial and final time boundaries are 8.29%, 2.15%, 0.65% and 0.26% in 5ms, 10 ms, 15 ms and 20 ms respectively.

As shown in both Table 6.6 and Table 6.7, the percentage of boundary deviation is still more in initial time boundaries of phonemes as compared to final time boundaries of them. The performance improvement also differed in male and female speakers. Like the previous two experiments, minimum percentage of boundary deviation is achieved at phase2 and the percentage of boundary deviation also decreases rapidly when the boundary of deviation from 5ms to 20ms.

Generally in experiment III, the automatic speech segmentation performance becomes improved up to some Gaussian mixture values since it considers different probability density distribution unlike experiment I and II. The performance improvement also differed in male and female speakers. Finally, we have concluded that best results found in tied state with Gaussian mixture 2 for male speaker and with Gaussian mixture 4 for female speaker.

6.4.2.2 Error analysis

Error analysis is given by identifying phonemes that score badly from the best scored results at phase2 in context dependent Gaussian mixture 2 and Gaussian Mixture 4 for male and female speakers respectively. It is observed that the phoneme time boundaries vary whenever there is transition from one phoneme to another phoneme. It implies that phoneme time boundaries correspond to abrupt changes in the acoustic signal.

Even though this error analysis requires further study with respect to different phoneme contexts and phonetic transition between phonetic classes/catagories[77, 78], the variability of the different boundaries is analyzed with the boundaries grouped in terms of the broad phonetic classes of the respective phonemes. The boundary deviations of phonemes depend on the context that they have used, most errors occurred beyond 5ms tolerance values are plosives phonemes, nasals phonemes and silence. Most phonemes occurred beyond tolerance values of 10ms and 15ms are silence and shifting of phoneme boundaries with refence to manuallly labeled phoneme boundaries. It is observed that almost all phonemes are included in tolrance values of 20ms but the phonemes resulted beyound 20ms values are due to misalignment dueto recognition problems in HTK toolkit and transcription errors.

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATION

This chapter presents the conclusions drawn from the findings of the experiment and the researcher‟s recommendations on further actions that can be taken and future research areas.

7.1 Conclusion

The overall focus of the thesis work is speech segmentation at phoneme level and that is achieved by identifying the boundaries between phonemes in a continuous speech signal. These segments or phonemes are recognizable and meaningful parts of speech utterances.

As phoneme based speaker dependent Hidden Markov Model is the most commonly used model for automatic speech segmentation[26], it is applied to our research. The HMM model with three emitting states and two non emitting states without skipping is used to model individual Amharic phonemes. MFCC feature vectors together with their first and second, namely MFCCs + delta + delta-deltas are selected for individual HMM models. The delta and delta-delta coefficients are included to make the model sensitive to the dynamic behavior of the signal[5]. HTK toolkit is used to implement the HMM model in two phases. These phases differ in basic units that exist in the lexicon which is used as pronunciation dictionary; the second phase unlike the first phase includes epithetic vowels of Amharic language while the first phase is built with direct transliteration of Amharic words into their corresponding Latin representations. In both phases three experiments are conducted; automatic speech segmentation in context independent, context dependent with single Gaussian mixture and context dependent with multiple Gaussian mixtures.

The automatic speech segmentation system is evaluated with manual segmentation results by comparing automatic segmented phonemes to manually labeled phonemes with their time boundaries. The evaluation was done in terms of boundary deviations within tolerance values of 5ms, 10ms, 15 ms and 20ms.

The evaluation of the experiments shows that the percentage of boundary deviation minimizes as we go from context independent to context dependent and from context dependent with single Gaussian mixtures to context dependent with multiple Gaussian mixtures due to considerations of phonemes context and different probability density functions per state respectively. Finally best performance with minimum percentage of time boundary deviations are achieved at phoneme based speech segmentation in context dependent with Gaussian mixture 2 and Gaussian mixture 4 for male speaker and female speaker respectively.

For male speaker, the percentage of boundary deviation obtained with initial time boundary are 6.29%, 2.09%, 0.81% and 0.44%, with end time boundary are 3.78%, 1.11%, 0.61% and 0.39% and with both initial and final time boundaries are 12.23%, 3.87%, 1.67% and 0.87% in 5ms, 10 ms, 15 ms and 20 ms respectively.

For female speaker, the percentage of boundary deviation obtained with initial time boundary are 3.78%, 0.50%, 0.07% and 0.06%, with end time boundary are 3.29%, 0.46%, 0.07% and 0.06% and with both initial and final time boundaries are 8.29%, 2.15%, 0.65% and 0.26% in 5ms, 10 ms, 15 ms and 20 ms respectively. At the end, the test result shows that the percentage of boundary deviation also differs in each speaker even within initial and final time boundaries of Amharic phonemes.

7.2 Recommendations

As pointed out in the conclusion of the study, phoneme level automatic speech segmentation for Amharic language is done and results with minimum boundary deviation is achieved. However, there are also lots of works remaining in the area. In this section, we would like to forward the following recommendations:

- Obviously speech database with phoneme boundaries are required for other speech research areas like speech synthesis. Thus, we recommend to prepare database with very large corpus size and store in a reliable database. Manual speech segmentation is required in order to check its correctness to approve its reliability by other experts.
- Further research is also required on the precision of phoneme boundaries and their consistency in different phoneme contexts and phonetic transitions between phonetic

catagories like transition from nasal to semivowel, semivowel to vowel, semivowel to vowel, nasal to silence,stop to fricative, stop to silence, vowel to vowel and nasal to nasal using HMM model[77, 78].

- It is also required to conduct a research on automatic speech segmentation using ANN approach. This approach takes different input features like pitch, duration and amplitude and it resolves phoneme matching problems with reference to manual segmentation [46].
- It is also required to conduct this research using supervised method of automatic phoneme segmentation by implementing various algorithms[9]. These algorithms can use different frame sizes with the prior knowledge about time boundaries of phonemes. This supervised method is also useful to compare its results with the results presented in our research.
- It is clear that individual Amharic phonemes are modeled with three emitting and two non emitting HMMs. Since the duration of phonemes is variable, we recommend to continue the research with non uniform HMM topology for acoustic models[30].
- Since the research is conducted with two speakers (one male and one female speakers), automatic speech segmentation without speaker dependent is very essential. This speaker independent automatic speech segmentation is expected to improve the performance of speech segmenter through speaker adaptation techniques[71].
- Conduct researches for other Ethiopic languages by adopting this thesis work.

References

[1] M. M. A. Muhammad Jamil Anwar, Shahid Masud, and Shafay Shamail, "Automatic Arabic Speech Segmentation System," *International Journal of Information Technology*, 2006.

[2] R. Okko, "*Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture"*, 2007.

[3] D. Steinberg, "*Phonemic Structure of Pre-Exilic, Tiberian and Israeli Hebrew Contrasted*, ," 2010.

[4] K. Lodge, *A Critical Introduction to Phonetics*, 2009.

[5] S. S. A. Archana Balyan1, Amita Dev3, "phonetic segmentation based on HMM of Hindi speech," 2010.

[6] D. F. A. M. O. plero Cosi, "A preliminary statistical evaluation of manual and automatic segmentation discrepancies," *Italy,* 2009.

[7] M. D. C. C. Dours, H. Kabr, J.M. pcatte,G.prennou and M.Vigoroux, "A multi-level automatic segmentation system:SAPHO and VERIPHONE," *preceedings of EUROSPEECH 89, Paris,france,* vol. 2, pp. 83-89, 1989.

[8] T. S. A. K. Vale, "Automatic alignment of phonetic labels with continous speech," *preceedings of ICSP-90, kobe,Japan,* vol. 2, pp. 997-1000, 1990.

[9] F. D. Brugnara F. , and Omologo M. , "Automatic Segmentation and Labeling of Speech Based on Hidden Markov Models," *Speech Communication,* vol. 12, pp. 357-370, 1993.

[10] F. D. Brugnara F., and Omologo M., "Automatic segmentation and labeling of speech based on hidden Markov models," *Speech Commun. ,* vol. 12, pp. 357-370, 1993.

[11] A. C. A. Kim Y.-J., "Automatic segmentation combining an HMM-based approach and spectral boundary correction," *in Proceedings of International Conference on Spoken Language Processing, Denver, CO,* pp. 145-148, 2002.

[12] B. L. Pellom, and Hansen, J. H. L., "Automatic segmentation of speech recorded in unknown noisy channel characteristics," *Speech Commun.25,* pp. 97-116, 1998.

[13] Y. C. A. Q. Wang, "A Speaker Based Unsupervised Speech Segmentation Algorithm Used in Conversational Speech," *Springer-Verlag Berlin Heidelberg 2007,* pp. 396–402, 2007.

[14] V. W. A. M. E. Odette Scharenborga, "*Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries*," 2009.

[15] T. H. Jianhua, H.U., "Syllable Boundaries based Speech Segmentation in Demi-Syllable Level for Mandarin with HTK," 2002.

[16] E. A. B. R. Maged Nofal, Hadia El Henawy and Nemat S. Abdel Kader, "Arabic Automatic Segmentation System and its application for Arabic Speech Recognition System," 2004.

[17] K. D. A. T. Laureys. (2010). *A Comparison of Different Approaches to Automatic Speech Segmentation*. Available: http://www.esat.kuleuven.ac.be/~spch.

[18] S. Nefti and O. Boëffard, "Acoustical and topological experiments for an HMM-based speech segmentation system," 2001.

[19] M. R. H. A. B. Nath, " StockMarket Forecasting Using Hidden Markov Model: A New Approach," 2005.

[20] R. B. S. Cox, and P. Jackson, "Techniques for accurate automatic annotation of speech waveforms," *in Proceedings of the International Conference on Spoken Language Processing, Vol V., Sydney, NSW,* pp. 1947-1950, 1998.

[21] J. K. A. A. Black, "CMU ARCTIC databases for speech synthesis," *Tech Rep. CMU-LTI-03-177, CMU Language technologies Institute,* 2003.

[22] C. B. John Kominek, and Alan W Black, "Evaluating and correcting phoneme segmentation for unit selection synthesis," *in Proceedings of Eurospeech, Geneva, Switerzland,* 2003.

[23] S. N. A. O. Bo¨effard, "Acoustical and topological experiments for an HMM-based speech segmentation system," 2001.

[24] S. N. Abhinav Sethy, "Refined speech segmentation for concatenative speech synthesis," *University of Southern California,* 2002.

[25] K. K. A. R. W. King, "Automatic accent classification of foreign accented Australian," *Australia,* 2006.

[26] A. L. Iosif Mporas, Todor Ganchev and Nikos Fakotakis, "Using Hybrid HMM-based Speech Segmentation to Improve Synthetic Speech Quality," 2009.

[27] M. I. A. A. M. M. A.-G. Mohammed A. Al-Manie, "Arabic speech segmentation: Automatic verses manual method and zero crossing measurements," *Indian Journal of Science and Technology,* vol. Vol. 3 No. 12, Dec 2010.

[28] T. N. A. K. K. Svenden, "Automatic alignment of phonemic labels in continous speech," *Telenor Research COST 249, Nacy, March 6-7* 1995.

[29] B. P. Sarah Hoffmann, "Fully Automatic Segmentation for Prosodic Speech Corpora," *Interspeech 2010,Speech Processing Group, ETH Z¨urich, Switzerland,* 2010.

[30] J. C. B. Kalu U. Ogbureke, "Improving Initial Boundary Estimation for HMM-based Automatic Phonetic Segmentation," *School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland,* 2009.

[31] E. B. Y.K. Muthusamy, and R.A. Cole. (October 1994). *Reviewing Automatic Language Identification*.

[32] T. Dutoit, "An Introduction to Text-to-Speech Synthesis," *Kluwer Academic Publishers,* 1997.

[33] P. Z. Iosif Mporas, Nikos Fakotakis "Broad Phonemic Class Segmentation of Speech Signals in Noise Environments " *Artificial Intelligence Group, Wire Communication Laboratory Department of Electrical and Computer Engineering, University of Patras* 2010.

[34] Y. S. A. Y. Lee, "Phoneme segmentation of continuous speech using multi-layer perceptron," *Electronics and Telecommunications Research Institute, KOREA,* 1996.

[35] Z. Li, Z. Wu, Y. He, and C. Fulei, "Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery," *Mechanical Systems and Signal Processing,* vol. 19, pp. 329-339, 2005.

[36] A. J. H. H. B.L. Pellom, "Automatic segmentation of speech recorded in unknown noisy channel characteristics," *Speech Communication,* vol. 25, pp. 97-116, 1998.

[37] L. R. Rabiner, "A Tutorial on Hidden Markov Mod- els and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, pp. 257-285, Feb. 1989.

[38] J. Makhoul, Roucos, S., Gish, H., , "Vector Quantiza- tion in Speech Coding," *Proc. IEEE,* vol. 73, pp. 1551-1585, Nov,1985.

[39] L. A. H. G. A. L. V. G. D.T. Toledano, "Automatic Phonetic Segmentation," *Trans. Speechand Audio Proc.,* vol. 14, pp. 617-625, Nov.2003.

[40] R. B. S. Cox, and P. Jackson, "Techniques for accurate automatic annotation of speech waveforms," *in Proceedings of the International Conference on Spoken Language Processing,Sydney, NSW,* vol. V, pp. 1947-1950, 1998.

[41] O. D. F. Malfrere, and T. Dutoit, "Phonetic alignment: Speech synthesis based vs. hybrid HMM/ANN," *in Proceedings of the International Conference on Spoken Language Processing,Sydney, NSW,* vol. *I*V, pp. . 1571-1574, 1998.

[42] H. A. M. Moustafa Elshafei Mohammad Ali, and Mansour Al-Ghamdi, "Automatic segmentation of Arabic speech," 2005.

[43] P. W. S. Matthew E. Dunnachie, David H.Crowford, and Mike Davies, "Filler Models for Automatic Speech Recognition Created from Hidden Markov Models using the K-Means Algorithm," *Proceedings of 17th European Signal processing conference (EUSIPCO),* 2009.

[44] J. A. Markowitz, "Using Speech Recognition," *Upper Saddle River, New Jersey: Prentice Hall, Inc., 1996* . 1996.

[45] J. V. L. Petr Poll´ak, Radek Skarnitzl, "Phone Segmentation Tool with Integrated Pronunciation Lexicon and Czech Phonetically Labelled Reference Database," 2008.

[46] E. Keller, "Neural network motivation for segmental distribution. Proceedings of 5th International Conference on Spoken Language Processing,". *Paper 937. Sydney, Australia,* 1998.

[47] M. N. M. Huda, Ghulam / Horikawa, Junsei / Nitta, Tsuneo "Distinctive phonetic feature (DPF) based phone segmentation using hybrid neural networks," *In INTERSPEECH-2007,* pp. 94-97, 2007.

[48] K. J. A. D. Murphy, "Segmenting Melodies into Notes " 2001.

[49] C. Bishop, "Neural networks for pattern recognition," *Oxford University Press, Oxford,* 1995.

[50] V. K. Santhosh Yuvaraja, Sathish Chandra Pammi, Kishore Prahallad, Alan W Black, "BUILDING A TAMIL VOICE USING HMM SEGMENTED LABELS," 2008.

[51] V. S. C.S. Kumar, N. Udhyakumar, and R. Srinivasan, "Rule-based automatic grapheme to phoneme conversion for Tamil," *in Proc. ICSLT, Delhi, India,* 2004.

[52] A. P. P. M. Anthony Psaila, "Building an Automatic Speech Annotation System," 2008.

[53] A. C. M. Pierre Lanchantin, Xavier Rodet, Christophe Veaux, "Automatic Phoneme Segmentation with Relaxed Textual Constraints," *IRCAM, Analysis-Synthesis Group1, place Igor Stravinsky, F-75004 Paris, France,* 2008.

[54] Corpatext. (2006). *Corpatext 1.02.* Available: www.lexique.org/public.Corpatext.php.

[55] G. E. S. Young, D. Kershaw, G. Moore, J.Odell,D. Ollason, V.Valtchev, and P.Woodland., *The HTK Book*, 2002.

[56] I. Anees, *EI-Aswat Allaghawia*, 1981.

[57] K. D. A. T. Laureys, "A Comparison of Different approaches to Automatic Speech Segmentation," *In Proc. 5th International Conference on Text, Speech and Dialogue, Brno, Czech Republic* pp. 277-284, September 2002.

[58] L. M. A. F. C. Bender, *The Ethiopian Writing System*, 1976.

[59] Y. Baye, *የአማርኛሰዋስው*1986.

[60] S. P. K. Sebsibe H/Mariam , Alan W Black, Rohit Kumar, and Rajeev Sangal, , "Unit Selection Voice for Amharic using Festivox," *5th ISCA Speech Synthesis Workshop –99 Pittsburgh,* pp. 103-107, ,2005.

[61] Y. Baye, *የአማርኛሰዋስው*1987.

[62] A. Getahun, *ዘመናዊየአማረኛሰዋስው በቀላል አቀራረብ*, 1989.

[63] Y. Baye, *የአማርኛሰዋስው*1997.

[64] W. M. Solomon Teferra Abate, Bairu Tafila, "An Amharic Speech Corpus for Large Vocabulary Continuous Speech Recognition."

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

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

[67] G. Hudson, "Phonology of Ethiopian Languages," *the Handbook of Phonological Theory. Glodsmith, John A. Blackwell Publishing.,* 1996.

[68] C. Jany, Gordon, M., Nash, C. M., and Takara, N., "How Universal Is The Sonority Hierarchy?," *A Cross-Linguistic Acoustic Study. In proceedings the 16th International Congress of Phonetic Sciences. Saarbrücken, Germany,* 2007.

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

[70] M. Seyoum, "The syllable Structure and Syllablification in Amharic," *Masters of philosophy in general linguistic thesis. Department of Linguistics, Trondheim, Norway,* 2001.

[71] D. T. Toledano, Gomez, L.A.H. and Grande, L.V., "Automatic phonetic segmentation," *in IEEE Transactions on Speech and Audio Processing,* pp. 617-625, 2003.

[72] E. Keller, *Fundamentals of phonetic science*, 1994.

[73] O. D. F. Malfrere, and T. Dutoit, "Phonetic alignment: Speech synthesis based vs. hybrid HMM/ANN," *in Proceedings of the International Conference on Spoken Language Processing, Vol IV.,Sydney, NSW,* pp. 1571-1574, 1998.

[74] J. H. A. Ljolje, and J.P.H. Van Santen, "Automatic speech segmentation for concatenative inventory selection," *in Progress in Speech Synthesis, J.P.H. Van Santen, Ed: Springer,* pp. 305-311, 1997.

[75] J. Tebelskis, "Speech Recognition using Neural Networks," *CMU-CS-95-142,* p. 92, 1995.

[76] G. C. Vincent Pollet, "Statistical Corpus-based speech Segmentation," *Text-To-Speech Department Scansoft Belgium,* 2004.

[77] A. E. A. G. Aversano, "Text Independent Methods for Speech Segmentation " 2005.

[78] G. K. Ladon Baghai-Ravary, John Coleman, "Precision of phonemes Boundaries Derrived using Hidden Markov Models," *INTERSPEECH 2009 BRIGTON,* 2009.

Appendixes

Appendix A : Amharic alphabets (adopted from [58])

Order							Labialised				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ			ዧ		
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ			ዯ		
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዸ	ዹ	ዺ	ዻ	ዼ
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ			ጇ		
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ			ጷ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ					

Appendix B: Amharic phonetic list, IPA Equivalence and its ASCII Translation table (adopted in [62])

IPA	Transcription	Amharic equivalence
Consonants		
[p]	[p]	ፐ
[t]	[t]	ተ
[k]	[k]	ከ
[ʔ]	[ax]	አ
[b]	[b]	በ
[d]	[d]	ደ
[g]	[g]	ገ
[pʼ]	[px]	ጰ
[tʼ]	[tx]	ጠ
[cʼ]	[cx]	ጨ
[q]	[q]	ቀ
[f]	[f]	ፈ
[s]	[s]	ሰ
[ʃ]	[sx]	ሸ
[h]	[h]	ሀ
[sʼ]	[xx]	ጸ
[tʃ]	[c]	ቸ
[gʼ]	[j]	ጀ
[m]	[m]	መ
[n]	[n]	ነ
[nʼ]	[nx]	ኘ
[l]	[l]	ለ
[r]	[r]	ረ
[j]	[y]	ይ
[w]	[w]	ወ
[v]	[v]	ቨ
[z]	[z]	ዘ
[zʼ]	[zx]	ዠ
Vowels		
[ɛ]	[e]	ኧ
[u]	[u]	ኡ
[ɪ]	[ii]	ኢ
[ɑ]	[a]	ኣ
[e]	[ie]	ኤ
[ɨ]	[ix]	ኧ
[o]	[o]	ኦ

Appendix C: Python code used for ASCII transliteration

```
import codecs, sys, string

worddict = {}

mapfile = codecs.open(r'C:/Users/eshete/Desktop/latin/char_table.txt', 'r', 'utf-8')

corfile = codecs.open(r'C:/Users/eshete/Desktop/doc1query.txt', 'r', 'utf-8')

outfile = codecs.open(r'C:/Users/eshete/Desktop/esheteTranslation1.txt', 'w', 'utf-8')

maps = mapfile.read().encode("utf-8")

corpus = corfile.read().encode("utf-8")

def autodecode( maps ):
   if maps.startswith(codecs.BOM_UTF8):
      out = maps.decode( "utf8" )
      return out[1:]
   else: return maps.decode( "ascii" )
mapstripped = autodecode(maps)

corstripped = autodecode(corpus)

for line in mapstripped.split("\n"):
   (i,j) = line.split()
   worddict[i] = j
for char in corstripped:
   if worddict.__contains__ (char):
      outfile.write(worddict[char])
   else:
      outfile.write(" ")
outfile.close()

"""
for i in worddict.iterkeys():
   outfile.write(i)
"""
```

Appendix D: Sonority scale of Amharic consonants (adopted from[66])

Class	category	Amharic Phonemes	Sonority scale
stops	**Voiceless**	p, t, k, kwa, ax	**1**
	Voiced,	b, d, g, gwa	**2**
	Glottalized	px, tx , q, qwa	**3**
Affricatives	**Voiceless**	f, s, sx, h	**4**
	Voiced,	v, z, zx	**5**
	Glottalized	xx, hwa	**6**
Fricatives	**Voiceless**	c	**7**
	Voiced	j	**8**
	Glottalized	cx	**9**
Nasals	**-**	m, n, nx	**10**
liquids	**-**	l, r	**11**
Glides	**-**	w, y	**12**

Appendix E: Summary of the epenthesis vowel insertion procedure (adopted from[66]):

Rule #	Position	Observed Sequence	Epenthesis	Exception
1	final	#CC	#CixC	If the first phoneme is consonant and the next consonant is glide /w/
2	medial or initial	CCC	CCixC	If sonority of the middle consonant is greater than The rest (CixCC)
3	medial or initial	C1C1C (CC:)	C1C1ixC (C:ixC)	
4	medial or initial	CC1C1(CC:)	CixC1C1 (CixC:)	
5	medial or initial	C1C1C2C2 (C:C:)	C1C1ixC2C 2 (C:ixC:)	
6	final	CC#	CixC#	If the sonority of the last phoneme is less or equal to the preceding

Appendix F: Epenthetic Vowel insertion algorithm or procedure (adopted from[66])

1. Accept input word and scan from left to right.

2. If consonant cluster occurs at word initial position, insert epenthetic vowel between them.

***Exception**: If the first phoneme is consonant and the next consonant is glide /w/. (Rule #1)*

3. If three consonants are appeared in sequence word medially or word final position, insert epenthetic vowel before the third consonant. (Rule #2)

***Exception**: If the middle consonant sonority is greater than the rest insert epenthetic vowel after the first consonant in the cluster.*

4. If a cluster of consonants contains the geminate and singleton in sequence, insert epenthetic vowel after the geminated consonants.(Rule #3)

5. If a cluster of consonants contains the singleton and geminate in sequence, insert epenthetic vowel after the singleton consonants. (Rule #4)

6. If a cluster of consonants contains two different geminates in sequence, insert epenthetic vowel between the two geminate consonants. (Rule #5)

7. If the sonority of the final consonant is greater than that of the preceding consonant, the epenthetic vowel is inserted between the final consonant clusters. (Rule #6)

8. Repeat 2 up to 7 until all the phonemes are parsed in the phonemes list.

Appendix G: Sample Amharic text corpus

1 የነቀምቴ ስታዲዮም ግንባታ ስልሳ በመቶ ተጠናቀቀ::
2 አጠደ በዱባይ ማራቶን ለድል ከሚጠበቁት አትሌቶች አንድዋ ነች::
3 ጥሩነሽ በዔደንበራ አገር አቋራጭ ውድድር አሸነፈች::
4 በኢትዮጵያ የኔትቦል ስፖርት እንዲስፋፋ እንግሊዝ ትደግፋለች::
5 የመቀሌ ስታዲዮም የመጀመሪያው ራፍ ግንባታ ተጠናቀቀ::
6 ኃይሉ በቶኪዮ ማራቶን አሸነፈ::
7 ጠይባ በቦስተን ማራቶን ለድል ትጠበቃለች::
8 የፌዴራል ማረሚያ ቤቶች ስፖርት ክለብ አዲስ የስራ አስኪያጅ ኮሚቴ መረጠ::
9 በቃና የሂዩስተን ማራቶንን ክብረ ወሰን በማሻሻል አሸነፈ::
10 ኃይሌ በማንቸስተር የጎዳና ሩጫ ለድል ይጠበቃል::
11 ብዙነሽ በሙምባይ ማራቶን ለድል ትጠበቃለች::
12 ፌዴሬሽኑ ባጠደቃቸው መመሪያዎች ከባለድርሻ አካላት ጋር ተወያየ::
13 ኃይሌ በኒውዮርክ ግማሽ ማራቶን ውድድር ይካፈላል::
14 ሁነኛው በስፔን የአገር አቅዋራጭ ውድድር አሸነፈ::
15 በላሊበላ ከተማ ታላቁ ሩጫ ተካሄደ::
16 ከአፍሪካ ሃያ አምስት ምርጥ ስፖርተኞች ኢትዮጵያውያን ግንባር ቀደም ስፍራ ይዘዋል::
17 ገንዘቤ በጌንት የአንድ ሺ አምስት መቶ ሜትር ሩጫ አሸነፈች::
18 ሲራጅ በሮም የማራቶን ውድድርን በባዶ እግሩ በማጠናቀቅ ታሪክ አስመዘገበ::
19 ኢትዮጵያ ለዶኃ ውድድር በመሰረት ደፋር የሚመራ ቡድን ትልካለች::
20 ታዋቂ አትሌቶች በሚገኙበት በኃዋሳ ከተማ የሩጫ ውድድር ይካሄዳል::
21 ለወልድያ ስታዲዮም ግንባታ የሚውል ገቢ ማሰባሰብ ተጀመረ::
22 ፀጋየ ከበደ በለንደን ማራቶን አሸነፈ::
23 ኃይሌ የታላቁ ማንቸስተር ሩጫ ውድድርን አሸነፈ::
24 ቀነኒሳ በዶኃው የዳይመንድ ሊግ ውድድር አይሳተፍም::
25 ኢትዮጵያ በኦሎምፒክ ለመሳተፍ ዝግጅት እያደረገች ነው::
26 በአገር አቀፍ ደረጃ በተለያዩ የስፖርት አይነቶች ስልጠና እየተሰጠ ነው::
27 ስፔን የአለም ዋንጫን ያሸነፈችባት ቅዋስ በጨረታ ሰባ አራት ሺህ ዶላር አወታች::
28 ለአለም ዋንጫ በኮከብነት አስር ተጫዋቾች ታጩ::
29 ኢትዮጵያ በአስራ ሶስተኛው የአለም ወጣቶች ሻምፒዮና የአምስተኛ ደረጃን አገኘች::
30 ኢትዮጵያ በሞስኮ በተካሄዱ የሩጫ ውድድሮች አሸነፉ::
31 የኢትዮጵያ ብሄራዊ የእግር ክዋስ ቡድን አሰልጣኝ እንግሊዛዊ ነው::
32 አመታዊ የአዲስ አበባ የክለቦች ብስክሌት ሻምፒወና የፍጣሜ ውድድር ተካሄደ::
33 የአዲስ አበባ የዱላ ቅብብል ውድድር ሰኔ ላይ ይካሄዳል::
34 የአዳማ ዩኒቨርስቲ ለታዳጊ ወጣቶች የስፖርት ስልጠና እየሰጠ ነው::
35 የአለም የወጣቶች ኦሎምፒክ ሻምፒዮና በሲንጋፖር እየተካሄደ ነው::
36 ታሪኩ በበርሊን የሶስት ሺህ ሜትር አሸነፈ::
37 ታደሰ በሊዝበን ግማሽ ማራቶን አሸነፈ::
38 ደቡብ አፍሪካ ያስተናገደችው የአለም ዋንጫ ታላቅ ውጤት የተመዘገበበት እንደነበር ፊፋ ገለጠ::
39 የመቀሌ ስታዲዮም እድሳት እየተደረገለት ነው::
40 መሰረት የአመቱ ምርጥ አትሌትነት ምርጫን በመምራት ላይ ነች::
41 የኢትዮጵያ እግር ክዋስ ፌዴሬሽን ጠቅላላ ጉባዔ ተጀመረ::
42 የጋራ ብልጥግና አገሮች የስፖርት ሻምፒዮና ተጀመረ::
43 የዘንድሮው የታላቁ ሩጫ ውድድር ምዝገባ ተጠናቀቀ::
44 መንግስቱ ወርቁ ከዚህ አለም በሞት ተለየ::
45 አዝመራውና ሱሌ የታላቁ ሩጫ ውድድርን አሸነፉ::
46 ኃይሌ ሩጫ የሚያቆምበትን ትክክለኛ ጊዜ እንደማያውቀው ተናገረ::
47 ፌዴሬሽኑ ብጥብጥ ባስነሱ ወገኖች ላይ ተገቢውን እርምጃ እንደሚወስድ ገለጠ::
48 ኃይሌ የአለማችን የምንጊዜም ምርጥ ወንድ አትሌትነትን ምርጫ በድምጥ ብልጫ እየመራ ነው::
49 አቡትሪካ የአፍሪካ የአመቱ ኮከብ ተጫዋች ተብሎ ተሸለመ::
50 ሀገር አቀፉ የከፍተኛ ትምህርት ተቅዋማት የስፖርት ፌስቲቫል በሰንደር ተጀመረ::

Appendix H: Grammar file

$Sentence=yeneqemtie sixtadiiyom gixnbata sixlsa bemeto tetxenaqeqe | bixzunesx bemumbay maraton ledixl tixtxebeqalec | leweldixya sixtadiiyom gixnbata yemiiwl gebii masebaseb tejemere| yeiityopxya bhierawii yeixgixr kwas budixn aseltxanx ixngixliizawii new| yeiityopxya ixgixr kwas fiedieriesxixn txeqlala gubaie tejemere | yekixlloc yesport wixddixr mixrt sixportenxocn lemafrat yascixlal | yeiityopxya ixgixr kwas fiedieriesxixn txeqlay gubaie yeaqwam meglecxa awetxa | iityopxya beasixr sxiih mietixr yewendoc wixddixr asxenefec | yealmeda cxerqa cxerq fabriika yeixgixr kwas kixleb abalat tesxelemu | abebe dixnqiesa benayjeriiya yemawnixtien ries wixddixr asxenefe | beiityopxyana bealem bank mekakel yebdixr sixmmixnetoc teferemu | yeixngixliiz balehabtoc beiityopxya iinvest lemadreg fixlagot ixndalacew gelexxu . . . .| tekesasxu kixrsixtyan hono sayfata bebietu ixqubat yasqemetxe beixsrat yixqetxal;

({NS_B} $Sentence {NS_E})

Appendix I: Generated Lattice Format

VERSION=1.0
N=7347 L=8247
I=0 W=NS_E
I=1 W=!NULL
I=2 W=yixqetxal
I=3 W=beixsrat
I=4 W=yasqemetxe
I=5 W=ixqubat
I=6 W=bebietu
I=7 W=sayfata
I=8 W=hono
I=9 W=kixrsixtyan
I=10 W=tekesasxu
I=11 W=aye
I=12 W=fixdawixn
I=13 W=teyzo
I=14 W=bepoliis
I=15 W=adragiiw
I=16 W=asmat
I=17 W=yixgelexxna
I=18 W=hulu
I=19 W=miistxixru
I=20 W=aweqec
I=21 W=giiyorgiis
I=22 W=melke
I=23 W=iiyesusna

.

.

.

J=8239 S=7340 E=7339
J=8240 S=7341 E=7340
J=8241 S=7342 E=7341
J=8242 S=7344 E=7342
J=8243 S=7344 E=7343
J=8244 S=7343 E=7344
J=8245 S=7346 E=7344
J=8246 S=1 E=7345

Appendix J: Phone based dictionary output file

NS_B	[ ]	sil
NS_E	[ ]	sil
ab	[ab]	a b sp
aba	[aba]	a b a sp
abal	[abal]	a b a l sp
abalat	[abalat]	a b a l a t sp
abarii	[abarii]	a b a r ii sp
abat	[abat]	a b a t sp
abay	[abay]	a b a y sp
abeba	[abeba]	a b e b a sp
abeje	[abeje]	a b e j e sp
aberetac	[aberetac]	a b e r e t a c sp
abietuta	[abietuta]	a b ie t u t a sp
ablixtxa	[ablixtxa]	a b l ix tx w a sp
abnet	[abnet]	a b n e t sp
abnetu	[abnetu]	a b n e t u sp
abrixham	[abrixham]	a b r ix h a m sp
abutriika	[abutriika]	a b u t r ii k a sp
abzanxaw	[abzanxaw]	a b z a nx a w sp
acaw	[acaw]	a c a w sp
adebabay	[adebabay]	a d e b a b a y sp
adega	[adega]	a d e g a sp
adege	[adege]	a d e g e sp
adegenxa	[adegenxa]	a d e g e nx a sp
adelem	[adelem]	a d e l e m sp
adere	[adere]	a d e r e sp
aderege	[aderege]	a d e r e g e sp
aderegu	[aderegu]	a d e r e g u sp
aderoc	[aderoc]	a d e r o c sp
aderu	[aderu]	a d e r u sp
aderun	[aderun]	a d e r u n sp
adiis	[adiis]	a d ii s sp
adiisu	[adiisu]	a d ii s u sp
admixtxu	[admixtxu]	a d m ix tx w u sp
adragiiw	[adragiiw]	a d r a g ii w sp
adrixgachu	[adrixgachu]	a d r ix g a c h u sp
.		
.		
zixq	[zixq]	z ix q sp
zixryawoc	[zixryawoc]	z ix r y a w o c sp
zon	[zon]	z o n sp
zonal	[zonal]	z o n a l sp

Appendix K: The configuration parameter used at coding

```
#coding parameters
SOURCEFORMAT = WAVE
TARGETFORMAT = HTK
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

Appendix L: The Prototype HMM

~o <VecSize> 39<MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
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> 39
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
<State> 3
<Mean> 39
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> 39
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
<State> 4
<Mean> 39
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> 39
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
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>

Appendix M: Sample Output of automatic speech segmentation without tied state

#!MLF!#
Sentence1.lab

Initial time	End time	Phoneme	word
0	1500000	sil	silent
1500000	2100000	y	yeneqemtie
2100000	2400000	e	
2400000	2900000	n	
2900000	3600000	e	
3600000	4600000	q	
4600000	5100000	e	
5100000	5800000	m	
5800000	6500000	t	
6500000	7300000	ie	
	etc		

Appendix N: Tree.hed script

```
RO 100.0 stats

TR 0

QS "R_NonBoundary" { *+* }
QS "L_NonBoundary" { *-* }
QS "R_Silence" { *+sil }
QS "R_Silence" { sil-* }
QS "L_Stops" {b-*,d-*,g-*,p-*,t-*,k-*,px-*,tx-*,*+q}
QS "R_Stops" {*+b,*+d,*+g,*+p,*+t,*+k,*+P,*+tx,*+q}
QS "L_Fricatives" {v-*,z-*,Z-*,f-*,s-*,sx-*,h-*,*+x}
QS "R_Fricatives" {*+v,*+z,*+zx,*+f,*+s,*+sx,*+h,*+x}
QS "L_Affricates" {j-*,C-*,*+cx}
QS "R_Nasal" { *+m,*+n, *+nx}
QS "L_Nasal" { m-*,n-*, nx-*}
QS "R_Liquid" { *+l,*+r}
QS "L_Liquid" { l-*,r-*}
QS "R_semivowels" {*+w,*+y }
```

```
QS "L_semivowels" {w-*,y-* }
QS "R_Vowel" { *+e,*+ii,*+o,*+u,*+a,*+ie,*+ix }
QS "L_Vowel" { e-*,ii-*,o-*,u-*,a-*,ie-*,ix-* }
QS "L_a" {a-*}
QS "R_a" {*+a}
QS "L_b" {b-*}
QS "R_b" {*+b}
QS "L_c"  {*-c}
QS "R_c" {*+c}
QS "L_d" {d-*}
QS "R_d" {*+d}
.
.
.
QS "L_zx" {zx-*}
QS "R_zx" {*+zx}
QS "L_ii" {ii-*}
QS "R_ii" {*+ii}
QS "L_ie" {ie-*}
QS "R_ie" {*+ie}
QS "L_ix" {ix-*}
QS "R_ix" {*+ix}

TR 2

TB 350.0 "ST_sil_2_" {("sil","*-sil+*","sil+*","*-sil").state[2]}
TB 350.0 "ST_a_2_" {("a","*-a+*","a+*","*-a").state[2]}
TB 350.0 "ST_b_2_" {("b","*-b+*","b+*","*-b").state[2]}
TB 350.0 "ST_l_2_" {("l","*-l+*","l+*","*-l").state[2]}
TB 350.0 "ST_t_2_" {("t","*-t+*","t+*","*-t").state[2]}
TB 350.0 "ST_r_2_" {("r","*-r+*","r+*","*-r").state[2]}
TB 350.0 "ST_ii_2_" {("ii","*-ii+*","ii+*","*-ii").state[2]}
TB 350.0 "ST_y_2_" {("y","*-y+*","y+*","*-y").state[2]}
TB 350.0 "ST_e_2_" {("e","*-e+*","e+*","*-e").state[2]}
TB 350.0 "ST_j_2_" {("j","*-j+*","j+*","*-j").state[2]}
TB 350.0 "ST_c_2_" {("c","*-c+*","c+*","*-c").state[2]}
TB 350.0 "ST_ie_2_" {("ie","*-ie+*","ie+*","*-ie").state[2]}
TB 350.0 "ST_u_2_" {("u","*-u+*","u+*","*-u").state[2]}
TB 350.0 "ST_ix_2_" {("ix","*-ix+*","ix+*","*-ix").state[2]}
```

```
TB 350.0 "ST_tx_2_" {("tx","*-tx+*","tx+*","*-tx").state[2]}
TB 350.0 "ST_w_2_" {("w","*-w+*","w+*","*-w").state[2]}
.
.
.
TB 350.0 "ST_sx_4_" {("sx","*-sx+*","sx+*","*-sx").state[4]}
TB 350.0 "ST_xx_4_" {("xx","*-xx+*","xx+*","*-xx").state[4]}
TB 350.0 "ST_cx_4_" {("cx","*-cx+*","cx+*","*-cx").state[4]}
TB 350.0 "ST_p_4_" {("p","*-p+*","p+*","*-p").state[4]}
TB 350.0 "ST_v_4_" {("v","*-v+*","v+*","*-v").state[4]}
TB 350.0 "ST_px_4_" {("px","*-px+*","px+*","*-px").state[4]}
TB 350.0 "ST_zx_4_" {("zx","*-zx+*","zx+*","*-zx").state[4]}
TB 350.0 "ST__4_" {("","*-+*","+*","*-").state[4]}

TR 1

AU "fulllist"
CO "tiedlist"

ST "trees"
```

Appendix O: Sample output of automatic speech segmentation with tied state

#!MLF!#
Sentence1.lab

Initial time	End time	phoneme	word
0	1400000	sil	silent
1400000	2100000	y	yeneqemtie
2100000	2400000	e	
2400000	2900000	n	
2900000	3600000	e	
3600000	4600000	q	
4600000	5100000	e	
5100000	5800000	m	
5800000	6500000	t	
6500000	7300000	ie	
	etc		

Appendix P: A sample script for creating multiple mixture components

```
MU 16 {*a*.state[2-4].mix}
MU 16 {*b*.state[2-4].mix}
MU 16 {*l*.state[2-4].mix}
MU 16 {*t*.state[2-4].mix}
MU 16 {*r*.state[2-4].mix}
MU 16 {*ii*.state[2-4].mix}
MU 16 {*y*.state[2-4].mix}
MU 16 {*e*.state[2-4].mix}
MU 16 {*d*.state[2-4].mix}
MU 16 {*h*.state[2-4].mix}
MU 16 {*u*.state[2-4].mix}
MU 16 {*k*.state[2-4].mix}
MU 16 {*j*.state[2-4].mix}
MU 16 {*c*.state[2-4].mix}
MU 16 {*ie*.state[2-4].mix}
MU 16 {*tx*.state[2-4].mix}
MU 16 {*n*.state[2-4].mix}
.
.
MU 16 {*ix*.state[2-4].mix}
MU 16 {*zx*.state[2-4].mix}
MU 16 {*sil*.state[2-4].mix}
```

Appendix Q: Sample output of automatic speech segmentation with tied state with multiple Gaussian mixture 16

#mlf#
Sentence1.lab

Initial time	End time	Phoneme	word
0	1300000	sil	silent
1300000	2100000	y	yeneqemtie
2100000	2400000	e	
2400000	3000000	n	
3000000	3600000	e	
3600000	4600000	q	
4600000	5200000	e	
5200000	5900000	m	
5900000	6500000	t	
6500000	7300000	ie	
	Etc		

Appendix R: Sample Output of manual speech segmentation

Sentence1.lab

Initial time	End time	Phoneme	word
0	160	sil	silent
160	220	y	yeneqemtie
220	260	e	
260	290	n	
290	360	e	
450	480	q	
480	530	e	
530	580	m	
640	690	t	
690	750	ie	
	etc		

Declaration

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

ESHETE DERB

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

SEBSIBE HAILEMARIAM

November, 2011

Addis Ababa, Ethiopia