ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

DURATION MODELLING OF PHONEMES FOR AMHARIC TEXT TO SPEECH SYSTEM

By

Yonas Demeke WoldeMariam

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

March, 2012

Addis Ababa

Addis Ababa University

School of Graduate Studies

College of Natural Science

Faculty of Computer and Mathematics Sciences

Department of Computer Science

DURATION MOELLING OF PHONEMES FOR AMHARIC TEXT TO SPEECH SYSTEM

By

Yonas Demeke WoldeMariam

APPROVED BY

EXAMINING BOARD:

1. Dr. Sebsibe HaileMariam, Advisor _____

2. _____ _____

3. _____ _____

Dedication

To Jesus Christ, my personal Savior and Lord

ACKNOWLEDGEMENTS

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

There are a very large number of people to thank in connection with this work. I shall begin at the beginning, by thanking my supervisor, Dr. Sebsibe HaileMariam. To him I am deeply grateful, for his valuable guidance, advice and suggestion. To me, he is a man of knowledge and integrity. Thanks to Nadew Tademe for his provision of resources related with TTS. Thanks to Dr. Derb Ado for his professional support in phonetics, moral encouragement and provision of books. Thanks to Ozelm Ozturk for his e-mail discussion and suggestion. My Special Thanks to Ermias Sebsibe, who had generously provided me his lap top for this thesis. Finally I would like to express love, thanks, appreciation, and respect to my sister Fikirte Demeke for her caring throughout my study.

Table of Contents

List of Tables

List of Figures

List of Appendixes

List of Acronyms

ANN	Artificial Neural Network
ANOVA	Analysis Of Variance
ASR	Automatic Speech Recognition
CART	Classification And Regression Tree
CV	Consonant-Vowel
CC	Correlation Coefficient
DB	Database
DSP	Digital Signal Processing
F0	Fundamental Frequency (pitch)
G2P	Grapheme to Phoneme
IPA	International Phonetic Alphabet
LPC	Linear Predictive Coding
LR	Linear Regression
HMM	Hidden Markov Model
MOS	Mean Opinion Score
NLP	Natural Language Processing
RMSE	Root Mean Squared Error
SR	Speaker Recognition
SOP	Sums of Products
TTS	Text to Speech

ABSTRACT

Naturalness of synthetic speech highly depends on appropriate modeling of prosodic aspects. Mostly, three prosody components are modeled: segmental duration, pitch contour and intensity. The general goal of duration modeling is to find a computational relation between a set of affecting factors and the segment duration. A number of text-to-speech synthesizers for Amharic language have used synthesis techniques that require prosodic models for good quality synthetic speech. However, due to different reasons like unavailability of adequately large and properly annotated speech databases for Amharic language, prosodic models for these synthesizers have still not been developed. Hence, in this thesis work duration modeling of phonemes for Amharic speech synthesis is done.

In this thesis two major tasks have been performed, development of concatinative unit selection synthesizer and data-drive duration model. Unit selection voice has been built on Festival speech synthesis framework using phone as basic unit. We have used a speech corpus having a size of 1hour, 16 minutes and 29 seconds, labeled at phoneme level. After phonetic, prosodic, and acoustic features extraction inventory for each phone has been constructed. In order to synthesize the input text the synthesizer uses cluster unit selection algorithm adopted from Festival speech synthesis. At synthesis time units that minimize acoustically defined target and join costs are then selected from a cluster. In order to build duration model we have extracted features that affect duration of Amharic phones and the whole data is split into training (90%) set and test (10%) set, they consist 45,500 and 5500 segments, respectively. Classification and Regression trees have been used to build our duration model. The resulting model is integrated into the synthesizer.

In order to evaluate the performance and effectiveness of the duration model, we have conducted objective and subjective tests. From objective test we found correlation between actual and predicted durations is 0.3901 and the Root Mean Squared Error (RMSE) of prediction is 0.8403 in z-score domain. Subjective evaluations are done to ascertain the improvement in the quality of synthesized speech using the duration model. In this thesis, the Mean Opinion Score (MOS) evaluation technique is used. The results from the MOS were found to be 3.5 and 3.58 for

intelligibility and naturalness respectively for speeches synthesized by synthesizer with duration model. In the synthesizer without duration model the result obtained for intelligibility and naturalness are 3.31 and 3.33 respectively.

Keywords: Speech synthesis, duration modeling, unit selection, root mean square error, correlation coefficient

CHAPTER ONE
INTRODUCTION

1. 1 General Background

Speech is one of the most effective means of communication for humans. Speech is produced by air-pressure waves emanating from the mouth and the nostrils of a speaker. The source of air during speech is the lungs. Speech sounds are usually considered as either voiced or unvoiced. When the vocal folds are held close together and oscillate against one another during speech production, the sound is said to be voiced. When the vocal folds are too slack or tense to vibrate periodically, the sound is said to be unvoiced. The place where the vocal folds come together is called the glottis [1, 2].

The speaker intentionally or unintentionally communicates much of his or her attitude to or feeling about what is being said by 'modulations' of a 'neutral' prosodic element in the speech which is dictated by the grammatical nature of the utterance. By *how* the speaker speaks, rather than by *what* he or she actually says, the listener can become aware of what the speaker feels, or what the beliefs are toward what is being said. In modern phonetics the word „prosody" refers to those properties of speech that cannot be derived from the segmental sequence of phonemes underlying human utterances. Examples of such properties are the controlled modulation of the voice pitch[1], the stretching and shrinking of segment and syllable durations, and the intentional fluctuations of overall loudness [3]. The prosody of continuous speech depends on many separate aspects:

- Dialectal variation
- Variation across groups within a single dialect community

[1] pitch is used for the rate of vibration that is perceived by the listener, and in general the pitch and fundamental frequency can be taken as same concepts [4]

Individuals may use slightly different qualities for the same underlying vowel; or slightly different places of articulation for the same underlying consonant (dental vs. alveolar for /t/, for example)

- Phonologically conditioned variation within an individual speaker: - Some variability in prosodic form is determined by phonological factors, giving rise to conditioned allophones. For instance ways in which the type of pitch pattern is determined by the number and makeup of the syllables which go with it, depending on such factors as the phonological length of the vowel on which the nucleus falls and on the nature of the consonants (manner of articulation, voicing) between syllables.
- Contextually conditioned variation within an individual speaker: - Detailed empirical studies of the distribution of intonational patterns in spontaneous interaction suggest that many aspects of intonational patterning are determined by the particular interactional task that the speaker is engaged in. Intonation means how the pitch pattern changes during speech [1].

Prosody is shaped by the relative level of the fundamental frequency, the intensity and last but not least by the duration of the pronounced phones [5]. The duration of the phones controls the rhythm and the tempo of speech and the flattening of the prosody in a speech waveform would result in a monotonous, neutral and toneless. Rhythm refers to the perceived regularity of prominent units in speech. These regularities may be stated in terms of patterns of stressed versus unstressed syllables, syllable length (long versus short) or pitch (high versus low) – or some combination of these variables [7], tempo refers to speed of speaking; alternatively known as rate.

In some languages, such as in Finnish, phoneme duration has a distinctive phonological role denoting that short and long phoneme duration convey differing meanings [8]. Speech sound durations are affected by segment identity, the identities of preceding and following phonemic segments, vocal effort, the effects of lexical stress and accent, the effects of sentence, phrase and word boundaries [3].

Segment identity is, in many languages, involved in the opposition between phonologically long and short phonemes. It has also been found that the closed interval of a voiceless stop is much longer than the one in a voiced stop, and that this difference contributes to identification [9]. A clear example of an effect of the following phonemic segment is that the vowel preceding a voiceless stop consonant is longer than the one preceding a voiced stop within the same syllable. The duration of the silent interval of a stop consonant is affected by the preceding vowel segment: after a long vowel this duration is markedly shorter than after a short vowel. It has been shown that as vocal effort increases, vowel durations increase. These differences are related to the wider opening of the mouth in loud speech compared to normal speech. Lexically stressed syllables are often considerably longer than lexically unstressed syllables, although this difference itself depends much on position within word and phrase. Perception of lexical stress depends to a large extent on the pattern of syllable durations [10]. Over and above the effect of lexical stress there is a considerable effect of sentence accents or phrasal accents. Eefting in [11] found that in prepared speech accented words are roughly 20 per cent longer than unaccented ones. This difference appears to be equally distributed over lexically stressed and unstressed syllables. Segments at word boundaries tend to be somewhat longer than segments within words. This difference contributes to word boundary detection [11]. Segments in syllables immediately preceding sentence boundaries and major and minor phrase boundaries in the stream of speech are considerably longer than segments in other syllables, other things being equal [12, 13].

The general goal of duration modeling is to find a computational relation between a set of affecting factors and the segment duration. Several areas of speech technology, among which Text To Speech (TTS), Automatic Speech Recognition (ASR) and Speaker Recognition (SR) benefit from duration modeling. In TTS, the correct segmental duration contributes to the naturalness of synthetic speech [15]. In hidden Markov model (HMM)-based ASR, state duration models improve the speech recognition performance [28]. Finally, Ferrer et al. in [16] achieved significant performance improvement in the speaker recognition task, when they used duration-based speech parameters for the characterization of the speaker‟s voice.

With respect to the way duration models are built, the duration prediction approaches can be divided in two major categories: the rule-based [12] and the data-driven methods [15, 17].

The rule-based methods use manually produced rules, extracted from experimental studies on large set of utterances, or based on previous knowledge. The extraction of these rules requires labour of expert phoneticians. In the most prominent attempt in the rule-based duration modelling category, proposed by Klatt in [12], rules which were derived by analyzing a phonetically balanced set of sentences, were used in order to predict segmental duration. These rules were based on linguistic information such as positional and prosodic factors. Initially a set of intrinsic (starting) values was assigned on each phone which was modified each time according to the extracted rules. Data-driven methods for the task of phone duration modelling were developed after the construction of large databases. Data-driven approaches overcame the problem of the extraction of manual rules by employing either statistical methods or artificial neural network (ANN) based techniques which automatically produce phonetic rules and construct duration models from large speech corpora. Several machine learning methods have been used in the phone duration modelling task. The linear regression (LR) [18] models are based on the assumption that among the features which affect the segmental duration there is linear independency. These models achieve reliable predictions even with small amount of training data but do not model the dependency among the features. On the other hand, decision tree models and in particular classification and regression tree (CART) models [19], which are based on binary splitting of the feature space, can represent the dependencies among the features. Another technique which has been used on the phone duration modelling task is the sums-of-products (SOP), where the segment duration prediction is based on a sum of factors and their product terms that affect the duration [13]. Bayesian networks models have also been introduced on the phone duration prediction task. These models incorporate a straightforward representation of the problem domain information and despite their time consuming training phase, they can make accurate predictions even when unknown values come across in some features [20]. Furthermore, instance-based algorithms [17] have been used in phone duration modelling. In instance-based approaches the training data are stored and a distance function is employed during the prediction phase in order to determine which member of the training set is closer to the test instance and predict the phone duration.

Our interest in this study is modeling duration for TTS systems. The goal of TTS is to convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person. Basically, there are three types of speech synthesis: Articulatory Synthesis, Formant Synthesis and Concatenative Synthesis. These approaches are used to process the textual input into a sound output. All synthesis methods have some benefits and problems of their own and it is quite difficult to say which method is the best one. The TTS synthesis procedure consists of two main phases. The first one is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the acoustic output is produced from this phonetic and prosodic information. These two phases are usually called as high- and low-level synthesis. A simplified version of the procedure is presented in Figure 1.1. Input text, optionally enriched by tags that control prosody or other characteristics, enters the high-level synthesis where a text analysis module detects the document structure (in terms of, e.g., lists versus running text, paragraph breaks, sentence breaks, etc.), followed by text normalization (expansion to literal word tokens, encompassing transcription of acronyms, abbreviations, currency, dates, times, etc.). The tagged text then enters a phonetic analysis module that performs homograph disambiguation, and grapheme-to-phoneme conversion. The latter process is also called „„letter-to-sound“ conversion. The string of tagged phones enters a prosodic analysis module that determines pitch, duration (and amplitude) targets for each phone. Finally, the string of symbols that was derived from a given input sentence is passed on to the low-level synthesis where it controls the voice rendering that corresponds to the input text.

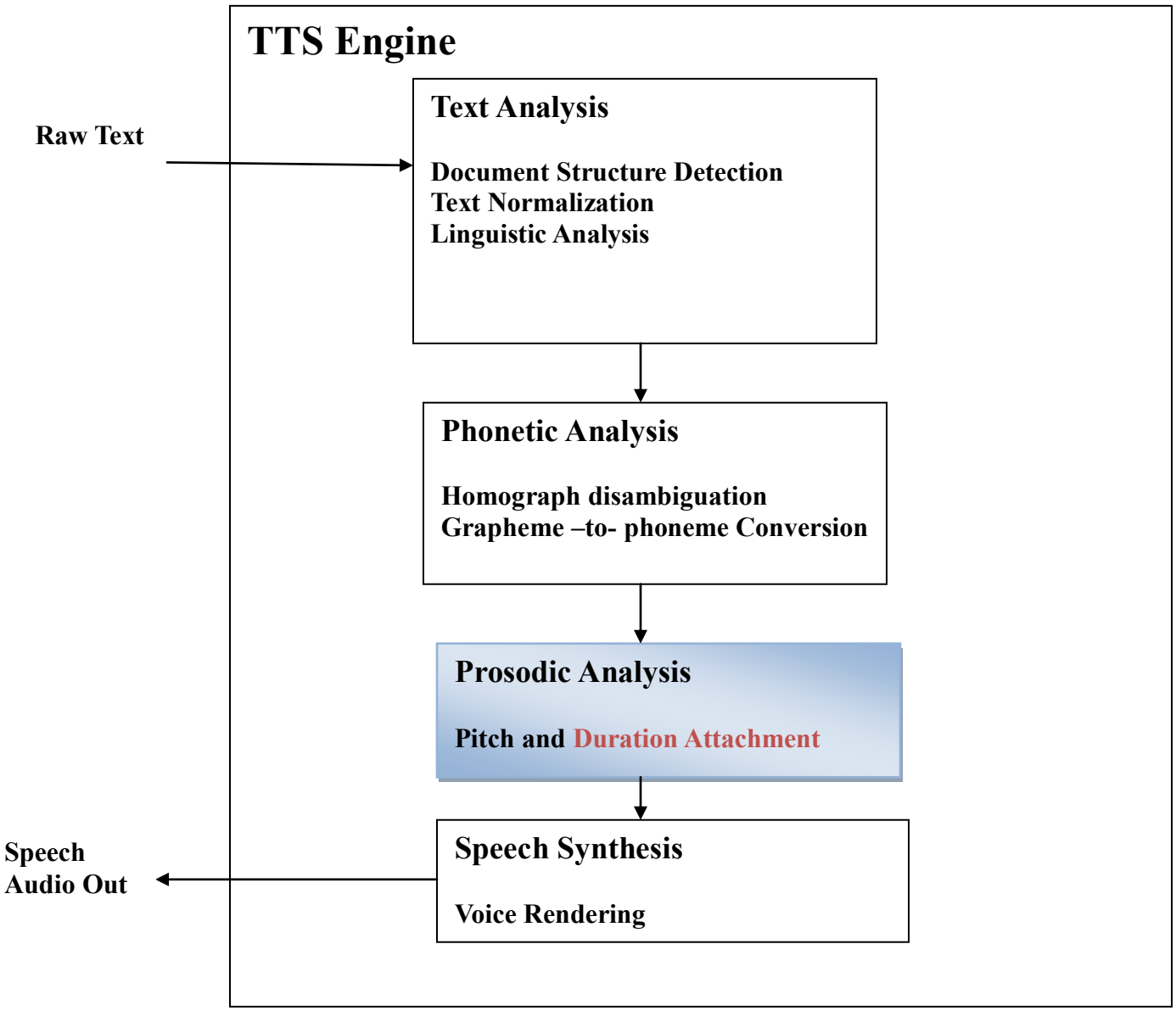

Figure 1.1 Block diagram of general Text-to-Speech System.

In TTS there are two major issues concerning the quality of the synthetic speech, namely the intelligibility and the naturalness [5]. The intelligibility of speech is the degree to which that speech can be understood. A highly intelligible utterance is an utterance whose words can be correctly identified. The higher the intelligibility of an utterance the more accurately the phonemes in that utterance are identified. Naturalness might be considered to be the degree to which a speech synthesis system is able to model the target natural speech model. It can also represent how close to the human natural speech, the synthetic speech is perceived. One of the

most important factors for achieving intelligibility and naturalness in synthetic speech is the accurate modeling of prosody. The task of prosodic modules in TTS synthesizers is that of computing a set of prosodic parameters starting from the linguistic information contained in the text that has to be synthesized. Since duration is one of the most important prosodic features that contribute to the perceived naturalness of synthetic speech, accurate modeling of phones‟ duration is essential.

1.2 Statement of the Problem

So far, different researchers have studied and implemented TTS systems in different techniques for Amharic language [22, 23, 24]. Perceptual results found from these studies reveal more works have to be done to improve the quality of speech produced by Amharic TTS systems. Moreover, some of them reported naturalness of the synthetic speech could be improved through prosody modeling, particulary duration modeling. In [22] they discovered that synthesized speech has lack of quality due to the proper assignment of epenthetic vowel duration. Also in [24] apart from mentioning there is a problem of quality of the speech produced by the Amharic TTS system they built, they recommended about consideration of prosodic effect to improve the system. Eventhough, primilinary anlysis and modeling of duration has been attempted in [43], it has been done to basic Amahric units with small data and limited number of features. Therefore this research tries to address duration modeling issues on the top of what has been done by these researchers to improve the quality of Amharic TTS. In order to increase the quality of the synthesized speech we will build duration model and integrate it into Amharic TTS that will be built in this thesis.

1.3 Motivation

The researcher will conduct this study to improve the quality of speech produced by Amharic TTS and make beneficial those who use this system dedicatedly like visually impaired peoples.

Objectives

- The primary objective of this research is to build duration model for Amharic and integrate with TTS system to improve naturalness and intelligibility of artificially generated waveforms.

Specific Objectives

- To study the characteristics of Amharic writing system
- To identify linguistic features that has correlation with duration pattern
- To select appropriate features for use in duration modeling
- To select feature extraction techniques
- To select appropriate duration modeling technique (i.e CART)
- To select appropriate TTS model (i.e Concatinative Unit Selection)
- To build the TTS model and develop a prototype
- To integrate the duration model with the Amharic TTS
- To measure performance improvement of the TTS system

Scope of the Study

The scope is limited to duration modeling of Amharic language for use in TTS systems. In this thesis we will cope with the task of duration modeling just to be used in TTS systems not in ASR or ASR.

1.4 Methods and Tools

Data Collection

In order to implement duration prediction model a parallel corpus (speech and text) with labeled speech files, the source of this data is from the data used in [25]. Our corpus consists of 899 sentences composed of 7, 232 words 33,500 phonemes. The corpus will be divided into training data (90% of the total segments) and test data (10% of the total segments).

Tools and Techniques

The following tools will be used to accomplish the overall tasks:

- Microphone for recording
- Speaker for playback record /synthesized speech wavefile
- WaveSurfer for displaying and labeling the speech waveform.
- Active Perl for developing scripts for text processing
- g++ for compiling C++ code that will be written for grapheme to phoneme convertor and testing prototype
- Wagon for generating Classification and Regression trees from training data. It will be also used for objective test.
- Festival (Festvox) for TTS prototype development
- Edinburgh Speech tools to be used by Festival for speech data processing

Modeling Methodologies

We will use data driven Classification and Regression Trees (CART) based duration modeling and Concatenative based TTS model.

Evaluation

In order to test the performance of our duration model we will use two metrics: correlation coefficient (r) between observed and predicted values of duration and Root Mean Squared Error (RMSE) in milliseconds (ms) as objective evaluation. Subjective evaluation will be made to measure performance improvement of the TTS using Mean Opinion Score (MOS).

Correlation coefficient is defined in [26].

$$r = \frac{\sum_{m=1}^{M}(d_m^{obs} - \bar{d}^{obs})(d_m^{pred} - \bar{d}^{pred})}{\sqrt{\{\sum_{m=1}^{M}(d_m^{obs} - \bar{d}^{obs})^2\}\{\sum_{i=m}^{M}(d_i^{pred} - \bar{d}^{pred})^2\}}} \quad (1)$$

where M is the size of the test set, d_m^{obs} is an observed duration of a phone in the m-th feature vector, $\bar{d}^{obs}$ is the mean observed duration across the test set; d_m^{pred} is a predicted duration of a phone in the m-th feature vector, $\bar{d}^{pred}$ is the mean predicted duration across the test set.

For the RMSE, the formula is taken from [27]

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(d_m^{obs} - d_m^{pred})^2} \quad (2)$$

Where M is the size of the test set, d_m^{obs}. is the observed duration, and d_m^{pred} is the predicted duration in the m-th feature vector, respectively.

The smaller the RMSE (Root Mean Squared Error) and the larger the correlation the better the performance will. Having a positive correlation coefficient mean there is a positive linear relationship between actual and predicted duration.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

As it is mentioned in chapter one duration modeling is important in both automatic speech recognition and text-to-speech. In ASR, state duration models are usually constructed to assist in the Hidden Markov Model (HMM)-based speech recognition. In TTS, synthesis of proper duration information is essential for generating a highly natural synthetic speech. The most important parameter in the quality of the speech systems is intelligibility. The naturalness of the utterance produced by the TTS system has a direct effect on the usefulness of the TTS system. Experiments based on acoustic analysis have shown that the naturalness and intelligibility of the utterance produced by TTS systems greatly depends upon the prosodic features duration and intonation. Having correct parameters for duration and intonation, the naturalness of the speech produced can be greatly improved [29].

Currently, intelligibility of the best TTS systems is extremely good, and certainly good enough for many real applications. However, it rarely takes a listener more than 500 millisecond (ms) to decide that speech generated by TTS is not recorded natural speech, let alone speech generated by an actively communicating human. In speech synthesis the accurate modeling of prosody becomes mandatory for producing synthetic speech of high quality. The main aspects of prosody are the phone duration, fundamental frequency and loudness of speech. The phone duration controls the rhythm and the tempo of the speech [6]. Consequently, flattening the prosody in synthetic speech would lead to a monotonous and without rhythm speech, degrading the quality and specially the naturalness of speech.

Speech exhibits intricate temporal patterns, which must be mimicked by TTS systems in order to sound natural. More precisely TTS systems should be able to produce segment and pause durations that do not significantly differ from natural ones. In a text-to-speech system, the duration of each phone may be predicted by a duration model. This model is usually trained

using a database of phones with known durations; each phone (and the context it appears in) is characterized by a feature vector that is composed of a set of linguistic factor values [30].

Many researches have been conducted over the years in the field of duration modeling which utilize different techniques and interesting results are obtained for various languages [12, 31]. When developing a new model for duration prediction in a TTS system, its performance is usually measured by comparing the predicted durations and the observed "original" durations in a database. The performance of the new model is then expressed in terms of error rates such as root mean square error (RMSE) and the correlation coefficient (CC).

2.1 Phonetic conditioning factors affecting vowel and consonants duration

Klatt and Cooper in [32] refer to the intrinsic or inherent phonological duration of phonetic segments in English. A term used to refer to a property of sound which is thought to be crucial to the identity of a contrast. A segment of a particular type must have the property before it can be perceived to belong to that type. For example, a tiny amount of friction follows the release of a stop consonant, but as the duration of this friction exceeds a certain minimal amount, so the segment will be perceived as an affricate; a longer duration will lead to its perception as a fricative. Such examples of intrinsic duration can also be found in vowels, where low vowels are intrinsically longer than high vowels [7].

Vowels are greatly affected in duration by a number of factors such as the identity of the preceding and following consonant, the rate of speaking, the syllable stress, and the importance or emphasis assigned to the word by the speaker [14]. It so happens that in English, the voicing of a postvocalic consonant strongly affects the duration of a preceding vowel [33]. House in [33] concluded that the shortening of vowels before voiceless consonants is due to an articulatory activity arbitrarily imposed by the phonological system of English. Fischer [34] for Danish language has reported that the duration of a vowel depends on the movement of the speech organs required in order to come from the vowel position of the following consonant. The greater the extent of the movement, the longer the vowel will be. An increase in vowel duration, when the point of articulation of the postvocalic consonants shifts farther back in the mouth, has also

been observed for Spanish. The influence of the manner of articulation of a consonant upon the duration of a preceding vowel seems to be largely dependent on the language [14].

The intrinsic duration of consonants is influenced by their place of articulation and by the manner of articulation. Most investigators agree that labials are longer than alveolars and velars, other factors kept constant [14].

2.2 Factors influencing segment duration

Speech sound durations are affected by both within syllable factors and between syllable factors. Examples of within syllable factors are segment identity, and the identities of preceding and following phonemic segments. Segment identity is, in many languages, involved in the opposition between phonologically long and short phonemes. On a more phonetic level, we observe for example that open vowels like /a/ have, other things being equal, longer durations than closed vowels like /e/ or /o/, simply because it takes more time to open the mouth further than to open it less [33]. It has also been found that the closed interval of a voiceless stop is much longer than the one in a voiced stop, and that this difference contributes to identification [35]. A clear example of an effect of the following phonemic segment is that the vowel preceding a voiceless stop consonant is longer than the one preceding a voiced stop within the same syllable. This difference contributes to the perception of stop voicedness [35].

Examples of between syllable factors are the relation between overall vocal effort and segment durations, the effects of lexical stress and accent, the effects of sentence, phrase and word boundaries, and the effect of rhythmical alternation in sequences of unstressed syllables. It has been shown that as vocal effort increases, vowel durations increase and consonant durations decrease. These differences are related to the wider opening of the mouth in loud speech compared to normal speech. It appeared impossible, however, to model these differences in a linear scaling model of the behavior of lips and jaw. This suggests that there is extensive reorganization of articulatory behavior, by which other perceptually relevant aspects of temporal patterning remain better preserved. Other things being equal, lexically stressed syllables are often considerably longer than lexically unstressed syllables, although this difference itself depends

much on position within word and phrase. Eefting in [11] found that in prepared speech accented words are roughly 20 per cent longer than unaccented ones. This difference appears to be equally distributed over lexically stressed and unstressed syllables. It cannot be tampered with without reducing the perceived acceptability of speech and speed of processing. Segments at word boundaries tend to be somewhat longer than segments within words. This difference contributes to word boundary detection [11]. Segments in syllables immediately preceding sentence boundaries and major and minor phrase boundaries in the stream of speech are considerably longer than segments in other syllables, other things being equal [13].

The durational difference between lexically stressed and unstressed syllables and their segments is also far from fixed, and depends both on the type of segments in the syllable and on syllable position in word and utterance. In word medial and utterance medial position this difference between stressed and unstressed syllables may be considerable, in the order of hundreds of milliseconds, but in prepausal position the difference often is negligible, apparently because prepausal lengthening has exhausted the "stretchability" of the syllable. It is as if each particular segment type within a particular syllabic environment can only vary its duration between a maximum and a minimum that are typical for that segment in that environment [12].

Such extreme interactions between many different factors affecting syllable and segment durations (of which many more examples are known, in [13]) have the result that the systematic effects on speech sound durations of any one particular factor can only reliably be assessed when we take the effects of many other factors into account. This has some consequences. One is that we cannot study or model the quantitative effects of rhythmical factors, presumably coinciding with the "between syllable factors" listed above, in isolation from "within syllable factors". Another is that one needs rather large databases and statistical tools to get an appropriate impression of the factors involved and their quantitative interactions. One also needs sophisticated quantitative models to account for the effects of these factors and their interactions.

An important restriction being that, due to the nature of the Text-to-Speech synthesis problem, only those features that can be automatically derived from text can be considered. In all the systems, regardless of the technique used for the modeling, it is crucial to find the parameters

that are most significant for duration modeling [30]. It has long been noticed that the durations of vowels and consonants vary according to the contexts and a great deal of studies have been performed to find these factors and their percentage effect on the phone durations.

2.3 Properties of Segmental duration data

Segmental duration is among the aspects of speech, which a text to speech system must infer from text. Two challenges are faced when trying to predict segmental duration from text. The first challenge is that the feature space has an extremely uneven frequency distribution. The feature space is the product of all factors F_1 x…..x F_n, where a factor F_i is defined as a partition of mutually exclusive and exhaustive possibilities such as {1_stressed, 2_stressed, unstressed}. The linguistic space is the subset of vectors that actually occur in the language, and is significantly smaller than the feature space. Thus as a result the training samples often include a very small and uneven subset of the feature space [29]. However it has been that text samples as small as few sentences are guaranteed to contain exceptionally rare feature vectors. The second challenge is that the factors interact. By interaction we mean that magnitude of an effect is not constant but is affected by other factors. These properties of the segmental duration data are discussed below:

2.3.1 Coverage Issues

Construction of a training database having an adequate coverage of the linguistic space is a very difficult task. This is due to the large number of factors that have major effects on segmental duration, for example for vowels at least eight factors are known to have a significant impact. Each of these factors will have several levels, this leaves us with a large feature space for studying the effect of vowel durations. In order to construct a training database, the to-be-read text must be generated to cover these feature vectors, but the lopsided of these vectors in text makes it very difficult to have complete coverage. Once having text with adequate coverage, the speech recordings should be done at a controlled speech rate. Speakers have a general tendency to talk faster in longer sentences. Multiple recordings need to be done to cater with this

variability of the speaking rate, however deviations from the average speaking rate still remain. The only way to counter these problems is to have multiple data points for each feature vector but these results in a significant increase in the amount of recording.

In practice the data used for duration modeling does not have all the feature vectors but the distribution is such that they provide durations that are accurate for some vectors, marginally accurate for a larger number of vectors, and are lacking duration information for a still larger number of vectors, these are the rare vectors of language that are not easy to capture as it drastically increases the duration data [29].

2.3.2 Interactions

Interactions among factors pose problems for models to capture the effect of any factor. This applies to the simplest sum-of-products models, i.e. the additive and multiplicative models. Factors are said to interact when other factors modulate the effect of one factor. Two types of factors exist, ordered factors and categorical factors.

Ordered factors are the factors whose effects are always in the same direction. Interactions involving ordered factors are normally well behaved in that their effects are amplified by other factors but not reversed or otherwise. Many factors are ordered for example, in spoken English it is difficult to imagine that two conditions differing only in syllabic stress and having longer vowel duration in the unstressed condition than in the stressed condition. More generally a set of factor-wise orders together define a partial order over the feature space. Having so if some feature vectors are missing in the training data, chances are that at least some of each subset is present, which can then be used to predict the duration of the missing vectors. Not all factors that apply to vowel duration are relevant for consonants [29].

Consider the stress factor for vowels and for intervocalic consonants, obviously the stress levels of the immediately surrounding vowels has a great impact for intervocalic consonants but has no significance for vowels. So the sum of products model made for vowels is not applicable in this case. In general there are categorical distinctions in the feature space that cannot be bridged by

sum of products models because either different factors apply, factors require different restrictions, factors have different effects with respect to categorical distinction. Thus the sums of products models here have to be different for both the categories. Such factors that belong to different categories are called **categorical factors**.

2.4 Types of models used for duration prediction

The general goal of duration modeling is to find a computational relation between a set of affecting factors and the segment duration. Duration modeling approaches can be categorized into two approaches: rule-based and data-driven. The former is a conventional one which uses linguistic expertise to manually infer some phonologic rules of duration generation based on observations on a large set of utterances. A prevalent method of the approach for TTS uses sequential rules to initially assign the duration of a segment with an intrinsic value, and then successively applies rules to modify it [36]. The data-driven approach tries to construct a duration model from a large speech corpus, usually with the aid of statistical methods. It first designs a computational model to describe the relationship between the segment duration and some affecting factors, and then trains the model on the speech corpus. The training goal is to automatically deduct phonologic rules from the speech corpus and implicitly memorize them in the model‟s parameters [11].

In this section, some commonly used models in the TTS systems will be reviewed. In historical order, they are Lookup table model, Additive and Multiplicative Models, Klatt‟s Model, Classification and Regression Tree Model and finally Sum-of-Products Models.

2.4.1 Lookup Table

Lookup table is the simplest statistical model used for duration prediction. It consists of assigning a value to each possible combination of parameters of the descriptor vector. These values are found using the training data and finding the base average duration for each feature vector. This approach however cannot be generalized and requires that the training database

should cover the feature space completely in order to find the average durations for each feature vector [37].

2.4.2 Additive and multiplicative models

In statistics, an additive model is a nonparametric regression in which the combined effect of the explanatory variables (and their interaction) is equal to the sum of their separate effects. This process can be represented as:

$$D_i \quad : \quad F_i \longrightarrow R \tag{3}$$

Each D_i is a parameter vector, whose number of components is equal to the number of levels of factor F_i. For example if F_1 is the Stress factor, and D_1 (1 stressed) = 15ms, D_1 (2 stressed) = 6 ms, and D_1 (unstressed) = -20ms, then in this case the effect is that 1 stressed vowels are 35ms longer than unstressed vowels. These parameter values can be estimated from data and represent the effect of Stress factor in this particular case.

Duration prediction with additive model is done according to the formula [37]

$$DUR\,(f) = D_1\,(f_1) + D_2\,(f_2) + \ldots + D_N\,(f_N) \tag{4}$$

for a feature vector $\vec{f} = (f_1, \ldots, f_N)$. Here fi represents a value on the i-th factor. For example, if the i-th factor is word-position, then fi can be „word-initial" or „word-final". The effect of factor i on the duration is given by the parameter $A_i(f_i)$ when it has level f_i. To give an example, if feature vector is $\vec{f} = (f_1, f_2, f_3)$ corresponding to the word position factor levels, A_1(word − initial),A_2(word − middle),A_3(word − final) represent the effects of the word-position factor. When the effects of one factor are changed by another factor, these two factors are called to interact in the additive sense. For example for i = 1, 2, 3, in addition to F_i we have $E_i : F_i \rightarrow \boldsymbol{R}$ and the combination rule is given by:

$$DUR\,(f) = D_1\,(f_1) + D_2\,(f_2) + D_1\,(f_1) \times D_3\,(f_3) \tag{5}$$

It is evident that determining the magnitudes is no longer straight forward as now the factors interact. If the third factor is phone identity then the effects of stress are given by:

.

$$\text{StressEffect}(/e/) = [\, D_1\,(1\ \text{stressed}) + E_1\,(1\ \text{stressed}) \times D_3\,(/e/)\,] - [D_1\,(1\ \text{unstressed}) + E_1\,(1\ \text{unstressed}) \times D_3(/e/)\,] \quad (6)$$

$$\text{StressEffect}(/i/) = [D_1\,(1\ \text{stressed}) + E_1\,(1\ \text{stressed}) \times D_3\,(/i/)] - [D_1\,(1\ \text{unstressed}) + E_1\,(1\ \text{unstressed}) \times D_3(/i/)] \quad (7)$$

The stress effects will be different for these two vowels, meaning that the stress factor and vowel identity factor interact. For interactions in the multiplicative sense the + is replaced by × and. In multiplicative interactions, the effects are measured in fractions rather than millisecond amounts [37].

2.4.3 Klatt 's Model

The most prevalent rule-based duration model is a sequential rule based system proposed by Klatt [38] which was for English. His model was based on information presented in phonetic literature about the different factors affecting segmental duration. Rules which were derived by analyzing a phonetically balanced set of sentences were used in order to predict segmental duration. These rules were based on linguistic information such as positional and prosodic factors. Initially a set of intrinsic (starting) values was assigned on each phone which was modified each time according to the extracted rules. Based on a large number of experiments modifications from an inherent duration of all the phones are describe by a set of 11 rules. This model assumes that;

- Each phonetic segment type has an inherent duration that is specified as one of its distinctive properties
- Each rule tries to affect a percentage increase or decrease in the duration of the segment.
- Segments cannot be compressed shorter than a certain minimum duration. The model is represented by the following formula:

$$\mathrm{DUR} = \mathrm{MINDUR} + \frac{(\mathrm{INHDUR} - \mathrm{MINDUR}) \times \mathrm{PRCNT}}{100} \tag{8}$$

Where

INHDUR = inherent duration of the segment

MINDUR = minimum duration of the segment if stressed

PRCNT = percentage shortening determined by applying rules determined from experiments (i.e. phrase final lengthening, polysyllabic shortening).

The rules presented by the Klatt model are applied successively starting with initial (or inherent) segment duration. The rules proposed by Klatt are:

1. Pause insertion rule: Insert a brief pause before each sentence internal main clause and at other boundaries delimited by an orthographic comma
2. Clause-final lengthening: The vowel or syllabic consonant in the syllable just before a pause is lengthened. Any consonants in the rhyme (between this vowel and the pause) are also lengthened.
3. Phrase-final lengthening: Syllabic segments (vowels and syllabic consonants) are lengthened if in a phrase-final syllable. Durational increases at the noun-phrase/verb-phrase boundary are more likely in complex noun phrase or when subject-verb-object order is violated; durational changes are much less likely for pronouns. The lengthening is perceptually important
4. Non-word-final shortening: Syllabic segments are shortened slightly if not in a word-final syllable.
5. Polysyllabic shortening: Syllabic segments in a polysyllabic word are shortened slightly.
6. Non-initial-consonant shortening: Consonants in non word-initial position are shortened.
7. Unstressed shortening: Unstressed segments are shorter and more compressible than stressed segments.
8. Lengthening for emphasis: An emphasized vowel is significantly lengthened.
9. Postvocalic context of vowels: The influence of a postvocalic consonant (in the same word) on the duration of a vowel is such as to shorten the vowel if the consonant is voiceless. The effects are greatest at phrase and clause boundaries.

10. Shortening in clusters: Segments are shortened in consonant sequences (disregarding word boundaries, but not across phrase boundaries).
11. Lengthening due to plosive aspiration: A stressed vowel or sonorant receded by a voiceless plosive is lengthened.

2.4.4 Classification and Regression Trees (CART)

Classification and Regression Trees is a classification method which uses historical data to construct so-called decision trees. Decision trees are then used to classify new data. In order to use CART we need to know number of classes a prior [39].

Classification and regression are two important problems in statistics. Each deals with the prediction of a response variable y given the values of a vector of predictor variables x. Let X denote the domain of x and Y the domain of y. If y is a continuous or discrete variable taking real values, the problem is called regression. Otherwise, if Y is a finite set of unordered values, the problem is called classification. In mathematical terms, the problem is to find a function d(x) that maps each point in X to a point in Y. The construction of d(x) requires the existence of a training sample of n observations $L = \{(x_1, y_1). . . (x_n, y_n)\}$. In computer science, the subject is known as supervised learning. The criterion for choosing d(x) is usually mean squared prediction error $E\{d(x)-E(y|x)\}^2$ for regression, where E(y|x) is the expected value of y at x, and expected misclassification cost for classification [39].

If Y contains J distinct values, the classification solution (or classifier), may be written as a partition of X into J disjoint pieces $A_j = \{x: d(x) = j\}$ such that $\mathcal{X} = \cup_{j=1}^{J} A_j$. A classification tree is a special form of classifier where each A_j is itself a union of sets, with the sets being obtained by recursively partitioning the x-space. This permits the classifier to be represented as a decision tree. A regression tree is similarly a tree-structured solution in which a constant or a relatively simple regression model is fitted to the data in each partition.

A classification or regression tree algorithm has three major tasks:

(i) how to partition the data at each step,

(ii) (ii) when to stop partitioning, and

(iii) (iii) how to predict the value of y for each x in a partition?

There are many approaches to the first task. For ease of interpretation, a large majority of algorithms employ univariate splits of the form $x_i \leq c$ (if x_i is non-categorical) or $x_i \in B$ (if x_i is categorical). The variable x_i and the split point c or the split set B are often found by an exhaustive search that optimizes a node impurity criterion such as entropy (for classification) or sum of squared residuals (for regression). There are also several ways to deal with the second task, such as stopping rules and tree pruning. The third task is the simplest: the predicted y value at a leaf node is the class that minimizes the estimated misclassification cost (for classification), or the fitted value from a model estimated at the node (for regression) [39].

In CART based modeling the feature space is divided to minimize prediction error and constructs a tree representing the partition of the feature space [40]. In CART, in the training phase, a tree is formed by successively dichotomizing the factors (e.g., the stress factor is split into 1-stressed, 2-stressed vs. unstressed) to minimize the variance of the durations under the two newly formed subsets of the speech corpus. For each node of the tree, the observed average duration of the associated subset of the speech corpus is listed. In other words, CART is a general purpose statistical method that imposes little structure on the data.

2.4.5 Sum of Products Model (SoP)

Sums-of-Products models [13] are general linear models. The general linear model can be seen as an extension of linear multiple regression for a single dependent variable. The general purpose of multiple regressions is to quantify the relationship between several independent or predictor variables and a dependent or criterion variable.

The sums-of-products model derives from analysis of variance (ANOVA). The ANOVA customarily is used for hypotheses testing, in particular testing for the existence of main effects and (additive) interactions. In a **two-way ANOVA**, the effects of two factors or treatments can

be investigated simultaneously. Two-way ANOVA also permits the investigation of the effects of either factor alone and of the *two factors together*. The effect on the population mean that can be attributed to the levels of either factor *alone* is called a **main effect**. An **interaction effect** between two factors occurs if the total effect at some pair of levels of the two factors or treatments differs significantly from the simple addition of the two main effects. Factors that do not interact are called *additive*. Three questions answerable by two-way ANOVA:

- Are there any *factor A main effects*?
- Are there any *factor B main effects?*
- Are there any *interaction effects between factors A and B?*

The sum of products model represents the duration for phonemes/context combination described by the feature vector $\vec{f}$ as [41]:

$$\mathrm{DUR}\left(\vec{f}\right) = \sum_{i \in K} \prod_{j \in I_i} S_{i,j}\left(f_j\right) \tag{7}$$

Here, K is a set of indices, each corresponding to a product term;
I_i is the set of indices of factors occurring in the i-th product term.
S_{ij}, are "factor scales" which correspond to the weight on the contribution of the value of the j-th linguistic factor x_j . $S_{ij}(\cdot)$ is a "factor scale" and is a function of the value of its argument (its argument being a linguistic factor). When the linguistic factor is discrete (which is the case throughout this article), then this function is implemented as a table, which has one entry for each possible value of the corresponding linguistic factor. The table entries are learnt from data.

The sum of products models capture the phenomenon of directional invariance, according to which the effects of a factor, like stress or prepausal position have always effects on the same direction. A combination of sums and products more beautifully captures and reflects the properties of duration, as directional invariance and interactions. Several sum-of-products models have been proposed which model either the duration or logarithm of the duration, and all these models give near about the same results. For vowels the best results are obtained by the following sum-of-products model [41].

$$D(v,a,c,p,t) = S_{1,1}(v) + S_{2,1}(v,a) + S_{3,1}(v) *S_{3,2}(p) *S_{3,3}(c)* S_{3,4}(t) \quad (10)$$

Here (v) denotes vowel identity, (a) stress, (p) sentence position, (c) class of post vocalic phone and (t) the manner of articulation of post vocalic phone. For a given number of factors there are many different sum-of-product models. For example for two factors there are five possibilities: S1,1* S1,2 , S1,1 *S2,2 , S1,1 + S2,1* S2,2 , S1,2 + S2,1 *S2,2 , S1,1 + S2,1 *S2,2 +S3,2 . The number of possible models increases rapidly with the number of factors: it is roughly proportional to 2^{2n-1} -1, where n is the number of factors. There is a direct mathematical link between the ordinal properties of the data set and the sum-of products model: In the data set when the factors are ordered they exhibit regular patterns of joint independence and amplificatory interactions. The sum-of-products model captures these properties of the segmental duration data. The key assumption made here is that the ordinal structure discovered in the training database can be found in the language in general (restricted to the same speaker and speaking mode). These properties exhibited are the resultant of the stable properties of the speech production apparatus [26].

2.5 An Overview of Speech Synthesis Techniques

As stated in the first section of chapter one, there are three different categories of waveform generation or so called types of speech synthesis: articulatory synthesis, formant synthesis, and concatenative synthesis.

Articulatory synthesis tries to model the human articulators as perfectly as possible, using parameters that model the mechanical motions of the articulators and the distributions of volume, velocity and sound pressure in the lungs, larynx, and vocal and nasal tracts, so as to be potentially satisfying method to produce high-quality synthetic speech. On the other hand, it is also one of the most difficult methods to implement and the computational load is also considerably higher than the other methods. Thus, it has received less attention than the other synthesis methods and has not yet achieved the same level of success as the other methods have achieved [1, 43].

Formant synthesis employs some set of rules to synthesize speech using the formants that are the resonance frequencies of the vocal tract. Since the formants constitute the main frequencies that make sounds distinct, speech is synthesized using these estimated frequencies [43]. Formant synthesis does not require any pre-recorded speech samples. Instead, the synthetic speech is created using an acoustic model. Parameters such as pitch, voicing, and noise levels are varied over time to synthesize speech waveforms.

Concatenative based speech synthesis technique uses various length of pre-recorded voices derived from natural speech. By connecting pre-recorded natural utterances, the intelligible and natural sounding synthetic speech will be produced. One of the most important activities in concatenative synthesis is to find appropriate unit length of the utterance to be stored in the database. With longer units, it provides high naturalness, less concatenation points and good control of co-articulation. However, the amount of units and memory required significantly increases as unit length increases. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex. In present systems units are usually words, syllables, phonemes, and diphones [1].
In general, concatenative TTS can be further classified into two sub-categories, diphone synthesis and unit selection based synthesis.

The diphone synthesis approach makes use of only one representative acoustic unit for each diphone; and the pitch, duration and amplitude of the diphones are modified according to some prosody prediction model. Nevertheless, prosody prediction is not error-free and signal processing methods for carrying out the modification introduce speech distortion.

Unit selection based synthesis uses a larger speech corpus selecting the best matching units based on a pre-defined distance measure. During synthesis, an algorithm selects one unit from the possible choices in an attempt to find the best overall sequence of units which matches the input specification [4]. It has been stated in [45, 49] that unit selection synthesis provides more natural sounding utterance than the one that uses only one instance of each unit in a database. It is also pointed out in the works that the naturalness is due to relying less on signal smoothing techniques that cause degradation in the quality of synthetic voice. That is, unit selection

synthesis employs almost no signal modification techniques that have impact on the degradation of the quality of synthetic voice [4].

In unit selection method, the selection of the best unit sequence from the database is typically treated as a search problem in which the best sequence of candidates from the inventory is the one that has the lowest overall cost [45]. This cost is often decomposed into two costs: a target cost (how closely candidate units in the inventory match the specification of the target phone sequence) and join cost (how well neighboring units can be joined) [45]. The target and concatenation costs are based on measures of phonetic features (identity of a speech unit, position of a speech unit, previous and next units, etc.) and prosodic features (F0, Mel frequency cepstrum coefficients, power, etc.) [53].

2.6 Duration Component in TTS

TTS system is composed of several modules and duration component is one of those. Thinking TTS duration component as a black box, inputs to this black box can be described as discrete vectors. To give an example, input vector to the TTS duration component can be a vector like the one below,

$\vec{f}$ =< /o/, stressed, accented,. . . , word-final >

Which is a typical vector representing the properties of phoneme /o/ with properties stressed, accented and in word-final position. Here, an element of such a vector is a level on a factor. A factor can be represented by a set. An example for word position factor is given below.

Word position = {word-initial, word-middle, word-final}

The set of all vectors forms the factorial space

$$S = F_1 \times F_2 \times \ldots \times F_N \tag{11}$$

where (1, . . . ,N) represents the factors. How much of this factorial space should be covered in the training data base depends on the model used.

$$\mathrm{DUR} : S \rightarrow \mathbb{R} \tag{12}$$

Task of duration component is to give a duration value for each input vector. Duration component maps discrete vectors onto the real numbers, R, as shown in the above equation [37].

CHAPTER THREE

RELATED WORKS

3.1 Introduction

In the past, durational characteristics of speech in various languages have been the subject of many recent researches. Many factors have been shown to have effects on segmental duration. Regarding approaches to develop duration model, as stated in chapter two we have two alternative techniques; the one is rule based and the other is data-driven. Comparatively rule-based approach requires phonetician and long term devotion to extract rules from speech data, Data-driven approaches overcame the problem of the extraction of manual rules by employing statistical methods which automatically produce phonetic rules and construct duration models from large speech corpora.

As the state of art is dominated by data-driven approach, we have focused on explaining duration models built for different languages using corpus based technique.

3.2 Duration Modeling of Indian Language Hindi and Telugu

In duration modeling for two Indian languages Hindi and Telugu, CART based data-driven is presented in [31]. For Telugu, the corpus is of around 14 minutes and consists of 156 read sentences. The corpus is segmented at phoneme level, thus yielding a total of 6846 segments. The corpus is divided into training data (5477 segments, 80% of the total segments) and test data (1369 segments, 20% of the total segments). For Hindi, the corpus used for study includes duration around 12 minute and consists of 121 read sentences. The corpus is segmented at phoneme level, thus yielding a total of 5014 segments. The corpus is divided into train data (4083 segments, 80% of the total segments) and test data (1021 segments, 20% of the total segments). Each segment in both the corpora (Telugu and Hindi) is annotated with the following features together with the actual segment (phoneme) duration:

- Identity of the current phoneme
- Identity of the preceding phoneme
- Identity of the following phoneme
- Position in the parent syllable
- Parent syllable initial; Returns 1 if the segment is the first segment in the syllable it is related to, otherwise 0.
- Parent syllable final; Returns 1 if the segment is the last segment in the syllable it is related to, otherwise 0.
- Parent syllable position type; the type of syllable position in the word it is related to. This may be any of: „single" for single syllable words, „initial" for word initial syllables in a poly-syllabic word, „final" for word final syllables in poly-syllabic words, and „mid" for syllables within poly-syllabic words.
- Number of syllables in the parent word
- Position of parent syllable in the word; the position of the syllable in the word it is related to. The index counts from 0.
- Parent syllables break information; Break level after the parent syllable. This feature is categorical and it has 4 possible values: 0 for word internal syllables,1 for syllables occurring in word boundary, 3 for syllables occurring in phrase boundary, 4 for syllables occurring in sentence boundary.
- Phrase length (in number of words).
- Position of phrase in the utterance.

For Telugu correlation obtained between actual and predicted durations is 0.8014 and the root mean squared error (RMSE) of prediction is 22.86 ms. For Hindi correlation obtained between actual and predicted durations is 0.7526 and the root mean squared error (RMSE) of prediction is 27.14 ms.

3.3 A Duration Model for Czech Text-to-Speech Synthesis

For Czech TTS, The database used for duration model consists of 56 sentences and 5081 phones [40]. The data was divided at random into training set (70% (3,563 segments)) and test set (30% (1,518 segments)). They used a phoneme system with 10 vowels and 25 consonants.

They used the following features and found correlation coefficient of the model was 0.79 and RMSE 20.3 ms.

- Identity of the current phoneme
- Identity of the preceding phoneme
- Identity of the following phoneme
- Word length
- Parent syllable position type
- Position of parent syllable in the word from the end
- Syllable position in the phrase from the beginning
- Syllable position in the phrase from the end
- Phoneme position in the syllable

3.4 Segmental Duration Modeling in Turkish

For Turkish duration model, the speech database has been split into two subsets [42]: training dataset is used to develop duration models and test dataset is used to evaluate the performance of the model on unseen data. The test set consists approximately 20% of the database and the remaining phonemes constitute the training set. The total number of instances in the training and test sets is 29527 and 7318, respectively.

They used the following features and the resulting CART tree predicts phoneme durations with a CC of 0.7815 and a RMSE of 20.0115ms.

- Identity of the current phoneme

- Identity of the preceding phoneme
- Identity of the following phoneme
- Lexical Stress: There exist two levels for lexical stress: Accented (A) or Not-Accented (NA). A segment is associated with an A if the vowel of the parent syllable is stressed and an NA otherwise.
- Position in Syllable
- Syllable Type: Two levels are used to denote parent syllable types: Heavy (H) and Light (L).
- Part-of-Speech
- Phrase Break Information: Speech corpus has been evaluated perceptually several times and major perceptual breaks in the utterances are marked manually. The marks mainly correspond to the speaker"s breathings. The feature is represented by three levels: Segment takes a Phrase Initial (PI) value if it immediately follows a phrase break, a Phrase medial (PM) value if there is no phrase break engagement and a Phrase Final (PF) if a phrase break immediately follows the segment.
- Syllable Position in Word: Syllables of the same word are counted from the left starting from 1. The database contains words of at most 10 syllables; however, there is no word that contains 9 syllables.
- Word/Syllable Position in Sentence: Words/Syllables are counted from left starting from 1. The longest sentence contains 19 words and 45 syllables. All phonemes of the parent word take the same value.
- Word Length: Each phoneme of the same word is annotated with the total number of syllables in that word. The attribute values are numeric and ranges from 1 to 10.
- Number of Syllables Word
- Number of Words in Sentence
- Number of Words from (to) the Preceding (Following) Phrase Break: The attributes identify the number of words between the parent word and the preceding (following) phrase break counting from 0.
- Number of Syllables from (to) the Preceding (Following) Phrase Break: This attribute is almost the same as the number of words from the preceding phrase break attribute counting from 0.

3.5 Decision Tree Based Segmental Duration Prediction Amharic for Amharic TTS System

For Amharic duration modeling done in [43] recorded speech consists of about 183 sentences. Then the recorded data was manually segmented at phoneme level and yielded 100 segments for this particular study.

The following are the list of features which are extracted for each segment in the corpus together with the actual segment duration:

- Phoneme Identity; (vowel, consonant)
- Preceding phoneme type; (voiceless fricative, voiceless stop, voiced stop, voiced fricatives, glides, ejectives, nasals)
- Next phoneme type ; (voiceless fricative, voiceless stop, voiced stop, voiced fricatives, glides, ejectives, nasals)
- Position in the parent syllable
- Position in a word
- Word length
- Singleton/Geminate decision(Yes/No)

The authors conducted objective evaluation and they reported that they found promising results.

3.6 Possible Feature Set Affecting Duration in Amharic

One of underlying principles of feature selection states that every phone has its intrinsic duration, while at the same time the phone is affected by its neighboring phones. Intrinsic durations and phone interaction have been studied for many languages. Such universal linguistic phenomena are also observed in Amharic. Together with factors identified in [43] and based on duration models used in the reviewed languages the following features are expected to be found in Amharic:

- Identity of the current phoneme

- Place of Articulation
- Identity of the preceding phoneme
- Identity of the following phoneme
- Position in syllable
- Syllable position in the word
- Number of syllables in the word
- Syllables break information
- Syllable Type
- Phrase length
- Position of phrase in the sentence
- Number of phonemes before and after the vowel in the specific syllable
- Parent syllables break information
- Lexica stress
- Number of Syllables between accented syllables
- Number of non-major phrase breaks
- Function Content distinction
- Part-of-Speech
- Singleton/Geminate decision

3.7 Selected Approach for Amharic Duration Model

Based on the literatures we reviewed, Classification and Regression Trees has been selected to be used as predictive model for Amharic duration modeling. Moreover the main reasons to choose this approach are:

- CARTs are that standard tools for their generation are widely available
- The tree generated by CART is more interpretable (in contrast to neural networks)
- CART is useful for the case of less researched language like Amahric, for which the most relevant features that affect segment duration and the way they are interrelated have not been studied in detail.

In chapter Five we will presented the details of how CART works and decision tree building procedures using standard tool (wagon) that implements CART algorithm to construct predictive model will be explained.

CHAPTER FOUR

THE AMHARIC LANGUAGE WRITING SYSTEM AND PHONEMESET

4.1 Introduction

Amharic is a Semitic language and the official language of the government of Ethiopia. It is the second most spoken Semitic language in the world, next to Arabic and is estimated to be spoken by well over 20 million people as a first or second language [44]. It is today probably the second largest language in Ethiopia (after Oromo, a Cushitic language) and possibly one of the five largest languages on the African continent. Amharic uses a unique script which has originated from ancient language the Ge"ez alphabet, which is the liturgical language of the Ethiopian Orthodox Church. Written Ge"ez can be traced back to at least the 4th century A.D. The first versions of the language included consonants only, while the characters in later versions represent consonant-vowel (CV) phoneme pairs [45].

Despite Amharic language has the large number of speakers, only very few computational linguistic resources are available. This has a direct impact specially to implement research works that are done so far and to be done also in the future. Speech synthesis for Amharic language is one of those areas that are not well explored. Even if this paper targets on duration modeling of phonemes for speech synthesis, it will discuss related issues which are important for the research work. We look at the Amharic writing system and phonemeset in the coming sections.

4.2 Amharic Writing System

The way Amharic orthographic characters are written is very similar to the way they are spoken. It means Amharic is a phonetic language. The mapping of the written form and the spoken form is one to one except the epenthetic vowel (i.e /ix/ which has important role during syllabification of the word in the language and allows splitting impermissible consonant clusters.) [22]. Amharic script (orthographic representation) consists of 33 core characters (shown in Table 3.1) each of which occurs in a basic form and six other forms. These seven forms of a character are

known as orders and represent syllable combinations consisting of a consonant and following vowel. This representation of each character in 7 different forms makes the number of core characters to be 231 (33 * 7), also called syllographs (ፊደል) [11, 46].

Table 4.1 Amharic alphabets with their seven orders.

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

4.3 Amharic Phonemeset

Based on which articulatory organs are used and the way they behave, phonemes are divided into two: consonants and vowels. As we mentioned in the above section, Amharic language has 34 phonemes. These phonemes are all distinct since there is a difference in the speed and way of air flow within the articulatory organs to generate each of them. Basically, each phoneme differs in three different aspects: vibration level, place of articulation, and manner of articulation [22]. The following sections deal with Amharic consonants and vowels.

4.3.1 Amharic Consonants

In Amharic language there are 27 consonants out of the total 34 phonemes. Each of them mainly differs in the manner of articulation, place of articulation, and vibration level. When the air pressure generated by the lung passes through nearly closed vocal cords, vibration is created since the air vibrates the vocal folds. Therefore, vibration level is one way of classifying consonants. Consider two sounds /ስ/ and /ዝ/, the /ዝ/ sound creates vibration but the sound /ስ/ does not. One can easily see the difference between these two consonants by filling the vibration created on throat. Thus, these two sounds differ mainly in their level of vibration. Sound units that are blocked by the vocal cord while they are generated are called voiced sounds ("ነዛሪ ድምጽ") and others are called voiceless sounds ("ኢ_ነዛሪ ድምጽ"). Based on this, Amharic consonants are divided into two: voiced sound ("ነዛሪ ድምጽ") and voiceless sound ("ኢ_ነዛሪ ድምጽ").

After the air passes the vocal cord, it has two options. One is to go through the nasal cavity and the other is to go through the mouth and this is determined by velum. As the air passes through either the mouth or the nose, the sounds which will be generated are different. In general, consonants that are generated by nasal cavity are called nasal sounds (የስርጎ ድምጾች). Whereas, those that are created by mouth cavity are called oral sounds (የአፍ ድምጾች). These differences in consonants occur because of the difference in place of articulation. The place of articulation (also point of articulation) of a consonant is the point of contact, where an obstruction occurs in the vocal tract between an active (moving) articulator (typically some part of the tongue) and a passive (stationary) articulator (typically some part of the roof of the mouth) [47]. Even for nasal sounds the active and passive articulator are needed. The other way to see the difference in

consonants is to consider how the air generated from the lung behaves based on the way active articulators behave. This is referred as manner of articulation.

In general, the Amharic consonants based on the way they are generated are classified and depicted in Table 4.2 [22, 47]. As can be seen from Table 4.2, there are six types of consonants: stops, fricatives, nasals, affricates, semivowels and liquids, which are classified based on manner of articulations. Nasals, liquids and semivowels are always voiced; stops, fricatives and affricates can be voiced or unvoiced. They can also be classified as labials, alveolar, palatals, velars, labio – velar and glottals based on place of articulation.

Table 4.2 Categories of the Amharic consonants

		Labials		*Alveolar*		*Palatals*		*Velars*		*Labio-Velar*		*Glottals*	
Stops	Voiceless	p	ፕ	t	ት			k	ክ	kx	ኧ	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	ቈ						
Nasals	Voiced	m	ም	n	ን	nx	ኝ						
Liquids	Voiced			l	ር								
				r	ል								
Glides		w	ው			y	ይ						

Stop consonants are generated when the air is blocked and immediately released by articulatory organs. Hence, the stop consonants are distinguished by which articulatory organ the air is blocked, that is, whether the air is blocked by lips, tongue and velars, or tongue and alveolar. From this perspective when we see the sounds listed in Table 4.2, / ፐ፣ብ፣ጵ፣ ት፣ድ፣ጥ፣ክ፣ግ and /ቅ/ are stop consonants. /ፐ፣ብ፣ ጵ/ are created when the air is blocked by lips thus they are labials, and / ት፣ድ፣ጥ/ are created when the air is blocked by alveolar. In short, they are made by placing

the tongue against the alveolar ridge, the hard ridge in the top of the mouth, and behind your teeth. /ከ፣ግ ፣ቅ/ are created by velars; in this case the back of the tongue stops the air at the back of the hard palate [47].

Out of these stop consonant phonemes, /ፐ/ and /ጵ/ are not Amharic phonemes, rather they are borrowed from other languages like, Greek and Latin. These phonemes came with words like /ጳጳስ፣ጠረጴዛ፣ፖሊስ/ which means Pope, table and police respectively [47].

Fricatives involve letting the air slide through a narrow opening in the mouth. They can be prolonged for some time and the air is not completely blocked. For instance, in the case of consonants like ፍ, ስ, and ሽ, the air is neither blocked nor free while they are generated. Nasal sounds like /ም፣ን፣ኝ/ cannot be generated if the air is blocked from passing through the nasal cavity. Hence, there should be an air flow through the nasal cavity to produce them. The air will be directed to the nose since different parts of the mouth blocks it from passing through the mouth. Therefore, the difference between these sounds is that, in the case of /ም/, the air is blocked by lips, for the sound /ን/ the air is blocked by alveolar and tip of tongue; whereas the sound /ኝ/ is created when the air is blocked by palatals and middle part of the tongue.

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

4.3.2 Amharic Vowels

One of the basic differences between consonants and vowels is that in the case of vowels, as it is explained in [47], the air generated by the lung will vibrate the vocal cord since the vocal cord is slightly closed when vowels are generated. Moreover, in generating different vowels the tongue plays an important role. The contribution of tongue can be seen in two different aspects. The first one is the way the tongue is positioned in terms of height and the second one is the movement of tongue to the front and back of the mouth. In addition to tongue, the shape of lips has also a

contribution in changing the vowels sound when it varies from rounded to non-rounded (flat) [47].

In Amharic there are seven vowels, these are ኧ, ኡ, ኢ, ኣ, ኤ, እ and ኦ. All are voiced and oral sounds. These vowels can be found with each grapheme, that is, each grapheme in Amharic is not a single sound rather it is assimilation of consonant with a vowel. Look at the following example given in Table 4.3 With the seven orders of "መ" and "በ", each of the vowels ኧ, ኡ, ኢ, ኣ, ኤ, እ and ኦ also occurs. Therefore, the sounds of letters of the Amharic language are a combination of a consonant and vowel [47].

Table 4.3 Amharic vowels along with seven orders of a consonant.

Order	First	Second	Third	Fourth	Fifth	Sixth	Seventh
V / C	ኧ e	ኡ u	ኢ ii	ኣ a	ኤ ie	እ ix	ኦ o
/m/	መ /me/	መ /mu/	ሚ /mii/	ማ /ma/	ሜ /mie/	ም /mix/	ሞ /mo/
/b/	በ /be/	ቡ /bu/	ቢ /bii/	ባ /ba/	ቤ /bie/	ብ /bix/	ቦ /bo/

Amharic vowels are divided into two based on the shape of the lip as rounded and unrounded. When one generates the sounds of "ኢትዮጵያ" ("Ethiopia") and "ሄዱ", ("hiedu") (which means "They went") the sound /ኢ/ in "ኢትዮጵያ" needs the lips to be flat and somewhat spread. Whereas, the sound /ኡ/ in "ሄዱ" does need the lips to be rounded in order to be generated. In general, vowels ኡ and ኦ are classified as rounded. Whereas, ኧ, ኢ, ኣ, ኤ, and እ are classified as non-rounded as far as the lip shape is concerned [23, 47].

The Amharic vowels, with respect to the movement and position of the tongue, exhibit different characteristics. The vowels /ኢ/, /እ/ and /ኡ/ are created when the tongue moves to the roof of the mouth. The sound /ኣ/ is created when the tongue takes the lower part of the mouth. Whereas the sounds /ኧ/, /ኤ/, and /ኦ/ are created when the tongue takes the middle position, that is when it is

not too high or too low. Hence, based on the height of the tongue vowels are divided into high, middle, and low.

The tongue also moves to the front and back of the mouth while we speak vowels. When it does this movement, it creates different vowels at different positions. In this perspective, /ኈ/, and /ኊ/ are front, /ኍ/, /ኧ/, and /ኣ/ are middle and /ኡ/, /ኦ/ are at back. Figure 4.1 [22, 47] summarizes all combination of positions and movement of tongue along with all possible vowels that can be created.

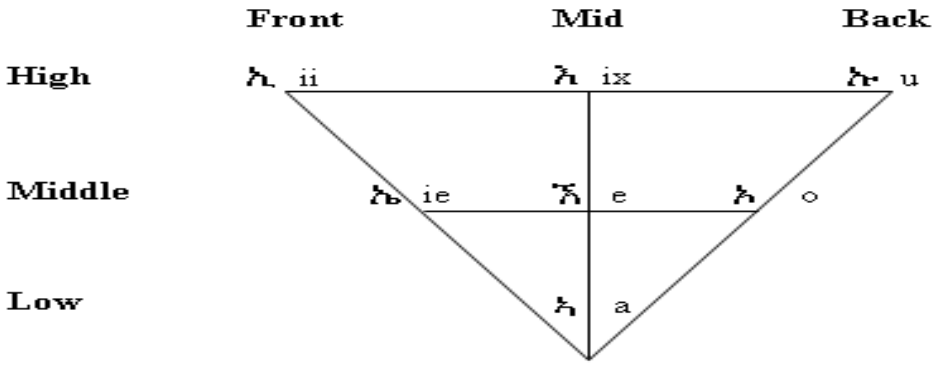

Figure 4.1 IPA maps of the Amharic vowels

4.3.3 Transcription of Amharic Characters

Transcription refers to the writing down of spoken utterance using a suitable set of symbols. In its original meaning the word implied converting from one representation (e.g. written text) into another (e.g. phonetic symbols).

Hence, in order to describe the correct pronunciation some kind of symbolic presentation is needed. There were some efforts made to construct language independent phonemic alphabets during the last decades. Among these, IPA (International Phonetic Alphabet) is one. IPA, which is one of the best known language-independent phonemic alphabets, consists of a huge set of symbols for phonemes, suprasegementals, tones/word contours, and diacritics.

According to [22], transliteration can be used as an alternative to the IPA alphabets. The transliteration scheme used in this research work is shown in Appendix A [22]. It is designed

based on the orthographic ordering of the script and acoustic similarity of the letters. The transliteration scheme used in [22] is adopted in this study. Based on the transliteration scheme, the Amharic word "መዳን" ("medan") (which means "to be saved") is transcribed into [medan]. Square brackets, [], are used during phonetic transcription.

CHAPTER FIVE

CLASSIFICATION AND REGRESSION TREES

5.1 Introduction

Classification and Regression Trees is a classification method which uses historical data to construct so-called decision trees. Decision trees are then used to classify new data. Decision trees contain a binary question (yes/no answer) about some feature at each node in the tree. The leaves of the tree contain the best prediction based on the training data. Decision lists are a reduced form of this where one answer to each question leads directly to a leaf node. A tree's leaf node may be a single member of some class, a probability density function (over some discrete class), a predicted mean value for a continuous feature or a gaussian (mean and standard deviation for a continuous value). Theorectically the predicted value may be anything for which a function can defined that can give a measure of *impurity* for a set of samples, and a distance measure between impurities. The basic algorithm is given a set of samples (a feature vector) find the question about some feature which splits the data minimising the mean "impurity" of the two partitions. Recursively apply this splitting on each partition until some stop criteria is reached (e.g. a minimum number of samples in the partition)[39].

In CART tree construction consists of three steps: building a tree, pruning sub trees and selecting an optimal tree. To build a tree we need a training set L in the form $\{f_n, y_n\}$, where f_n are feature vectors of corresponding objects and y_n values of the dependent variable. We start with the tree consisting only of a root node t_1 containing all of the cases in L. The task now is to find the optimal binary split of the data. For real-valued feature i all splits of the form $f_n^i < \tau$ are tested. For the M-valued categorical feature i, splits have the form $f^i \in \Theta$ where Θ goes through all subsets of the set of all possible values of the feature i. The best split across all features is selected and the data in the root node is splitted and sent into nodes .This procedure is

applied recursively to all descendants until a stopping condition is full-filled. Root mean square error is used as a splitting criterion:

$$\sum_{f \in t_L} (d(f) - \bar{y_L})^2 + \sum_{f \in t_R} (d(f) - \bar{y_R})^2 \tag{13}$$

After the tree construction phase, we have a relatively large tree. We successively prune some branches and construct a tree sequence $T_{\max} \supseteq \dots \supseteq T_k \supseteq \dots \supseteq T_K = t_1$. Among these trees we select the best tree using a test sample independent on a training sample [39].

5.2 CART for duration modeling

If a machine learning approach is taken, a database is used to infer the parameters of the algorithm making the duration prediction. Segment duration is influenced by a number of contextual factors such as segment identity, stress, accent, identity of preceding and following segments, position of a target segment within a syllable, word, and utterance. CART can deal with both types: categorical and real-valued features, but it has to be admitted that the main requirement for features is: all of them have to be computable from the raw text, otherwise the model would be inapplicable in TTS systems. The segmental durations are predicted by traversing the decision tree (CART) starting from the root node, taking various paths satisfying the conditions (feature values) at intermediate nodes, till the leaf node is reached. The leaf node contains the value of segmental duration prediction.

5.2.1 Classification and Regression tool –Wagon

As part of tools for statistical modelling speech includes methods for automatically building decision trees and decision lists from features data to predict both fixed classed (classification) or gaussians (regression). Wagon is the basic program that implements CART classification algorithm.

The basic CART building algorithm is a *greedy algorithm* in that it chooses the locally best discriminatory feature at each stage in the process. This is suboptimal but a full search for a fully optimized set of question would be computationally very expensive. Although there are

pathological cases in most data sets this greediness is not a problem. The basic building algorithm starts with a set of feature vectors representing samples, at each stage all possible questions for all possibles features are asked about the data finding out how the question splits the data. A measurement of impurity of each partitioning is made and the question that generates the least impure partitions is selected. This process is applied recursively on each sub-partions until some stop criteria is met (e.g. a minimal number of samples in a partition).

The *impurity* of a set of samples is designed to capture how similar the samples are to each other. The smaller the number the less impure the sample set is. For sample sets with continuous predictees Wagon uses the variance times number of sample points. The variance alone could be used by this overly favour very small sample sets. As the test that uses the impurity is trying to minimise it over a partitioning of the data, multiple each part with the number of samples will encourage larger partitions, which we have found lead to better decision trees in general.

Wagon has to automatically form questions about each feature in the data set. For discrete features questions are built for each member of the set, e.g. if feature n has value x. Our implementation does not currently support more complex questions which could achieve better results (though at the expense of training time). Questions about features being some subset of the class members may give smaller trees. If the data requires distinction of values a, b and c, from d e and f, our method would require three separate questions, while if subset questions could be formed this could be done in one step which would not only give a smaller tree but also not unecessarily split the samples for a, b and c. In general subset forming is exponential on the number items in the class though there are techniques that can reduce this with heuristics. However these are currently not supported. Note however the the tree formalism produced but Wagon does support such questions (with the operator "in") but Wagon will never produce these question, though other tree building techniques (e.g. by hand) may use this form of question.

For continuous features Wagon tries to find a partition of the range of the values that best optimizes the average impurity of the partitions. This is currently done by linearly spliting the range into a predefined subparts (10 by default) and testing each split. This again isn't optimal but does offer reasonably accuracy without require vast amounts of computation.

There are many ways to constrain the tree building algorithm to help build the "best" tree. Wagon supports many of theses. In the most basic forms of the tree building algorithm a fully exhaustive classifcation of all samples would be achieved. This, of course is unlikely to be good when given samples that are not contained within the training data. Thus the object is to build a classification/regression tree that will be most suitable for new unseen samples. The most basic method to achieve this is not to build a full tree but require that there are at least n samples in a partition before a question split is considered. We refer to that as the *stop* value. A number like 50 as a stop value will often be good, but depending of the amount of data you have, the distribution of it, etc various stop value may produce more general trees.

A second method for building "good" trees is to *hold out* some of the training data and build a (probably over-trained) tree with a small stop value. Then prune the tree back to where it best matches the held out data. This can often produce better results than a fixed stop value as this effectively allows the stop value to vary through different parts of the tree depending on how general the prediction is when compared against held out data.

It is often better to try to build more balanced trees. A small stop value may cause the tree building algorithm to find small coherent sets of samples with very specific questions. The result tree becomes heavily lop-sided and (perhaps) not optimal. Rather than having the same literal stop value more balanced trees can be built if the stop value is defined to be some percentage of the number of samples under consideration. This percentage we call a *balance* factor. Thus the stop value is then the largest of the defined fixed stop value or the balance factor times the number of samples.

A good technique we have found is to build trees in a *stepwise* fashion. In this case instead of considering all features in building the best tree, we increment build trees looking for which individual feature best increases the accuracy of the build tree on the provided test data. Unlike within the tree building process where we are looking for the best question over all features this technique limits which features are available for consideration. It first builds a tree using each and only the features provided looking for which individual feature provides the best tree. The selecting that feature is builds n-1 trees with the best feature from the first round with each of the

remaining features. This process continues until no more features add to the accuracy or some stopping criteria (percentage improved) is not reached.

This technique is also a greedy technique but we've found that when many features are presented, especially when some are highly correlated with each other, stepwise building produces a significantly more robust tree on external test data. It also typically builds smaller trees. But of course there is a cost in computation time.

The input data for wagon should consist of feature vectors, and a description of the fields in these vectors. A feature vector is a file with one sample per line, with feature value as white space separated tokens. If the features values contain whitespace then they must be quoted them using double quotes. Each vector must have the same number of features. Features may be specified as "ignored" in the description. By default the first feature in a data file is the predictee, though at least in wagon) the predictee field can be named at tree building time to be other than the first field.

It is it common to have thousands, even hundreds of thousands of samples in a data file, and the number of features can often be in the hundreds, though can also be less than ten depending on what it describes.

5.2.2 Classification and Regression tool –Wagon

Generated tree files are written as Lisp s-expressions as this is by far the easiest external method to represent trees. Even if the trees are read by something other than Lisp it is easy to write a reader for such a format. The syntax of a tree is

TREE ::= LEAF | QUESTION-NODE

QUESTION-NODE ::= "(" QUESTION YES-NODE NO-NODE ")"

YES-NODE ::= TREE

NO-NODE ::= TREE

```
QUESTION ::= "(" FEATURENAME "is" VALUE ")" |
        "(" FEATURENAME "=" FLOAT ")" |
        "(" FEATURENAME "<" FLOAT ")" |
        "(" FEATURENAME ">" FLOAT ")" |
        "(" FEATURENAME "matches" REGEX ")" |
        "(" FEATURENAME "in" "(" VALUE0 VALUE1 ... ")" ")"

LEAF ::= "(" STDDEV MEAN ")" |
      "(" "(" VALUE0 PROB0 ")" "(" VALUE1 PROB1 ")" ...
MOSTPROBVAL ")" |
      any other lisp s-expression
```

Note that not all of the question types are generated by Wagon but they are supported by the interpreters.

The leaf nodes differ depending on the type of the predictee. For continuous predictees (regression trees) the leaves consist of a pair of floats, the standard deviation and mean. For discrete predictees (classification trees) the leaves are a probability density function for the members of the class. Also the last member of the list is the most probable value. Note that in both cases the last value of the leaf list is the answer desired in many cases.

CHAPTER SIX

DESIGN AND INTEGRATION OF DURATION MODEL INTO AMHARIC SYNTHESIZER

6.1 Introduction

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

As mentioned in Chapter one, TTS system is composed of two parts; Natural Language Processing (NLP) and Digital Signal Processing (DSP). The NLP component includes pre-processing, sentence splitting, tokenization, text analysis, homograph resolution, parsing, pronunciation, stress, syllabification and prosody prediction. Working with pronunciation, stress, syllabification and prosody prediction sometimes is termed as linguistic analysis. Whereas, the DSP component includes segment list generation, speech decoding, prosody matching, segment concatenation and signal synthesis. Three basic prosody components are modeled: segment durations, intonation and intensity. Duration model is a standard part of current speech synthesizers.

Since we are dealing with the task of duration modeling to improve the quality of the speech produced by Amharic TTS, before we start to discuss about the construction and integration of duration model it is worth pausing to explain first the development of new speech synthesis system for Amharic language to which we will incorporate the duration model.

6.2 Building a Unit Selection Cluster Voice for Amharic

Our TTS system is implemented based on the unit selection approach using Festival speech synthesis framework. Festival is a free, language independent speech synthesis engine that can be used for developing text-to-speech synthesis systems [50]. The Festival [50] synthesis system uses a cluster unit selection technique for selecting speech units from a speech database. In this method, the speech inventory is divided into clusters, where each cluster holds units of the same phone class based on their phonetic and prosodic context. The appropriate cluster is selected for a target unit, offering a small set of candidate units. Units that minimize acoustically defined target and join costs are then selected from a cluster.

An outline of the steps to build a unit selection synthesizer are given below. A more detailed description of same is available in [51].

- Design speech and text corpus
- Creating letter-to-sound rules and phoneset
- Building utterance structures for the database
- Generating speech unit clusters
- Building the unit synthesizer

What have been done in each step to build unit selection voice for Amharic explained below.

6.2.1 System Architecture

The aim of this thesis as mentioned in Chapter one is to build a model for segmental duration for Amharic TTS system that is able to predict duration for input text. But in order to assess the impact of the duration model we need to build Amharic TTS to which we can incorporate the duration model.

Figure 5.1 shows the system architecture of the Amharic speech synthesizer using cluster unit selection that we designed and implemented in order to integrate our duration model. As the system architecture shows, the synthesizer has two parts: text analysis, and speech synthesis part. Text analysis part uses grapheme to phoneme convertor to match the word to its pronunciation. Whereas, the synthesis part select best sequence of units for target specification produced at the end of text analysis, and finally generates the speech waveform from the speech parameters.

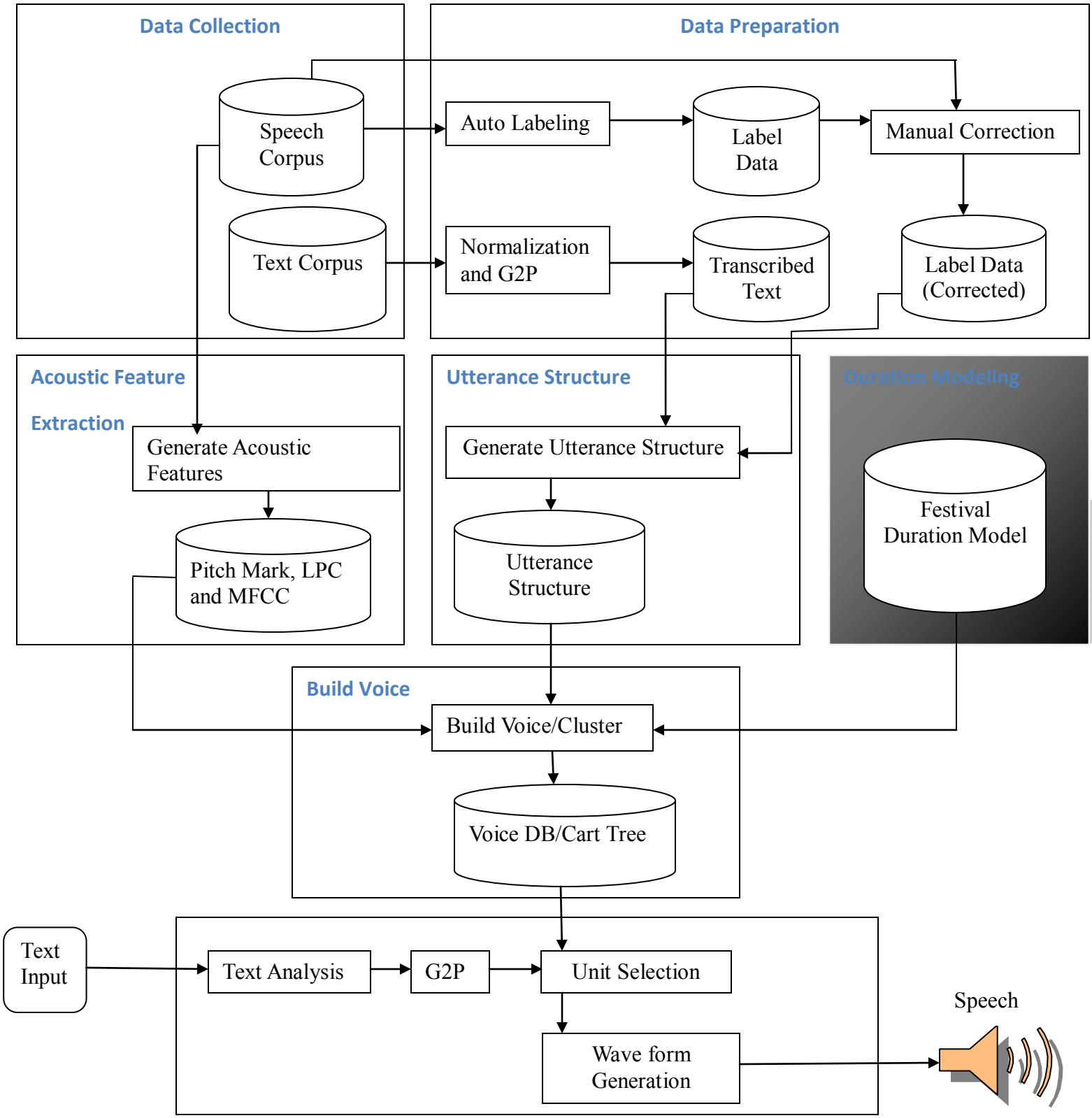

Figure 6.1 Block diagram of Amharic TTS system using unit selection based synthesis in the Festival speech synthesis framework

6.2.2 Data Collection and Preparation for Unit Selection Voice

Unlike diphone database which are carefully constructed to ensure specific coverage, one of the advantages of unit selection is that a much more general database is desired. However, although voices may be built from existing data not specifically gathered for synthesis there are still factors about the data that will help make better synthesis [50]. Like diphone databases the more cleanly and carefully the speech is recorded the better the synthesized voice will be. As we are going to be selecting units from different parts of the database the more similar the recordings are, the less likely bad joins will occur. However unlike diphones database, prosodic variation is probably a good thing, as it is those variations that can make synthesis from unit selection sound more natural. Good phonetic coverage is also useful, at least phone coverage if not complete diphone coverage. Also synthesis using these techniques seems to retain aspects of the original database [50].

We collected speech and text corpus for this thesis that have been used as experimental data in [25]. After the data is collected the next step is to check recorded utterances against the transcription text in order to design the prompt list in Festival format and correct the label manually. While preparing speech database for our TTS we identified recorded utterances with different problems against the transcription text. Some of them are noisy utterance and mis-spoken data that the speaker might have made a mistake in the content. Mistakes can be actual content (it is easy to read some words wrongly). Appropriate modifications have been made to get them ready to be used in voice building process. By doing so the speech and text corpora has been built.

6.2.3 Labeling the Utterance

From the activities of data collection and preparation, the speech data and transcribed texts are already available. Hence, the labeled utterance is left to generate the utterance structure. The process that generates the labeled utterance is labeling. Labeling is the process of giving a label for each speech signal in the utterance. Unit selection synthesizers are highly sensitive to the accuracy of labeling. Bad labels will adversely affect the quality of synthesis in a number of

ways. The phone label itself can be incorrect, potentially causing the wrong word to be said, or said with an undesired accent. Or the label boundaries can be inaccurate – e.g. spilling over into neighboring segments thereby distorting the speech with impurities. More subtly, bad labels can misdirect the join algorithm, degrading the choices effectively available for neighboring unit selections [52]. However it is time taking and laborious, as part of our efforts to improve speech synthesis, we have labeled the speech data base thoroughly using a tool called WaveSurfer. In order to suit the requirement of Festival framework generated label files are converted into Festival label file format using script that we wrote for this particular purpose. Sample labeled speech is included in Appendix B.

6.2.4 Incorporation of Amharic Phoneset and Grapheme to Phoneme Convertor

In Festival, the natural language modules require two types of analyses. The first type is the language specific analysis such as phones, lexicon, tokenization and others. The second type is the speaker specific analysis, where prosodic analysis such as duration and intonation are the main issues [51].

The phone-set definition is the first text analysis module in which every phoneme of the alphabet is classified according to phone features like consonant voicing and vowel height. The second text analysis module is the lexicon module. This module covers methods for finding the pronunciation of a word. This is done either by a lexicon, i.e., a large list of words and their pronunciation or by some letter to sound or grapheme to phoneme rules [51].

The Amharic phoneset incorporated in festival, corresponding to their characterizing features, are shown in Appendix C. Each phone has eight features that describe how the vocal organs behave when the sound is uttered. These features are vowel/consonant identification, consonant voicing, place of articulation, consonant type, vowel length, vowel height, vowel frontness, and lip rounding.

The grapheme to phoneme convertor is used to convert an orthographic text into its corresponding phonetic representation. We implemented the grapheme to phoneme conversion

architecture by making certain modification of syllabification algorithm proposed in [21]. The C# based syllabification program implemented in [21] which is Graphical based system and modified into C++ command line based G2P system as per the requirement of Festival tools.

After incorporation of Amharic phoneset and the Amharic grapheme to phoneme convertor into festival, it provides the label files for each sentence in the prompt list. We have made manual label correction using the label automatically generated and the corresponding wave file.

6.2.5 Generating Utterance Structure

Festival represents each speech unit internally with a data structure called an „utterance". The „utterance" structure holds all the relevant phonetic and prosodic information related to a speech unit within this data structure. The utterance structure also consists of a set of relations over a set of items. Each item represents an object such as a word, segment, syllable, phonemes, etc. Whereas, relations show how these items relate to each, for example, the relation between two phonemes in that specific utterance [50]. The phonetic information in an „utterance" structure describes the position of the speech unit in the word it appears and the information of units adjacent to it. Prosodic information holds information about the duration and pitch of the unit. Festival provides relevant scripts for building „utterance" structures for each speech unit.

As well as using Utterance structures in the actual runtime process of converting text to speech we also use them in database representation. Basically we wish to build utterance structures for each utterance in a speech database. Once they are in that structure, as if they had been (correctly) synthesized, we can use these structures for training various models. For example given the actually durations for the segments in a speech database and utterance structures for these we can dump the actual durations and features (phonetic, prosodic, context, etc.) which we feel influence the durations and train models on that data.

The label files generated using WaveSurfer were not directly used to generate the utterance structure since there is a difference in data format needed by Festival, which are used to generate

the utterance structure. Hence, in this thesis the format of the label file generated by WaveSurfer is changed by using the script we have written for this particular purpose.
Once utterance structure is generated for each utterance the next step is to generate speech unit clusters and build the synthesizer. A number of offline processes need to be done to create clusters of speech units belonging to the same class. These processes include building coefficients for acoustic distances (MFCC, F0 and energy coefficients), creating distance tables for each class of units based on acoustic distances and generation of features for building CART trees. Festival provides scripts to perform these tasks. Once these are in place, Festival uses a CART building program - „wagon" [48], to generate a CART tree for each class of units.

Using the letter-to-sound rules, phoneset and clusters of each speech unit built in the previous steps, Festival generates the necessary files that need to be used along with the core Festival speech synthesizer to build a unit selection synthesizer for a particular language using appropriate scripts. The final stage in building a cluster model is to collect the generated trees into a single file and dumping the unit catalogue, i.e., the list of unit names and their files and position in them. Part of units catalogue produced at the end of building Amharic unit selection synthesizer is shown in Appendix D.

6.3 Amharic Duration Modeling

By default, unit selection voices in Festival are accompanied by a durational model that has been trained from the 'f2b' voice of the Boston University Radio News Corpus. The duration model used by Festival framework is depicted in Figure 5.1. In order to improve the naturalness of the resulting Amharic voice in this type of synthesis this research has focused in building speaker-specific duration model that alter or predict the prosodic properties of the synthetic speech. Such properties must be adapted to each language to represent the prosodic particularities of the same one. The definition of a new prediction of duration model is very important because if the duration of each phoneme pronounced by the synthesis system is close to the natural duration, it makes appreciably more natural the synthetic voice.

This thesis proposes a self-learning duration model using a general segmented corpus made by Amharic speaker to allow produce a more representative voice model of Amharic for training

and considering the characteristics (or parameters) of every phoneme, then the model will predict the estimated duration of the phoneme.

6.3.1 Description of the Database

Speech corpora design is a fundamental issue to be handled for developing appropriate prosody, in particular, duration models. In order to develop a duration model for use in the Amharic unit selection voice, single native and male speaker reading corpus was created. Most parts consist of single sentences taken from a variety of contexts, (political news, economy news, sport news, health news, fictions, Bible, penal code and Federal Negarit Gazzeta) [25]. The entire size of the corpus is 1hour, 16 minutes and 29 seconds.

In order to begin the experimentation, carrying out measurements that allow statistical models of duration of phonemes, it is necessary to have a labeled corpus, putting special attention in the borders of the phonemes to obtain a better accuracy in the measurement of the durations to classify. The labels indicate the beginning and end of the phonemes in time. Using the automatically produced in [25] and manually verified labels, the durations on corpus were developed. The resulting speech corpus consists of 899 sentences, 7, 232 words and 33,500 phonemes.

6.3.2 Features Selection and Extraction

Even though, all the features listed in Chapter 2 have influence on duration of the given phone, with the following reasons only some of them have been chosen.

1. Since our main interest is speech synthesis of textual data we have focused at the derivation of features that could be extracted from text.
2. Those features their relative importance have been assessed in many languages and found to be effective for segmental duration prediction are included.
3. Another point that we considered to select features is availability of the tool for Amharic to automatically extract from the input Amharic text.

The following influencing factors for segmental duration are defined in accordance with the justification presented above. They are defined on the level of the respective feature, i.e. phone-level, syllable-level, word-level, sentence -level.

- Identity of the current phoneme; this feature is categorical and it has 35 possible values. (a, b, c, cx, d, e, f, g, h, ie, ii, ix, j, k, l, m, n, nx, o, p, px, q, r, s, sx, t, tx, u, ua, v, w, xx, y, z, zx)
- Lip rounding of the current phoneme; this feature is categorical and it has 3 possible values. (rounded, unrounded and not relevant)
- Syllable‟s onset-size; this is the number of phones before the vowel of the syllable.
- Position of parent syllable in the word; the position of the syllable in the word it is related to. The index counts from 0.
- Parent syllables break information; Break level after the parent syllable. This feature is categorical and it has 3 possible values: 0 for word internal syllables, 1 for syllables occurring in word boundary, 4 for syllables occurring in sentence boundary.
- Identity of the preceding phoneme; This feature is categorical and it has 36 possible values (a, b, c, cx, d, e, f, g, h, ie, ii, ix, j, k, l, m, n, nx, o, p, px, q, r, s, sx, t, tx, u, ua, v, w, xx, y, z, zx, pau)
- Position of preceding phoneme in syllable this feature is categorical and it has 2 possible values (onset, coda)
- Preceding phoneme syllable‟s coda-size; this is the number of phones after the vowel of the syllable.
- Identity of the following phoneme; This feature is categorical and it has 36 possible values (a, b, c, cx, d, e, f, g, h, ie, ii, ix, j, k, l, m, n, nx, o, p, px, q, r, s, sx, t, tx, u, ua, v, w, xx, y, z, zx, pau)
- Manner of articulation of the following phoneme; this feature is categorical and it has 7 possible values (stops, fricative, affricate, approximant, lateral, nasal,0)
- Position of following phoneme in syllable this feature is categorical and it has 2 possible values (onset, coda)
- Voicing of next of the following phoneme ; this feature is categorical and it has 3 possible values (voiced, unvoiced or not relevant)

- Phone class of the following phoneme ; phoneme ; this feature is categorical and it has 2 possible values (vowel, or consonant)

In order to extract features we used a script provided by Festival. The script allows us extracts features, these cover phonetic context, and syllable, word position and sentence break information from utterance structures. The extraction process takes each phoneme and dumps the named feature values for that phone into a file for each utterance structure. These feature files are then concatenated into a single file.

The resulting feature vectors are split into two subsets: training dataset is used to develop duration models and test dataset is used to evaluate the performance of the model on unseen data. The test set consists 10% of the data and the remaining phonemes constitute the training set. The total number of segments in the training and test sets is 27,500 and 5500, respectively. Sample feature vectors have been shown at Appendix E.

The training set is further split to use be used as a held-out test set in the training phase. That means 10% of training set is used as held-out data that are used for the purpose of generating more balanced tree by using pruning technique. Also at this stage we removed all silence phones from the training and test set. This is, perhaps naively, because the distribution of silences is very wide and often files contain silences at the start and end of utterances which themselves aren't part of the speech content (they're just the edges).

6.3.3 Duration Model Construction

In this thesis, a CART based method is used to map linguistic features to phoneme durations. CART is a predictive model that can be viewed as a tree. CART provides interpretability so that underlying dynamics between input space and outputs can be clearly identified. They can also be applied to any data and require less parameter tuning. For running the CART tree building process, "Wagon" classification and regression tree tool [48] is used. Development process of Amharic duration model is illustrated in Figure 5.3 and portion of the duration model is included at Appendix F.

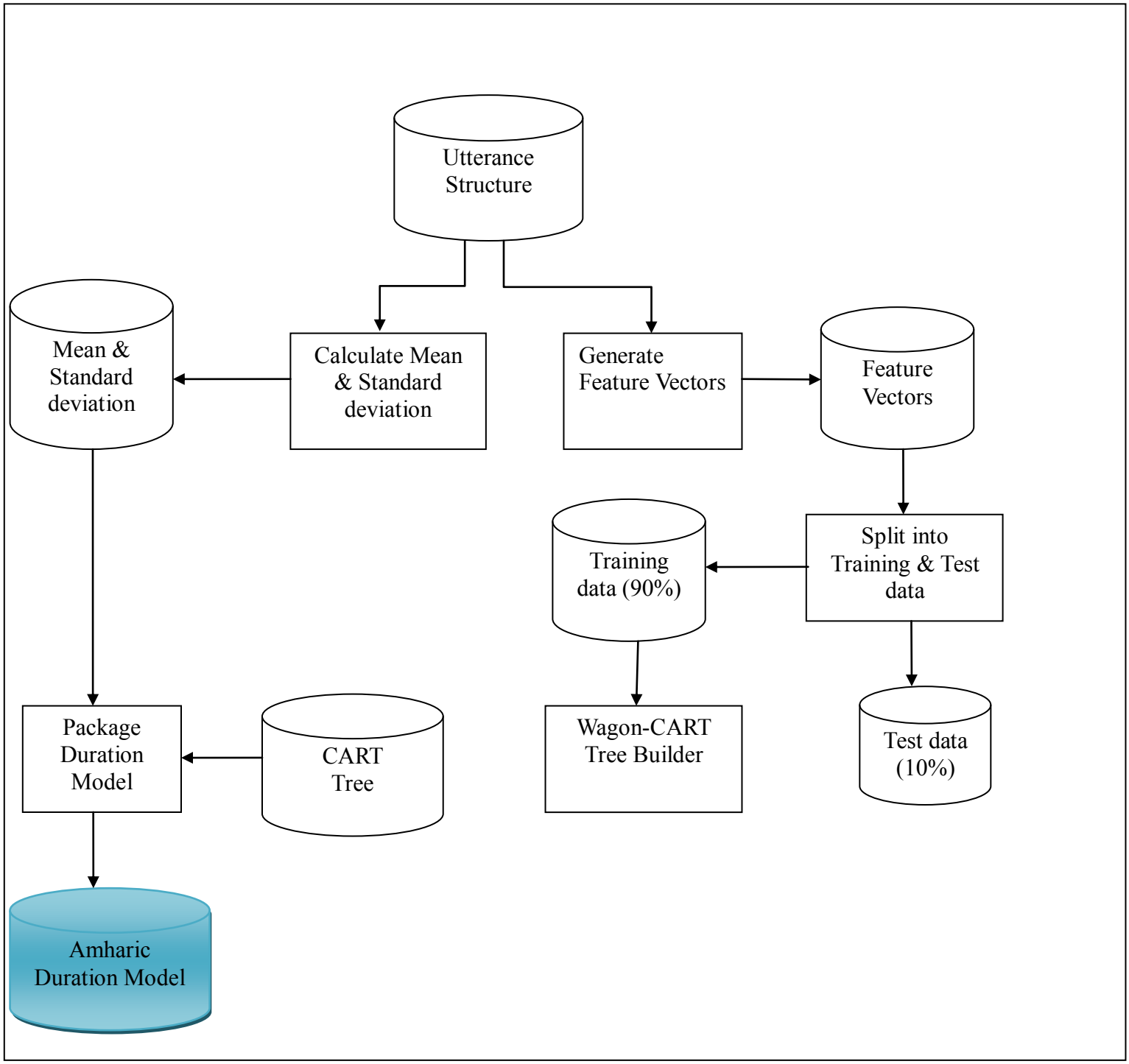

Figure 6.2 Development process of Amharic Duration Model

In order to construct the train and test sets for the phone duration modeling task, we calculated the mean and standard deviations of duration from the entries. In this way, we were able to construct models that predict the segment duration both directly using actual phone duration in seconds and also using the z-score of the phone durations. The z-score is a statistic quantity which indicates how many standard deviations an observation is above or below the mean. The z-scores allow comparison of observations from different normal distributions. The phone zscore duration is calculated by the formula (14) given below. A festival script in the festival distribution is used to load in all the utterance structures and calculate these values. The resulting means and standard deviations data for each phone is shown in the table below.

A particular unit *x* in p with duration d will have a phone-class zscore [52]

$$\mathbf{Zp}(x) = (\mathbf{d} - \bar{\mathbf{x}}) / \mathbf{s} \qquad (14)$$

Where d denotes absolute duration of an instance of phone x, $\bar{x}$ is average duration calculated by taking all instances of phone p and σ is standard deviation between instances in phone p. Table 5.1 shows the mean and standard deviation of all Amharic phoneme computed from the collected labeled speech file.

Table 6.1 Mean and Standard deviation of Amharic phones

Phone	Mean in second	Standard deviation in second
zx	0.132222	0.050132
v	0.059298	0.023393
nx	0.124576	0.03979
px	0.126092	0.030387
cx	0.118757	0.045096
c	0.127109	0.061129
j	0.098342	0.043007
z	0.086056	0.044686
p	0.085621	0.123783
w	0.082476	0.045487
d	0.075713	0.042476
xx	0.118911	0.048089
g	0.085194	0.038844
ii	0.047416	0.86956
l	0.071242	0.036011
tx	0.100465	0.041505
m	0.092978	0.638832
q	0.091054	0.037562
ie	0.110219	0.043851
b	0.081618	0.048268
f	0.102233	0.043659
o	0.092315	0.040687
h	0.05673	0.046123
n	0.077237	0.036423
y	0.076945	0.04688
r	0.055769	0.033235
ix	0.045793	0.038899
u	0.077587	0.043618
sx	0.130429	0.04737
a	0.124374	0.54555
s	0.107681	0.042199
k	0.058569	1.28752
e	0.06649	0.028533
t	0.081036	0.037402

In order to train a zscore model we need to convert the absolute segment durations. To do that we need the means and standard deviations for each segment in our phoneset. After prediction, the segmental duration is calculated back to absolute duration by using equation (14) shown above.

Z-scores are a better representation for the duration of phones as shown in [54] and give better results as stated in [50].

As we already have training and test data, the next step to forward the construction is to provide these data to wagon. Wagon CART tree builder to work it needs to know what possible values each feature can take. This can mostly be determined automatically but some features may have values that could be either numeric or classes, thus we used a pre-processing verification on the description file to get our desired result.

Now we can build the model itself. A key factor in the time this takes is the "stop" value that is the number of examples that must exist before a split is searched for. The smaller this number the longer the search will be, though up to a certain point the more accurate the model will be. The default in the distribution is 50 which may or may not be appropriate. Stop value for our duration model is directly adopted from the duration modeling script used in Festival.

6.3.4 Objective Evaluation of Duration Model

Objective evaluation of the duration models, by root mean squared prediction error (RMSE) and correlation between actual and predicted durations is performed.

The duration model is trained with train data (27,500 segments, 90% of the total segments) and evaluated with test data (5500 segments, 10% of the total segments). Correlation obtained between actual and predicted durations is 0.3901 and the root mean squared error (RMSE) of prediction is 0.8403. Note these values are not in the absolute domain (i.e. seconds) they are in the zscore domain.

For English, using the same tools and techniques as we used they found an RMSE value of 0.81 and correlation of 0.58. Moreover, in order to assess its impact on perception we conducted perceptual test that we will present in the next Chapter.

5.3.5 Segmental Duration Prediction

The segmental durations are predicted by traversing the decision tree (CART) starting from the root node, taking various paths satisfying the conditions at intermediate nodes, till the leaf node is reached. The path taken depends on various features like, the segment identity, preceding and following segment identities, position of the segment in parent syllable, position of the syllable in parent word etc. Leaf nodes hold a pair of numbers: the zscore mean of the cluster, and the cluster's standard deviation.

A portion of Amharic duration model for segmental duration prediction is shown in Figure 5.4. The tree assigns different durations for any segment when it occurs in different contexts. An average zscore of 1.9549 is assigned when it satisfies the following criteria: the phoneme is preceded by /ix/ (/ኽ/), parent syllable is the initial syllable in the parent word, the following segment is /l/ (/ል/). Otherwise an average zscore of -0.728674 is assigned when it satisfies the following criteria: phoneme is preceded by /ix/ (/ኽ/), parent syllable is the initial syllable in the parent word, the place of articulation of previous of preceding segment is labial, the following vowel has middle height and its identity is /ie/ (/ኺ/).

```
((p.name is ix)
 ((R:SylStructure.parent.pos_in_word < 1.4)
  ((n.name is l)
   ((1.39107 1.9549))
   ((pp.ph_cplace is l)
    ((n.ph_vheight is 2)
     ((n.name is ie)
      ((0.36093 -0.728674))
      ((0.705228 -0.4261)))
```

Figure 6.3 Portion of Amharic Duration Model

The algorithm to predict segmental duration followed by this particular portion of duration model is demonstrated in the flow chart of Figure 6.4

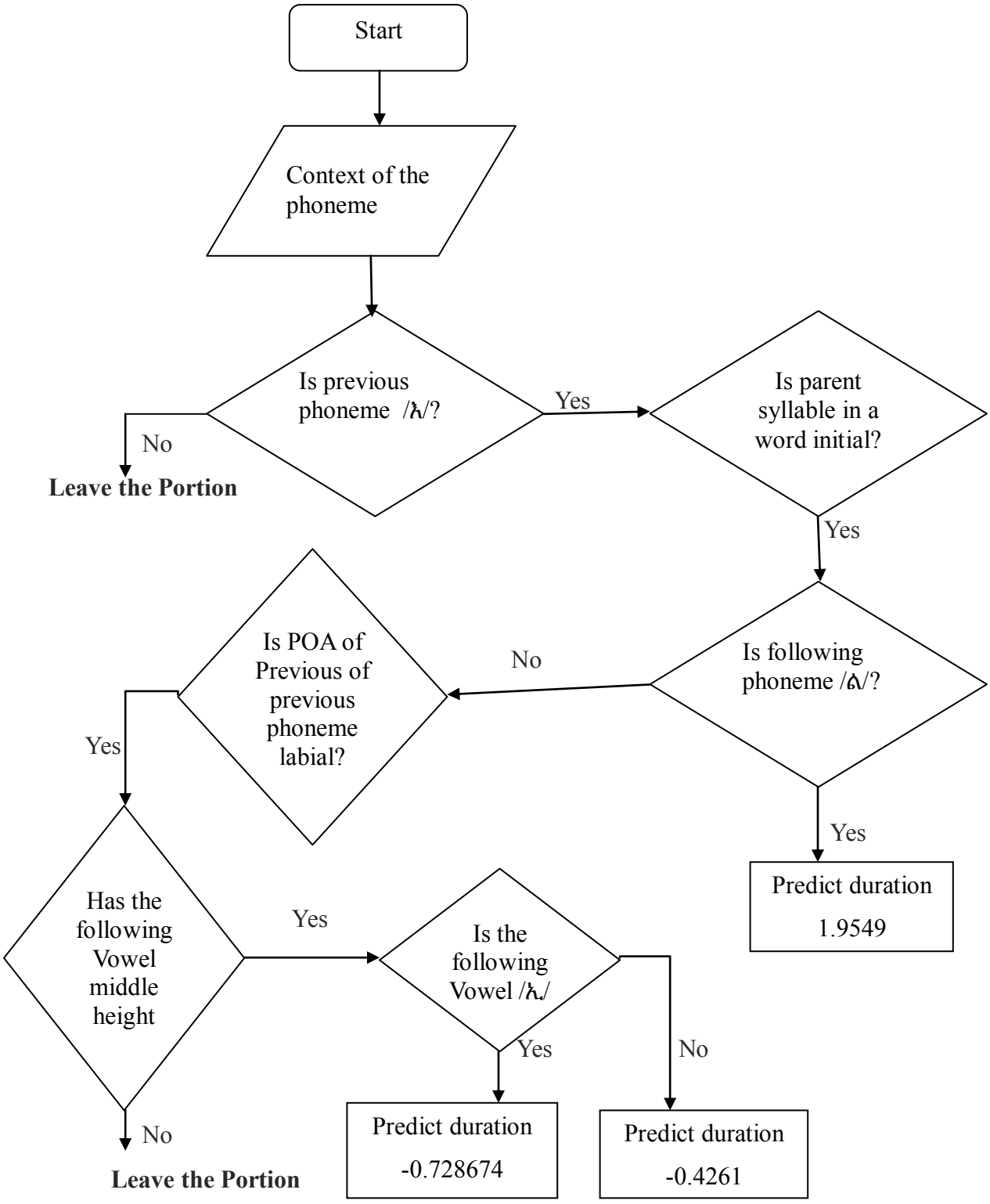

Figure 6.4 Flowchart to demonstrate portion of the algorithm of segmental duration prediction.

6.5 Integration of Duration Model into Amharic Unit Selection Synthesizer

The final step to complete the implementation of our speaker-specific duration model is to integrate it into the synthesizer that already we built using cluster unit selection. In order to do that in Festival framework, we have to package the model into a scheme file that can be used with a voice. This scheme file contains the means and standard deviations (so we can convert the predicted values back into seconds) and the prediction tree itself. Integration of Amharic duration model is depicted in figure 5.2.

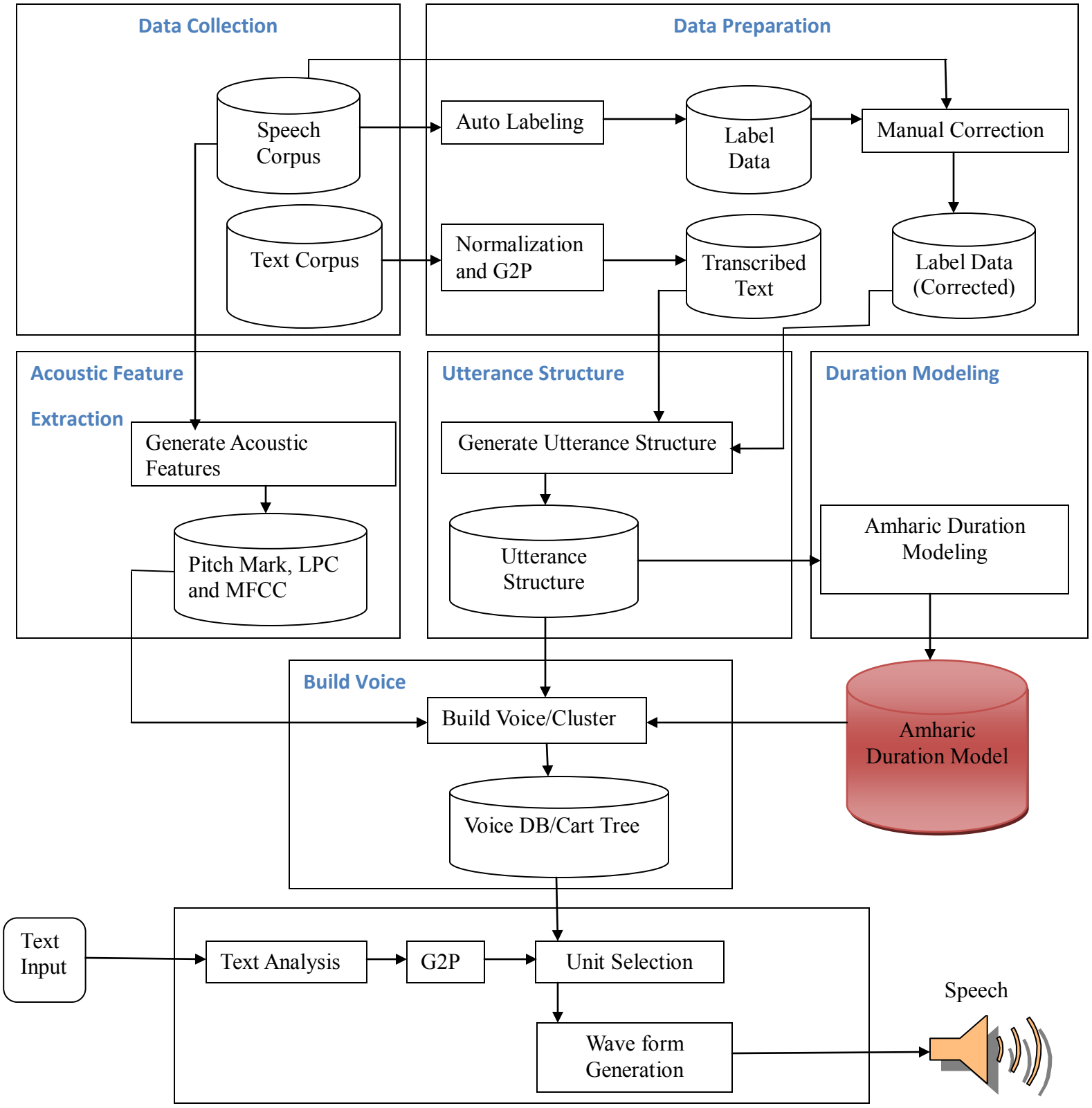

Figure 6.5 Block diagram of Amharic TTS system with the integration of Amharic Duration Model

CHAPTER SEVEN

PERCEPTUAL EVALUATION OF AMHARIC DURATION MODEL

7.1 Introduction

Perceptual evaluation is essential to determine the quality of synthesized speech. The perceptual evaluation in this thesis investigates the relevance predictive duration model for Amharic TTS.
In order to conduct subjective evaluation of Amharic duration model we have made perceptual test with two different versions of Amharic unit selection voice: the first one is without explicit duration model and the second one is with speaker-specific duration model built using the Amharic speech corpus.

In the following sections we present the method and the data used to conduct the evaluation. Finally the results of the evaluation will be presented.

7.2 Methods

In this research work mean opinion score (MOS) is used to test the output of the synthesized speech. MOS is an evaluation technique where evaluators indicate their assessments on a scale of bad (1) to excellent (5). Then the average of the opinion will be taken as the performance of the system.

As we stated in the above section the impact of the duration model in perception is assessed by comparing average result we are going to obtain to be ranked by speakers at the end of their perceptual judgment for the synthetic speech produced by the synthesizer with duration model and without duration model. So, two perceptual tests were carried out to evaluate the extent of naturalness and intelligibility of synthetic speech generated by the two versions speech synthesizer.

7.3 Data Preparation and Prototype Design for Testing

Twenty sentences with various lengths were selected from different sources, which were different from the data set used in the voice construction and duration modeling. These sentences are synthesized with the synthesizer without duration model and the output utterances are kept to be used for testing. Resynthesis was performed for the same sentences with the synthesizer with the duration model and the resulting utterances kept separately from previously generated ones.

To this end, we have a total of forty testing utterances generated from both versions of Amharic unit selection synthesizer. The next step to forward the evaluation is to devise a mechanism that will help us to play the utterances randomly without predefined order between same kind of utterances that were produced with and without duration model.

In order to meet the above requirements and avoid unnecessary effort, resource and bias, we developed a prototype using C++ code. The source code is included at Appendix G. The program plays utterances randomly and alternatively between synthesized with and without duration model. At the end of each utterance it prompts speakers to rate for the wave file it played and store results for forty utterances for both intelligibility and naturalness test.

7.4 Evaluation Results and Analysis

We conducted perceptua1 tests on 10 people who are native speakers of the Amharic language: 2 females and 8 males. All subjects are 20 to 40 years old in age. Each subject listens to all of the 40 sentences and gives a ranking value for the intelligibility of the speech and naturalness. They evaluated based on the quality of the speech output by giving a measure of quality as shown in Table 6.1.

Table 7.1 Perceptual Evaluation Categories

Category	Measure
Excellent	5
Very Good	4
Good	3
Fair	2
Poor	1
Very Poor	0

Table7.2 and Table7.3 show summary of evaluation average results found for intelligibility and naturalness. Table 7.2 is for the synthesizer without duration model and Table7.3 is for synthesizer with duration model. In order to illustrate the comparison between results found for both cases we presented Table7.3. As it can be seen from the result, the synthesizer with duration model provides better quality speech in terms of its intelligibility and naturalness.

Table 7.2 Results for synthesizer without duration model

	Intelligibility	Naturalness
Mean	3.31	3.33
Standard Deviation	1.225	0.956

Table 7.3 Results for synthesizer with duration model

	Intelligibility	Naturalness
Mean	3.5	3.58
Standard Deviation	1.235	1

The overall result shows that in both cases synthesizer provide good quality speech in terms of naturalness than intelligibility. Moreover, modeling duration for phonemes for Amharic speech synthesizer has shown improvement in quality both in its naturalness and intelligibility. We can make a conclusion that accurate duration assignment done by the TTS have great role for perceived naturalness and intelligibility of synthetic speech sounds.

CHAPTER EIGHT

CONCLUSIONS AND RECOMMENDATIONS

Prosody refers to characteristics of speech such as intonation, timing, stress, loudness, and other acoustical properties imposed by articulatory, emotional, mental, and intentional states of the speaker. One of the most prominent components of prosody is considered as duration. Duration modeling is important because TTS systems need to generate speech units with appropriate durations in order to produce natural sounding synthetic speech.

Many researches have been conducted over the years in the field of duration modeling which utilize different techniques or features. Duration modeling techniques can be divided into two major categories, the rule-based approaches and the corpus-based. State-of-the-art is dominated by corpus-based approaches. They appeared due to the increasing computational power and availability of large corpora. When large speech corpora and the computational means for analyzing these corpora became available, new data-driven approaches based on Classification and Regression Trees, linear statistical models and Artificial Neural Networks are proposed.

In this thesis, CART based data-driven duration model for Amharic language is developed and integrated into unit selection text to speech synthesizer.

Festival Speech Synthesis system has been selected for designing Unit selection Amharic synthesizers because the design processes does not require it to start from scratch. Since the quality of unit selection voice is highly dependent on accuracy of labeling, prior to the development of the synthesizer we have thoroughly labeled speech corpus of 899 utterances. In order to build the NLP module we used Amharic phoneset and grapheme to phoneme convertor. After the extraction of phonetic, prosodic and acoustic features from each utterance, cluster trees of each phoneme have been built using the scripts provided by Festival. DSP part have been adopted from Festival which implements unit selection algorithm to select appropriate units and to synthesize speech for input text.

In the task of duration modeling we identified features that affect duration of Amharic phones and we extracted phonetic features from utterance structures generated at the middle of offline database preparation for unit selection synthesizer. After we got feature vectors for all instances of phones that describe their context in natural utterances, we made a split of the whole corpus into 90 % for training and 10% for testing data. This data set has been fed to CART to get our duration model. The last step to achieve the implementation of the duration model is to integrate it into the TTS.

In order to evaluate the duration model we conducted objective and subjective tests. In objective test we have found reasonably acceptable RMSE and CC values as we compare with much researched language like English. In subjective evaluation, respondents were given speech synthesized by Amharic unit selection with and without duration then they gave a rank for each sentence for different criteria. Based on the value of the MOS, synthesizer with duration model performs better than that of the synthesized without duration model counterpart for both naturalness and intelligibility criteria. Finally we draw a conclusion that by using duration model for Amharic TTS, it is possible to contribute to the perceived naturalness and intelligibility of synthetic speech.

In this study, a duration model is developed that predicts duration of phones from a given linguistic context, by best selection of parameters from the decision tree. However, the duration model is attempted in [43], they didn't cover all Amharic phonemes and to integrate the model is left as future work. So from different angles that include optimality of the speech database, number of features used to model duration pattern and successful integration of the model, this thesis can be considered as pioneer work in duration modeling for Amharic phonemes. In order to control the prosody and get high quality of synthetic speech to be produced by Amharic TTS systems, we would like to recommend the following future works:

- Development of large corpora with taking care of data sparsity that would help for comprehensive analysis and modeling of segmental duration

- Investigate and exploit duration modeling in the field of emotional speech synthesis to synthesize expressive speech, furthermore synthesizing conversational speech rather than "reading style" speech.
- Develop a method of determining stress level of Amharic words in a sentences would make segmental prediction more accurate
- Incorporate more prosodic modeling such as stress, intonation, etc
- Using part of speech tagger might make the prediction more accurate so, being involved in developing a mechanism to extract part of speech from text as one linguistic feature would improve the performance of the duration model.

REFERENCES

[1] Sami Lemmetty, “Review of Speech Synthesis Technology” , Master‟s Thesis, Helsinki University of Technology, 1999

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

[3] Nooteboom, Sieb. "The Prosody of Speech: Melody and Rhythm." The Handbook of Phonetic Sciences. Hardcastle, William J. and John Laver (eds). Blackwell Publishing, 1999.

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

[5] Dutoit, T. An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers, Dordrecht. 1997.

[6] Yamagishi, J., Kawai, H., Kobayashi, T., 2008. Phone duration modeling using gradient tree boosting. Speech Communication. 50(5), 405-415.

[7] David Crystal A Dictionary of Linguistics and Phonetics 6th Edition. © 2008 David Crystal. ISBN: 978-1-405-15296-9

[8] Silén, H., Helander, E., Nurminen, J., Gabbouj, M., 2010. Analysis of Duration Prediction Accuracy in HMM-Based Speech Synthesis. In Proc. of Speech Prosody, Chicago, USA.

[9] Lisker, L. Closure duration and the intervocalic voiced voiceless distinction in English . Language, (33) , 42 9.1957.

[10] Nakatani, L.H. and Schaffer, J.A,. Hearing “words” without words. Journal of the Acoustical Society of America , (63) , 234 45,1978.

[11] Eefting, W.Z.F.. Timing in talking. Tempo variation in production and its role in perception . Ph.D. thesis. Utrecht: Utrecht University,1991.

[12] Klatt, D.H.,. Linguistic uses of segmental duration in English: Acoustic and perceptual evidence. Journal of the Acoustic Society of America. 59, 1209-1221,1976.

[13] Van Santen, J.P.H.,. Contextual effects on vowel durations. Speech Communication. 11(6), 513-546, 1992.

[14] Lehiste, I.. Suprasegmentals . Cambridge, MA: MIT Press,1970.

[15] Chen, S.H., Hwang, S.H., Wang, Y.R.,. An RNN-based prosodic information synthesizer for Mandarin textto-speech. IEEE Trans. on Speech and Audio Processing. 6(3), 226-239,1998.

[16] Ferrer, L., Bratt, H., Gadde, V.R.R., Kajarekar, S.S., Shriberg, E., Sonmez, K., Stolcke, A., Venkataraman, A.,. Modeling duration patterns for speaker recognition. In Proc. of EUROSPEECH-2003, Geneva, Switzerland, pp. 2017-2020, 2003.

[17] Lazaridis, A., Zervas, P., Kokkinakis, G.,. Segmental duration modeling for Greek speech synthesis. In Proc.of IEEE ICTAI-2007, Patras, Greece, pp. 518-521, 2007.

[18] Takeda, K., Sagisaka, Y., Kuwabara, H.,. On sentence-level factors governing segmental duration in Japanese. Journal of the Acoustical Society of America 86(6), 2081-2086,1989.

[19] Riley, M.D., "Tree-based modeling for speech synthesis.," In: G. Bailly, C. Benoit, and T. Sawallis (Eds.), Talking machines: Theories, models and designs,, pp.265–273, 1992.

[20] Goubanova, O., Taylor, P.,. Using Bayesian belief networks for model duration in text-to-speech systems. In Proc. of ICSLP-2000, Beijing, China, pp. 427-430, 2000.

[21] Nirayo Hailu Gebregziabher "Modeling Improved Amharic Sylliblification Algorithm". MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia,2011.

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

[23] Nadew Tademe "Formant-based speech synthesis for Amharic vowels". MSc Thesis, Faculty of Informatics, Addis Ababa University, Ethiopia, 2008.

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

[25] Esheta Dereb "Phoneme Level Automatic Speech Segmentation for Amharic Language Using HMM Approach" MSc Thesis, Faculty of Computer Science and Mathematics, Addis Ababa University, Ethiopia, 2011.

[26] Lee, P. M. Bayesian Satistics. Arnold, Cambridge, 1997.

[27] Bishop, C. Neural Networks for Pattern Recognition. Clarendon Press, Cambridge, 1998.

[28] Bourlard, H., Hermansky, H., Morgan, N.,. Towards increasing speech recognition error rates. Speech Communication. 18(3), 205-231, 1996.

[29] Sadaf Nawaz, Duration Modelling For Urdu Using the Sum of Products Model , MSc National University of Computer and Emerging Sciences (NUCES), Lahore, in October 2005.

[30] Goubanova, O., King, S., Bayesian networks for phone duration prediction, Speech Communication, 2007.

[31] Sridhar Krishna, Hema A. Murthy., Duration modelilng of Indian languages Hindi and Telugu, Indian Institute of Technology, Madras, Chennai – 600036, 2004.

[32] Klatt, D.H. and Cooper, W.E.. Perception of segment duration in sentence contexts . In A. Cohen and S.G. Nooteboom (eds), Structure and process in speech perception (pp. 69 89). Berlin: Springer Verlag, 1975.

[33] House, A.S..On vowel duration in English. Journal of the Acoustical Society of America, (33), 11748, 1961.

[34] Fischer-Jørgensen, E.. Sound duration and place of articulation in Danish. Zeitschrift fur Sprachwissenschaft und Kommunikationsforschung , (17) , 175 207, 1964.

[35] Slis, I.H. and Cohen, A.. On the complex regulating the voiced-voiceless distinction, I and II .Language and Speech , (12) , 80 102 and 137 56, 1969a and b.

[36] L. S. Lee, C. Y. Tseng, and M. Ouh-Young, "The synthesis rules in a Chinese text-to-speech system," IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp. 1309–1320, 1989.

[37] Van Santen, J. P. H., and Sproat, R. Methods and tools. In Multilingual Text to Speech Synthesis, R. Sproat, Ed. Kluwer Academic Publishers, 1998.

[38] Dennis H. Klatt, Synthesis by rule of segmental durations in English sentences., In B. Lindblom and S. Ohman, editors, Frontiers of Speech Communication Research, pages 287-300., Academic Press, New York., 1979.

[39] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone,C. J. Classification and Regression Trees. Chapman Hall,New York, USA, 1984.

[40] Batusek, R., (2002), "A Duration Model for Czech Text-to-Speech Synthesis", in Proc. of Speech Prosody, France, 2002.

[41] Antonio Bonafonte, Ignasi Esquerra, Albert Febrer, José A. R. Fonollosa, The UPC text-to-speech system for Spanish and Catalan, in Proceedings of ICSLP, 1998.

[42] Ozlem Ozturk, Tolga Ciloglu , Segmental Duration Modeling in Turkish, INTERSPEECH – ICSLP, 2006.

[43] Tadesse Anberbir ,Hyunjung Shin, and Dong Yoon Kim Decision Tree Based Segmental Duration Prediction for Amharic TTS system ISCIT, 2009.

[44] Atelach Alemu Argaw and Lars Asker, " An Amharic Stemmer: Reducing Words to their Citation Forms", Proceedings of the 5th Workshop on Important Unresolved Matters, pages 104–110, Prague, Czech Republic, June 2007.

[45] Hussien Seid and Bj¨orn Gamb¨ack "A Speaker Independent Continuous Speech Recognizer for Amharic", INTERSPEECH, Lisbon Portugal, 2005

[46] Worku Alemu, "The Application of OCR Technique to the Amharic Script", Master's Thesis, Addis Ababa University, 1997.

[47] ጌታሁን አማረ ፣ "የአማርኛ ሰዋሰው በቀላል አቀራረብ" ፣ አዲስ አበባ ፣ 1989.

[48] Taylor, P., Caley, R., Black, A. and King, S., "Edinburgh Speech Tools Library", 1.2.1 edition, University of Edinburgh, 1999. Republic, September 2005.vol. 1, 7-10 May 1996.82

[49] Black A. and Lenzo K., “Optimal Data Selection for Unit Selection Synthesis”, The 4th ISCA Workshop on Speech Synthesis, Perthshire, Scotland, UK, August 29 - September 1, 2001.

[50] A.W. Black, P. Taylor and R. Caley, “The Festival speech synthesis system.”,1998.

[51] Alan W Black and Kevin A Lenzo. Building Synthetic Voices, For FestVox 2.0 Edition. Language Technologies Institute, Carnegie Mellon University and Cepstral, LLC, 2003b.

[52] John Kominek, Alan W Black “Impact of Durational Outlier Removal From Unit Selction Catalogs” pp 155-160, 5th ISCA Speech Synthesis Workshop, Pittsburgh, PA, 2004.

[53] A.W. Black and P. Taylor, “Automatically clustering similar units for unit selection in speech synthesis,” in Proceedings of EUROSPEECH, pp. 601–604, 1997.

[54] Campbell N. and Isard S. "Segment durations in a syllable frame", Journal of Phonetics, 19:1 37-47, 1991.

Appendix A: Amharic phonetic list, IPA equivalence and its transliteration table

IPA	Transcription	Amharic equivalence
Consonants		
[p]	[p]	ፕ
[t]	[t]	ት
[k]	[k]	ክ
[ʔ]	[ax]	ዕ
[b]	[b]	ብ
[d]	[d]	ድ
[g]	[g]	ግ
[pʼ]	[px]	ጵ
[tʼ]	[tx]	ጥ
[cʼ]	[cx]	ጭ
[q]	[q]	ቅ
[f]	[f]	ፍ
[s]	[s]	ስ
[ʃ]	[sx]	ሽ
[h]	[h]	ህ
[sʼ]	[xx]	ጽ
[tʃ]	[c]	ች
[gʼ]	[j]	ጅ
[m]	[m]	ም
[n]	[n]	ን
[nʼ]	[nx]	ኝ
[l]	[l]	ል
[r]	[r]	ር
[j]	[y]	ይ
[w]	[w]	ው
[v]	[v]	ቭ
[z]	[z]	ዝ
[zʼ]	[zx]	ዥ
Vowels		
[ɛ]	[e]	ኧ
[ʊ]	[u]	ኡ
[ɪ]	[ii]	ኢ
[ɑ]	[a]	ኣ
[e]	[ie]	ኤ
[ɨ]	[ix]	እ
[o]	[o]	ኦ

Appendix B: Sample labeled speech

```
#
0.1675000 26 pau
0.2575000 26 f
0.3400000 26 ie
0.4175000 26 d
0.4925000 26 ie
0.5275000 26 r
0.6050000 26 ie
0.6925000 26 sx
0.7475000 26 ix
0.7950000 26 n
0.8775000 26 u
0.9275000 26 b
0.9950000 26 a
1.0800000 26 xx
1.1350000 26 e
1.2350000 26 d
1.2900000 26 e
1.3800000 26 q
1.4675000 26 a
1.5850000 26 c
1.6225000 26 e
1.6650000 26 w
1.7100000 26 m
1.7750000 26 e
1.9175000 26 m
1.9650000 26 e
2.0150000 26 r
2.0525000 26 ii
2.0875000 26 y
2.1275000 26 a
2.1525000 26 w
2.2225000 26 o
2.3150000 26 c
2.3900000 26 k
2.4375000 26 e
2.5000000 26 b
2.5925000 26 a
2.6350000 26 l
2.6850000 26 e
2.7375000 26 d
2.7850000 26 ix
2.8225000 26 r
2.8625000 26 ix
2.8925000 26 sx
3.0175000 26 a
3.1575000 26 a
3.2475000 26 k
3.3275000 26 a
```

```
3.3725000 26 l
3.4575000 26 a
3.5250000 26 t
3.7175000 26 g
3.8075000 26 a
3.8850000 26 r
3.9575000 26 t
3.9975000 26 e
4.0500000 26 w
4.1150000 26 e
4.1650000 26 y
4.2300000 26 a
4.2825000 26 y
4.4050000 26 e
4.4975000 26 pau
```

Appendix C: Amharic phoneset with their features

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

Appendix D: Portion of Unit Catalog generated at the end of Amharic Unit Selection Voice

```
EST_File index
DataType ascii
NumEntries 52500
IndexName AAU33_Amharic33_Esheta33
EST_Header_End
pau_1924 Sentence1 0.000000 0.070000 0.140000
y_2168 Sentence1 0.140000 0.175000 0.210000
e_8303 Sentence1 0.210000 0.225000 0.240000
n_2404 Sentence1 0.240000 0.265000 0.290000
e_8302 Sentence1 0.290000 0.325000 0.360000
q_722 Sentence1 0.360000 0.410000 0.460000
e_8301 Sentence1 0.460000 0.485000 0.510000
m_2218 Sentence1 0.510000 0.545000 0.580000
ix_5345 Sentence1 0.580000 0.597500 0.615000
t_2684 Sentence1 0.615000 0.632500 0.650000
ie_751 Sentence1 0.650000 0.690000 0.730000
s_1668 Sentence1 0.730000 0.795000 0.860000
ix_5344 Sentence1 0.860000 0.875000 0.890000
t_2683 Sentence1 0.890000 0.905000 0.920000
a_5404 Sentence1 0.920000 0.950000 0.980000
d_1219 Sentence1 0.980000 1.005000 1.030000
ii_1194 Sentence1 1.030000 1.045000 1.060000
y_2167 Sentence1 1.060000 1.075000 1.090000
o_1189 Sentence1 1.090000 1.120000 1.150000
m_2217 Sentence1 1.150000 1.190000 1.230000
g_1115 Sentence1 1.230000 1.255000 1.280000
ix_5343 Sentence1 1.280000 1.295000 1.310000
n_2403 Sentence1 1.310000 1.345000 1.380000
ix_5342 Sentence1 1.380000 1.392500 1.405000
b_2106 Sentence1 1.405000 1.417500 1.430000
a_5403 Sentence1 1.430000 1.475000 1.520000
t_2682 Sentence1 1.520000 1.550000 1.580000
a_5402 Sentence1 1.580000 1.625000 1.670000
s_1667 Sentence1 1.670000 1.715000 1.760000
ix_5341 Sentence1 1.760000 1.780000 1.800000
l_2152 Sentence1 1.800000 1.815000 1.830000
ix_5340 Sentence1 1.830000 1.850000 1.870000
s_1666 Sentence1 1.870000 1.890000 1.910000
a_5401 Sentence1 1.910000 1.960000 2.010000
b_2105 Sentence1 2.010000 2.040000 2.070000
e_8300 Sentence1 2.070000 2.100000 2.130000
m_2216 Sentence1 2.130000 2.155000 2.180000
e_8299 Sentence1 2.180000 2.210000 2.240000
t_2681 Sentence1 2.240000 2.270000 2.300000
o_1188 Sentence1 2.300000 2.335000 2.370000
t_2680 Sentence1 2.370000 2.400000 2.430000
e_8298 Sentence1 2.430000 2.460000 2.490000
tx_650 Sentence1 2.490000 2.515000 2.540000
```

```
e_8297 Sentence1 2.540000 2.570000 2.600000
n_2402 Sentence1 2.600000 2.615000 2.630000
a_5400 Sentence1 2.630000 2.675000 2.720000
q_721 Sentence1 2.720000 2.775000 2.830000
e_8296 Sentence1 2.830000 2.855000 2.880000
q_720 Sentence1 2.880000 2.925000 2.970000
e_8295 Sentence1 2.970000 3.020000 3.070000
pau_1923 Sentence1 3.070000 3.161250 3.252500
pau_1922 Sentence2 0.000000 0.077500 0.155000
a_5399 Sentence2 0.155000 0.205000 0.255000
xx_147 Sentence2 0.255000 0.305000 0.355000
e_8294 Sentence2 0.355000 0.391250 0.427500
d_1218 Sentence2 0.427500 0.457500 0.487500
e_8293 Sentence2 0.487500 0.522500 0.557500
b_2104 Sentence2 0.557500 0.593750 0.630000
e_8292 Sentence2 0.630000 0.658750 0.687500
d_1217 Sentence2 0.687500 0.727500 0.767500
u_1037 Sentence2 0.767500 0.796250 0.825000
b_2103 Sentence2 0.825000 0.845000 0.865000
a_5398 Sentence2 0.865000 0.885000 0.905000
y_2166 Sentence2 0.905000 0.948750 0.992500
m_2215 Sentence2 0.992500 1.012500 1.032500
a_5397 Sentence2 1.032500 1.078750 1.125000
r_2007 Sentence2 1.125000 1.141250 1.157500
a_5396 Sentence2 1.157500 1.200000 1.242500
t_2679 Sentence2 1.242500 1.292500 1.342500
o_1187 Sentence2 1.342500 1.368750 1.395000
n_2401 Sentence2 1.395000 1.417500 1.440000
```

Appendix E : Sample Feature vectors

-1.23084	t	ix	ie	1	1	1	1	final	5	0	1	0	4	coda	onset	coda
-0.02032	m	o	g	1	2	1	1	final	4	2	0	1	3	coda	coda	onset
-0.14893	ix	s	l	1	1	1	1	initial	3	1	0	1	0	onset	coda	onset
-0.54846	o	t	t	1	1	1	1	final	3	1	0	1	2	onset	coda	onset
1.174426	e	q	pau	1	1	0	1	final	5	1	0	1	4	onset	coda	coda
-0.86223	b	u	a	1	2	1	1	final	3	0	1	0	2	coda	onset	coda
0.520894	l	n	e	1	1	1	2	initial	2	0	1	0	0	coda	onset	coda
-0.22746	e	tx	b	1	1	1	1	mid	5	1	0	1	2	onset	coda	onset
-0.91717	ie	l	t	1	1	1	1	mid	4	1	0	1	2	onset	coda	onset
0.385864	e	n	c	1	2	0	1	single	1	1	0	0	0	onset	coda	coda
-0.66308	ix	px	y	1	1	1	1	mid	6	1	0	1	4	onset	coda	onset
-0.79423	o	b	l	1	2	1	1	final	4	1	0	0	3	onset	coda	coda
-0.37387	ix	n	d	1	1	1	1	mid	6	1	0	1	1	onset	coda	onset
-0.95371	n	ix	ix	1	1	1	1	mid	4	0	1	0	1	coda	onset	coda
-0.31508	e	d	g	1	1	1	1	mid	5	1	0	1	1	onset	coda	onset
0.103179	y	a	ix	1	1	1	1	mid	3	0	1	0	1	coda	onset	coda
-0.52165	y	ii	o	1	1	1	1	final	4	0	1	0	3	coda	onset	coda
-0.00906	sx	a	e	1	1	1	1	mid	4	0	1	0	1	coda	onset	coda
-0.70684	b	ix	a	1	1	1	1	final	3	0	1	0	2	coda	onset	coda
-0.74779	n	e	m	1	2	1	1	final	4	2	0	1	3	coda	coda	onset
-0.36993	d	e	ix	1	2	1	1	final	2	0	1	0	1	coda	onset	coda
-0.08134	a	q	l	1	1	1	1	mid	5	1	0	1	3	onset	coda	onset
0.127306	r	ie	a	1	2	1	1	final	4	0	1	0	3	coda	onset	coda
-0.13175	a	y	b	1	1	1	1	final	4	1	0	1	3	onset	coda	onset
-0.54969	r	o	t	1	3	1	1	final	2	2	0	0	1	coda	coda	coda
-0.41899	s	ii	y	1	2	1	1	final	2	2	0	1	1	coda	coda	onset
-0.01442	k	ix	ii	1	1	1	1	mid	4	0	1	0	2	coda	onset	coda
-1.25924	ie	t	m	1	1	1	1	final	3	1	0	1	2	onset	coda	onset
-0.75948	ix	y	l	1	1	1	1	mid	3	1	0	1	1	onset	coda	onset
0.210633	e	c	s	1	1	1	1	mid	6	1	0	1	3	onset	coda	onset
0.26502	d	o	a	1	1	1	1	mid	4	0	1	0	2	coda	onset	coda
-0.43867	d	e	ix	1	2	1	1	final	2	0	1	0	1	coda	onset	coda
-0.08849	a	q	l	1	2	0	1	final	4	1	0	0	3	onset	coda	coda
-0.81643	n	ix	u	1	1	1	1	final	5	0	1	0	4	coda	onset	coda
-0.15719	c	a	e	1	2	1	1	final	5	0	1	0	4	coda	onset	coda
-0.15466	a	y	w	1	1	1	1	mid	5	1	0	1	3	onset	coda	onset
-0.5465	d	e	ix	1	1	1	1	mid	6	0	1	0	3	coda	onset	coda
-0.07217	a	l	t	1	2	1	1	final	3	1	0	0	2	onset	coda	coda
-0.10883	a	y	y	1	1	1	1	mid	4	1	0	1	2	onset	coda	onset
0.529001	e	b	n	1	1	1	1	initial	4	1	0	1	0	onset	coda	onset
0.030255	ix	g	m	1	1	1	3	initial	2	1	0	1	0	onset	coda	onset

Appendix F : Portion of Amharic Duration Model

```
((p.name is ix)
 ((R:SylStructure.parent.pos_in_word < 1.4)
  ((n.name is l)
   ((1.39107 1.9549))
   ((pp.ph_cplace is l)
    ((n.ph_vheight is 2)
     ((n.name is ie)
      ((0.36093 -0.728674))
      ((0.705228 -0.4261)))
     ((lisp_coda_fric is 0)
      ((R:SylStructure.parent.position_type is final)
       ((nn.ph_ctype is l)
        ((0.628016 0.137413))
        ((0.672087 -0.307034)))
       ((nn.ph_cvox is +)
        ((ph_ctype is s)
         ((0.454785 0.0295566))
         ((lisp_onset_nasal is 0)
          ((0.928185 0.560237))
          ((0.632242 0.359048))))
        ((n.ph_cvox is +)
         ((0.685476 0.263189))
         ((0.653351 -0.00484945)))))
       ((0.873571 -0.489054))))
    ((ph_cplace is l)
     ((pp.ph_cplace is v)
      ((0.405323 -0.395409))
      ((nn.ph_ctype is f)
       ((0.527719 -0.288309))
       ((nn.ph_cvox is -)
        ((0.719222 0.261038))
        ((pp.ph_cvox is +)
         ((0.369123 -0.184948))
         ((0.586398 0.0335649))))))
     ((lisp_onset_glide is 0)
      ((pp.ph_ctype is s)
       ((syl_final is 0)
        ((R:SylStructure.parent.R:Syllable.nn.accented is 0)
         ((pp.ph_cplace is a)
          ((R:SylStructure.parent.syl_in < 16.5)
           ((0.759466 -0.0887153))
           ((0.675885 0.200855)))
          ((pp.ph_cvox is +)
           ((0.659718 -0.33879))
           ((0.92476 -0.385554))))
         ((1.10265 0.630513)))
        ((nn.ph_vheight is 3)
         ((0.636525 0.0377173))
         ((nn.ph_vheight is 2)
```

```
    ((1.38795 0.378824))
    ((1.47014 1.17929)))))
 ((n.ph_vfront is 3)
  ((0.718628 0.0226528))
  ((name is q)
   ((0.922994 1.29449))
   ((R:SylStructure.parent.last_accent < 0.6)
    ((R:SylStructure.parent.R:Syllable.nn.accented is 0)
     ((ph_cvox is +)
      ((R:SylStructure.parent.parent.word_numsyls < 4.2)
       ((ph_ctype is l)
        ((0.980475 0.222767))
        ((nn.ph_cplace is a)
         ((1.38983 1.15192))
         ((0.949473 0.745682))))
       ((0.826928 0.290098)))
      ((ph_cplace is a)
       ((1.13425 0.540668))
       ((0.460739 0.0608113))))
     ((nn.ph_cvox is +)
      ((pp.ph_vheight is 0)
       ((1.18947 0.841438))
       ((0.896516 0.395254)))
      ((1.04284 1.04957))))
    ((1.35487 1.27332))))))
((pp.ph_ctype is f)
 ((nn.ph_ctype is f)
  ((0.246566 -0.756484))
  ((R:SylStructure.parent.position_type is final)
   ((0.702855 -0.310396))
   ((1.01905 0.337393))))
 ((n.name is ix)
  ((1.34657 0.90925))
  ((ph_cplace is a)
   ((pp.ph_ctype is s)
    ((0.545013 -0.280189))
    ((1.05055 0.0114965)))
   ((pp.ph_cvox is +)
    ((0.865499 0.509979))
    ((0.525396 0.1755)))))))))
```

Appendix G: Source code of prototype for perceptual evaluation

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <fstream>
#include <string>
#include <ctype.h>
#include <sstream>
#include <cstring>
using namespace std;
class AmharicTTSDurationModeTest {

  public:
   void play(int);
   int getScore(int);

};

void play(int w)
{
string waveFile;
stringstream mystr;
mystr<<w;
waveFile = "play Amharic_Test_Data_40/"+mystr.str() + ".wav";
char* audioFile= (char*)malloc(sizeof(char)*(waveFile.length() + 1));
strcpy(audioFile,waveFile.c_str());
system(audioFile);
}
```

```
int getScore(int ref)
{
  int rate;
  cout<<"PLEASE GIVE RANK FOR THE SPEECH YOU LISTENED AND HIT ENTER
KEY";
  cin>>rate;
  return rate;
}

int main()
{
  cout<<"-----------------------------------------------------------------------------------------------
"<<endl<<endl;
  cout<<"-------WEL COME TO EVALUATE THE QUALITY(NATURALNESS) OF
AMHARIC TEXT TO SPEECH SYSTEM---------"<<endl<<endl;
  cout<<"NATURALNESS measure to what extent that synthesized speech looks like
human speech."<<endl<<endl;
  cout<<"Listen to the speech and rank for each one based on the following perceptual
evaluation catagories:"<<endl<<endl;

              cout<<"5 for EXCELLENT"<<endl;
              cout<<"4 for VERY GOOD"<<endl;
              cout<<"3 for GOOD"<<endl;
              cout<<"2 for FAIR"<<endl;
              cout<<"1 for POOR"<<endl;
              cout<<"0 for VERY POOR"<<endl<<endl;


  int start;
  cout<<"PLEASE ENTER 1 TO START YOUR EVALUATION AND HIT ENTER KEY";
  cin>>start;
```

```
 if(start != 1)
 {
   return 1;
 }

 AmharicTTSDurationModeTest t;
int wavReferences[100] =
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3
2,33,34,35,36,37,38,39,40};
 int number = 0;
 int Score [100];
 int scr1,scr2;
 ofstream sent ("SentencesSequences_Naturalness",ios::app);

 for(int i = 0 ; i <= 19 ; i++)
 {
      number = rand() % 2;

      if(number == 0)
      {
            sent<<wavReferences[i]<<"And"<<wavReferences[i + 20]<<"\n";
            play(wavReferences[i]);
            scr1 = getScore(wavReferences[i]);
            play(wavReferences[i + 20]);
            scr2 = getScore(wavReferences[i + 20]);

      }
      else
      {
            sent<<wavReferences[i + 20]<<"And"<<wavReferences[i]<<"\n";
            play(wavReferences[i + 20]);
```

```
            scr2 = getScore(wavReferences[i + 20]);
            play(wavReferences[i]);
            scr1 = getScore(wavReferences[i]);

        }

        Score[wavReferences[i]] = scr1;
        Score[wavReferences[i + 20]] = scr2;
    }
  cout<<endl<<endl<<endl;
  cout<<"-------------------------------------------------------------------------------------
"<<endl<<endl;
  cout<<"YOU HAVE COMPLETED THE NATURALNESS TEST FOR AMHARIC TEXT
TO SPEECH SYSTEM SUCCESSFULLY!!!"<<endl<<endl;
  cout<<"----------------------THANK YOU FOR YOUR COOPORATION!!!----------------------
------------"<<endl<<endl;
  cout<<"-------------------------------------------------------------------------------------
"<<endl<<endl;
        ofstream myfile ("Amharic_EvaluationResult_Naturalness",ios::app);
        myfile<<"----------------New Listener----------------------------"<<"\n";
        if(myfile.is_open())
        {
          for( int j = 1 ; j <= 40 ; j++)
          {
              myfile<<j<<": "<<Score[j]<<"\n";
          }
          myfile.close();
        }
else cout << "Unable to open file";
return 0;
}
```

Declaration

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

Declared by:

Yonas Demeke WoldeMariam **Date:**

_____

Signature:

Confirmed by advisor:

Dr.Sebsibie HaileMariam **Date:**

_____

Signature:

Place and date of submission: Addis Ababa, March 2012.