A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

by

ESRA VURAL

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2003

A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

APPROVED BY

Kemal OFLAZER .....
(Thesis Supervisor)

Hakan ERDOĞAN .....

Yücel SAYGIN .....

DATE OF APPROVAL: .....18/07/2003.....

to My Family

Acknowledgments

I am thankful to my thesis supersivor Kemal Oflazer for introducing me to this challenging subject. His knowladge, patience and insight was very helpful in advancing through the project. I am also grateful to Hakan Erdoğan for his motivations and criticisms during the preparation of the thesis. I would also like to thank to Yücel Saygın for serving on my thesis committee and for his revisions on the thesis.

I would like to thank to my parents and sister for their love, support and encouragement. Lastly, I am thankful to the patience and support of Aydın Akyol for enabling us to have his voice recordings.

A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

Abstract

Naturalness in Text-to-Speech systems is very important in achieving high quality waveform. The naturalness of the waveform is highly correlated with phonetic coverage and prosodic features such as, duration and F0 contour. Duration determines the timing for the synthesized phoneme, whereas F0 contour determines fundamental frequency component of the waveform.

This thesis presents the development of a prosodic Text-to-Speech System for Turkish Language using the Festival Tool [31]. We describe a complete realization of a new male voice, covering allophones of Turkish using duration and F0 parameters. The duration of the allophones and the word stress have been studied extensively. Sentence stress and phrasal stress are also discussed by in less detail.

Carrier words are designed approximately for all allophone-allophone combinations. 1680 carrier words are recorded in a sound-proof recording studio. LPC (linear predictive coding) and RES (residual) parameters are computed. The text normalisation module is implemented for abbreviations and numbers. Durations for the allophones are entered. Sentence level and word level F0 generation modules are implemented. By increasing the number of phonemes and giving prosody we obtained a more natural sounding Text-to-Speech System for Turkish Language.

TÜRKÇE İÇİN VURGULU METİNDEN SES SENTEZLEYİCİSİ

Özet

Metinden ses sentezleyicisi sistemlerinde doğallık kaliteli bir ses dalgası elde edilmesinde çok önemli bir rol oynar. Ses dalgasının doğallığı fonetik kapsama ve vurgusal özellikler olan perde frekans eğrisi ve süre bilgileriyle ilişkilidir. Süre bilgisi sentezlenen fonemin zaman bilgisini belirler, perde frekans eğrisi ise ses dalgasının temel frekans özelliklerini kapsar.

Bu tezde, Festival ses sentezleme sistemi kullanılarak, Türkçe için vurgulu metinden ses sentezleyicisi geliştirilmiştir [31]. Yeni bir erkek sesi, Türkçedeki alofonları kapsayarak, temel frekans ve süre bilgileri kullanılarak oluşturulmuştur. Alofonların süresi ve kelime vurgusu geniş çapta çalışılmıştır. Cümle vurgusu ve kelime öbek vurgusu daha az detaylı olarak çalışılmıştır.

Tüm alofon kombinasyonları için taşıyıcı kelimeler oluşturulmuştur. 1680 tane taşıyıcı kelime ses yalıtımlı bir kayıt stüdyosunda kaydedilmiştir. LPC ve RES parametreleri hesaplanmıştır. Kısaltmalar ve sayılar için metni normalize eden bir modül geliştirilmiştir. Alofonlar için süre bilgisi girilmiştir. Cümle ve kelime seviyelerinde F0 üretim modülleri geliştirilmiştir. Fonem sayısını arttırarak ve vurgu yaratarak Türkçe için daha doğal bir metinden ses sentezleyici sistem elde edilmiştir.

Table of Contents

List of Figures

List of Tables

A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

by

ESRA VURAL

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2003

A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

APPROVED BY

Kemal OFLAZER .....
(Thesis Supervisor)

Hakan ERDOĞAN .....

Yücel SAYGIN .....

DATE OF APPROVAL: .....18/07/2003.....

to My Family

Acknowledgments

I am thankful to my thesis supersivor Kemal Oflazer for introducing me to this challenging subject. His knowladge, patience and insight was very helpful in advancing through the project. I am also grateful to Hakan Erdoğan for his motivations and criticisms during the preparation of the thesis. I would also like to thank to Yücel Saygın for serving on my thesis committee and for his revisions on the thesis.

I would like to thank to my parents and sister for their love, support and encouragement. Lastly, I am thankful to the patience and support of Aydın Akyol for enabling us to have his voice recordings.

A PROSODIC TURKISH TEXT-TO-SPEECH SYNTHESIZER

Abstract

Naturalness in Text-to-Speech systems is very important in achieving high quality waveform. The naturalness of the waveform is highly correlated with phonetic coverage and prosodic features such as, duration and F0 contour. Duration determines the timing for the synthesized phoneme, whereas F0 contour determines fundamental frequency component of the waveform.

This thesis presents the development of a prosodic Text-to-Speech System for Turkish Language using the Festival Tool [31]. We describe a complete realization of a new male voice, covering allophones of Turkish using duration and F0 parameters. The duration of the allophones and the word stress have been studied extensively. Sentence stress and phrasal stress are also discussed by in less detail.

Carrier words are designed approximately for all allophone-allophone combinations. 1680 carrier words are recorded in a sound-proof recording studio. LPC (linear predictive coding) and RES (residual) parameters are computed. The text normalisation module is implemented for abbreviations and numbers. Durations for the allophones are entered. Sentence level and word level F0 generation modules are implemented. By increasing the number of phonemes and giving prosody we obtained a more natural sounding Text-to-Speech System for Turkish Language.

TÜRKÇE İÇİN VURGULU METİNDEN SES SENTEZLEYİCİSİ

Özet

Metinden ses sentezleyicisi sistemlerinde doğallık kaliteli bir ses dalgası elde edilmesinde çok önemli bir rol oynar. Ses dalgasının doğallığı fonetik kapsama ve vurgusal özellikler olan perde frekans eğrisi ve süre bilgileriyle ilişkilidir. Süre bilgisi sentezlenen fonemin zaman bilgisini belirler, perde frekans eğrisi ise ses dalgasının temel frekans özelliklerini kapsar.

Bu tezde, Festival ses sentezleme sistemi kullanılarak, Türkçe için vurgulu metinden ses sentezleyicisi geliştirilmiştir [31]. Yeni bir erkek sesi, Türkçedeki alofonları kapsayarak, temel frekans ve süre bilgileri kullanılarak oluşturulmuştur. Alofonların süresi ve kelime vurgusu geniş çapta çalışılmıştır. Cümle vurgusu ve kelime öbek vurgusu daha az detaylı olarak çalışılmıştır.

Tüm alofon kombinasyonları için taşıyıcı kelimeler oluşturulmuştur. 1680 tane taşıyıcı kelime ses yalıtımlı bir kayıt stüdyosunda kaydedilmiştir. LPC ve RES parametreleri hesaplanmıştır. Kısaltmalar ve sayılar için metni normalize eden bir modül geliştirilmiştir. Alofonlar için süre bilgisi girilmiştir. Cümle ve kelime seviyelerinde F0 üretim modülleri geliştirilmiştir. Fonem sayısını arttırarak ve vurgu yaratarak Türkçe için daha doğal bir metinden ses sentezleyici sistem elde edilmiştir.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction To Speech Synthesis

Speech is the one of the most effective means of communication between people. Synthesizing is the process of generating speech waveforms using machines based on the phonetical transcription of the message. Recent progress in speech synthesis has produced synthesizers with very high intelligibility but the sound quality and naturalness still remain a major problem.

1.2 Text-to-Speech Systems

A Text-to-Speech (TTS) System is a system that converts any form of computerized text into speech waveforms [1]. In other words, a Text-to-Speech system emulates a real human speaker that can read any text aloud.

1.3 Prosodic Turkish TTS Synthesizer

This thesis presents the implementation of a prosodic Turkish TTS Synthesizer. The prosody is generated by defining duration and intonation modules considering the language specific properties of Turkish. The format of a Turkish Lexicon is described for the transcription of ortographic form into phonetic representation. Letter-to-Sound module is used for the words that are not present in the lexicon.

1.4 Fundamental Differences Between TTS Systems and Other Talking Machines

The fundamental difference between TTS systems and other talking machines like casette-players lies under the fact that TTS systems generate all the words synthetically whereas other talking machines may concatenate pre-recorded words or sentences and generate speech. In this respect, TTS systems may generate any text input into speech waveform easily. So TTS systems are very important in tasks that require large vocabulary. TTS systems use grapheme to phoneme conversion for the automatic generation of speech.

1.5 Application Areas Of TTS Systems

In the mid-eighties, as a result of the improved techniques in natural language processing and signal processing, the concept of high quality TTS appeared. TTS systems became useful in many commercial and personal usage.
Here are some examples of high quality TTS applications:

- **Telecommunication Services:** TTS systems make it easy to access textual information over the telephone. Texts may range from simple messages consisting from a small corpus, to huge complex messages that can never be stored as speech waveforms. Queries to information retrieval systems can be made through speech (with the help of a speech recognizer) and the answer (synthesized text) is returned back to the caller. AT&T has developed TTS systems for applications such as integrated messaging and telephone relay service.

  1) Integrated Messaging (electronic mail or fascimile reader application), which is a very useful application when one is away from the office.

  2) Telephone Relay Service, an application that is used for having a telephone conversation with a hearing or speech impaired person. These applications were acceptable although the sythesized voice was not sounding very natural.

- **Language Education:** High quality synthesis can be combined together with a Computer Aided Education for learning a new language in an interactive way.

- **Aid to Handicapped People:** Astro-physycist Stephen Hawking, gives his lectures and presentations using TTS technology. A special keyboard that is designed for handicapped people and a synthesizer can be coupled to help the voice handicapped. Blind people may also use TTS systems when coupled with an OCR (Optical Character Recognition) System, this can make possible access to any written text.

- **Talking Books and Toys:** Talking books and toys are not new to market, but the quality of the synthesis becomes more important in the entertainment business.

- **Vocal Monitoring:** In some situations, oral information may be more appropriate than written information. TTS systems can be used in measurement and control systems.

- **Multimedia and Man-Machine Communication:** The interaction between man and the machines will be indispensable in the near future so TTS systems must be able to produce highly qualified speech synthesis.

- **Fundamental and Applied Research:** TTS systems are excellent tools for linguistic research since they have constant, stable features, in other words repeated experience provides identical results. Even the people can change their utterances. So TTS systems are excellent laboratory tools for intonative and rhythmic models.

1.6 Review of Previous Work

1.6.1 Current Work on TTS Systems

There are many TTS systems for various languages. In the past synthesizers were built only considering a single language. Thus converting this language specific synthesizer for another language was a tough job. Reinventing the wheel for every new languge was a waste of time and effort. Instead, research labs decided to invest their effort in developing multilingual synthesizers. Their aim is developing TTS synthesizers for as many languages, dialects and voices as possible. In this section, we review some of these multilingual TTS projects.

MBROLA Project was initiated by TCTS Laboratory in the Faculte Polytechnique de Mons, Belgium. The aim of this project is to develop multilingual speech synthesis for non-commercial purposes and increase the academic research, especially in prosody generation. MBROLA method is similar to PSOLA method. But it is named MBROLA since PSOLA is a trademark of CNET. The MBROLA-material is available free for non-commercial and non-military purposes [2]. The MBROLA synthesizer is based on diphone concatenation. It takes a list of phonemes with some prosodic information namely duration and pitch as input. The input data required by MBROLA contains a phoneme name, a duration in milliseconds, and a series of pitch pattern points which are two integers. For instance, the input "_ 80 30 130" describes that the synthesizer will produce a silence of 80 ms, and will put a pitch pattern points of 130 Hz at 30% of 80 ms. MBROLA produces speech waveforms of 16 bits at the sampling frequency of 16 kHz. It is not a TTS system since it does not accept raw text as input but it may be used as a low level synthesizer. The diphone databases are currently available for American/British English, Brazilian Portugese, Dutch, French, German, Romanian, Spanish and Turkish with male and female voices.

Festival TTS system was developed in CSTR at the University of Edinburgh by Alan Black and Paul Taylor. The basic architecture of Festival was influenced from ATR's CHATR system. The system is written in C++. Festival TTS Synthesis system is available for American and British English, Spanish and Welsh. This system supports residual excited LPC and PSOLA methods and MBROLA database. The system is available free for educational, research and individual use. The system is developed for three main user groups, those who want to use the system for arbitrary TTS, for those who are developing language systems and finally for those who are developing new synthesis methods.

The Universite de Provence is coordinator of the MULTEXT [3] series of projects. The aim is developing tools and corpora and linguistic resources for a variety of languages.

1.6.2 Turkish TTS Systems

Bozkurt [4] describes a synthesizer for Turkish using the MBROLA technique. MBROLA technique attempts to overcome phase mismatches. As the pitch cycles have been pre-processed they have a fixed phase. It uses the PSOLA method for modifying the prosody. Spectral smoothing can be done by directly interpolating the pitch cycles in the time domain.

Another TTS project for Turkish has been developed by Levent Arslan from Bosphorous University and his team. They used LPC modelling for the parameterization of the speech waveform. This TTS System assumes that the context of the current phoneme is effected from the most closest two phonemes both on the left and right. The synthesis module searches for the most appropriate lpc file in the database, extracts LPC parameters from that file and uses duration, F0 contour with these parameters to synthesize frames. Durations are determined using two types of information. Half of the duration information comes from the model and the other half comes from actual waveforms. Prosody module finds the most likely pitch contour of synthesis. For voiced regions, the prosody module calculates a non-zero period value which is used in generating an impulse train. For unvoiced regions white noise spectrum is used as an approximation to the excitation signal.

Chapter 2

Text-to-Speech

In this chapter, we will describe the phases of a Text-to-Speech system. The components that will be implemented for a TTS system will be explained in detail.

2.1 Stages Of TTS Conversion

The TTS synthesis process consists of two main phases as shown in Figure 2.1. The

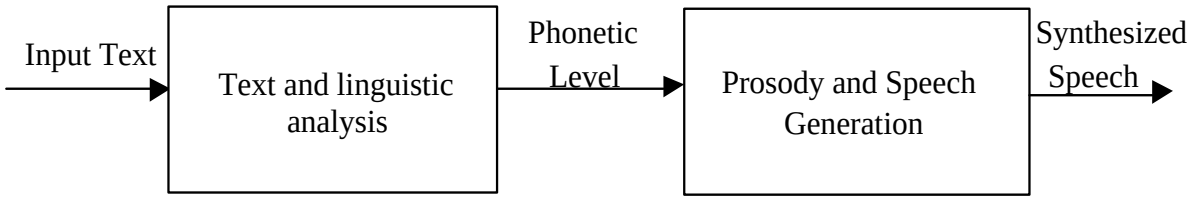

Figure 2.1: Simple TTS Synthesis Procedure

first phase is text analysis, where the input text is transcribed into some linguistic and phonetic representation. This phonetic representation usually includes the phonemes, duration, stress and intonation. The second phase is prosody and speech generation where the linguistic information is used to produce the correct speech waveform. The first part is implemented by a Natural Language Processing (NLP) module that is capable of producing a phonetic transcription of the given text, together with the intonation. The second part is implemented by a Digital Signal Processing (DSP) module, that transforms the phonetic and prosodic information into speech.

The basic TTS components are shown in Figure 2.2 [37]. The text analysis component normalizes text. The phonetic analysis component converts the processed text into the corresponding phoneme sequence. Prosodic analysis component

attaches appropriate pitch and duration information to the phoneme sequence. Finally, the speech synthesis component takes the parameters from the fully tagged phonetic sequence to generate the corresponding waveform. These modules will be described in detail later.

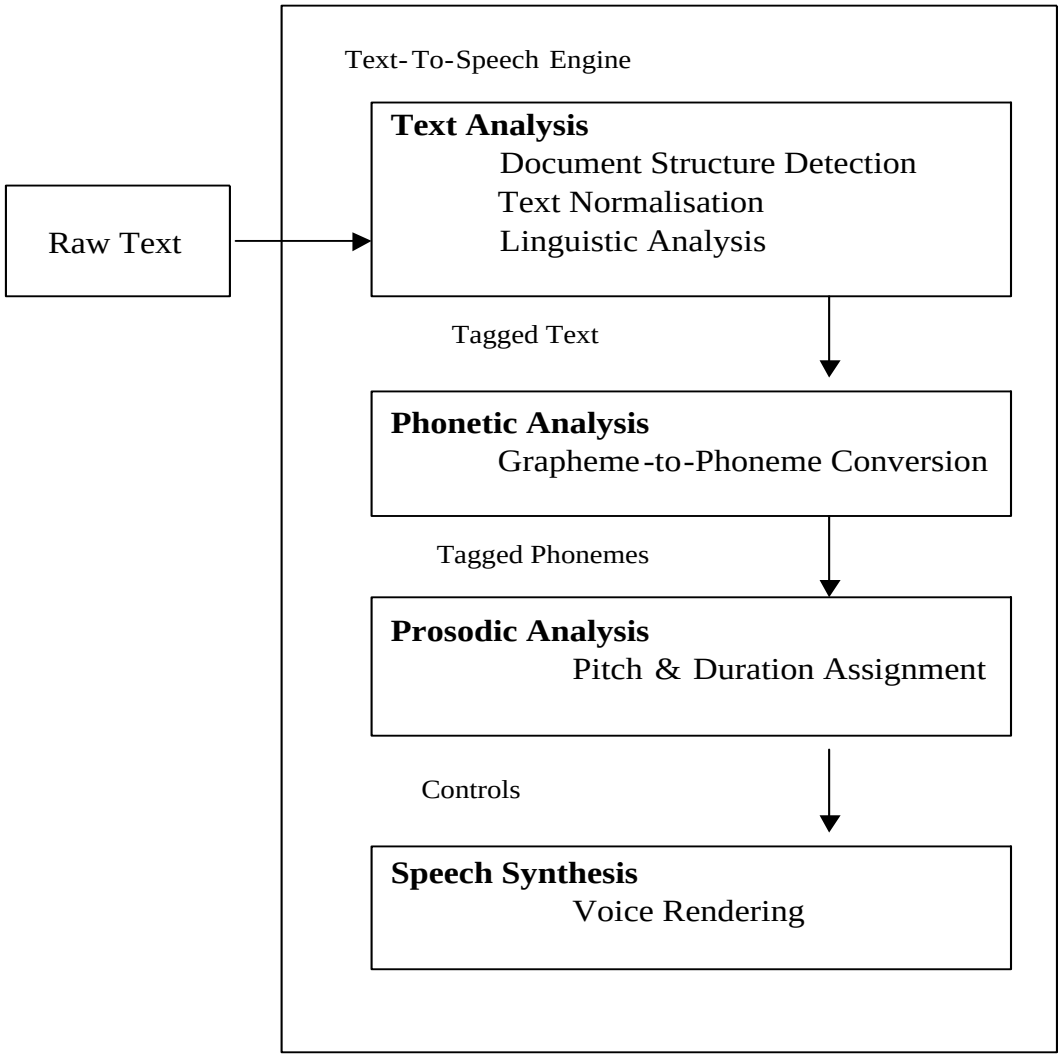

Figure 2.2: Basic System Architecture of a TTS system

2.2 The Natural Language Processing Component

Figure 2.3 [37] shows the skeleton of a general natural language processing module for TTS systems. The module consists of text analysis and phonetic analysis components.

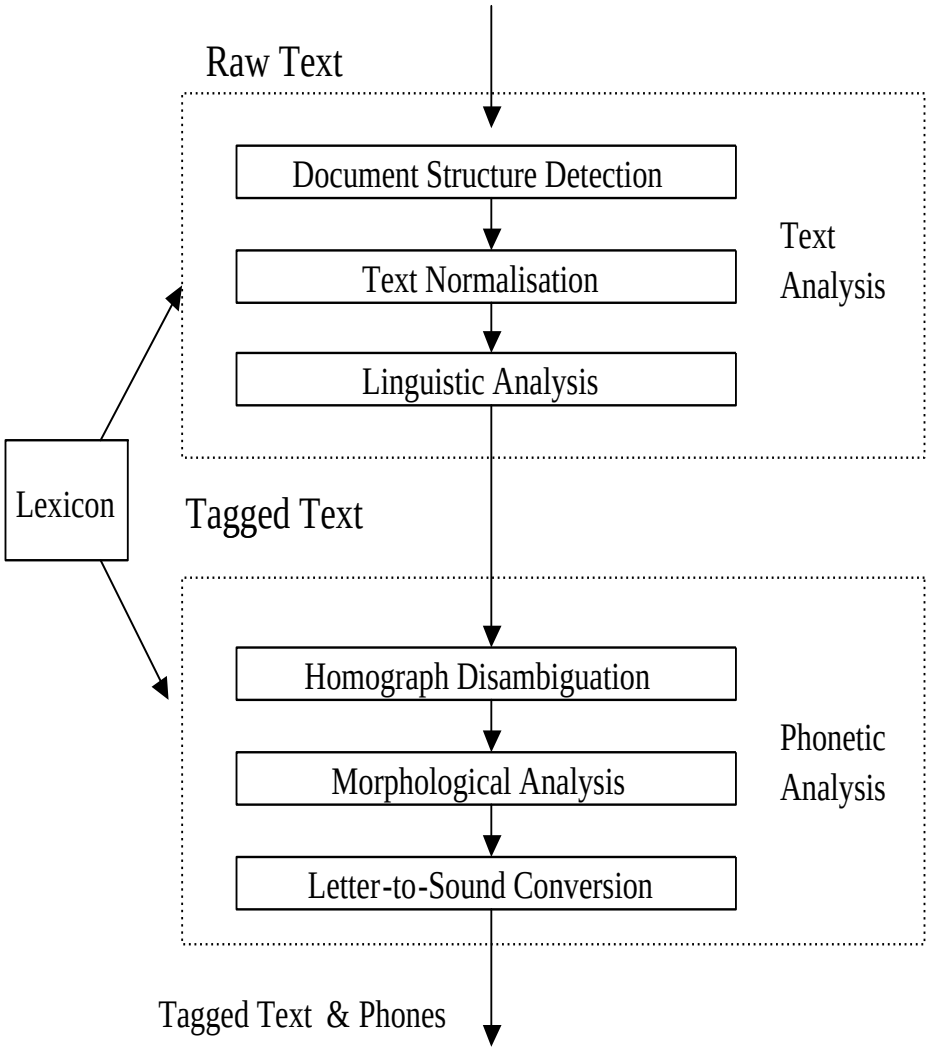

Figure 2.3: Natural Language Processing module of a general TTS Conversion System

2.2.1 Text Analysis

The text analysis component is responsible for determining document structure, conversion of non-ortographic symbols, and identification of language structure and meaning. Simple TTS systems just convert the nonortographic items such as numbers, into words. More ambitious TTS systems attempt to analyze white spaces and punctuations in order to determine both document structure and syntactic and semantic structure of the text. The syntactic and semantic structure enables the system to determine the phonetic and prosodic representation. The modules of the text analysis component will be described below.

Document Structure Detection Module:

The document structure detection module detects the layout of the document i.e sentence starts, paragraph starts. Document Structure effects the prosody. For instance, the beginning of a sentence must be detected in order to attach a different prosodic pattern. Likewise, a paragraph start should be identified so that a suitable prosodic pattern can be given.

Text Normalisation Module:

Text normalisation is the conversion of abbreviations, numbers, symbols and other non-ortographic entities of text into a common ortographic transcription that is suitable for subsequent phonetic conversion. It is the process of generating the ortography of the given text. In the following example number and percent sign is transformed into written form [37].

```
The 5% mixture -> THE FIVE PERCENT MIXTURE
```

Linguistic Analysis Module:

Linguistic Analysis Module determines the syntactic and semantic features of words, phrases, clauses and sentences. This module helps to resolve grammatical features and part of speech of individual words. Parsing also helps for deriving prosodic structure useful in determining segmental duration and pitch contour. Parsing can contribute to the syntactic type of the sentence. The prosody depends on the question type of the sentence, yes/no question sentences will differ in their prosody contour from who/where sentences.

2.2.2 Phonetic Analysis

The phonetic analysis module is responsible for converting lexical ortographic symbols to phonetic representation with stress information. The following modules are needed to have accurate pronounciations.

Homograph Disambiguation:

Sense disambiguity occurs when words have different syntactic/semantic meanings. Homograph disambiguation module disambiguates these ambiguities. Words with different senses are called polysemous words. Polysemous words that are pronounced differently are called homographs. Some homographs such as object, absent, minute etc can be resolved by their part-of-speech. Object is pronounced as (/**ah b jh eh k t**/) as a verb and (/**aa b jh eh k t**/) as a noun. But sometimes as

in the word *read*, even human readers can not resolve the pronounciation without the context.

Morphological Analysis:

The morphological analysis module relates a surface ortographic form to its pronounciation by analyzing its component morphemes, such as prefixes, suffixes and stem words. This decomposition process is referred as morphological analysis [12].

Letter-to-sound Conversion:

Letter-to-sound conversion is the last stage of phonetic analysis where general letter-to-sound rules are applied and a dictionary lookup is made to produce accurate pronounciations for an arbitrary word. The letter-to-sound (LTS) module automatically determines the phonetic transcription of the incoming text. The most reliable way for grapheme to phoneme conversion is via dictionary lookup. If dictionary lookup fails, rules may be used to generate the phonetic forms. An example rule can be given as follows:

> ortographic *k* can be changed to a velar[1] plosive[2] /k/ when k is word initial ('[') followed by n

can be given as

```
k -> /sil/   %   [  _ n
k -> /k/
```

The rewrite rules above indicate that *k* is transformed into silence when it is in word initial position and followed by *n*, otherwise it is rewritten as phonetic /k/. The underscore in the first line is a placeholder for the *k* itself. The word *knight* is an example for the realisation of the letter *k* into silence. Generally a TTS system requires hundreds of these rules for converting words that are not present in the exception list, or they may be listed offline in a lexicon. We can organize the LTS module's task into two different ways, dictionary based and rule based:

[1] Velar consonants bring the back of the tongue, up to the rear most top area of the oral cavity

[2] Plosive consonants are formed by a closure in oral cavity

Dictionary based solutions store a maximum of phonological knowledge into a lexicon. In order to keep its size small, entries are restricted to morphemes. The pronounciation of the surface forms is made by inflectional, derivational and compounding morphophonemic rules which describe how the phonetic transcriptions of their morphemic constitutents are combined into words. Morphemes that cannot be found in the lexicon are transcribed by rules. After a first phonemic transcription of each word has been obtained, some phonetic post-processing is generally applied, so as to account for coarticulatory smoothing phenomena [1]. This approach has been applied by the MITTALK system [9]. A dictionary of up to 12,000 morphemes covered about 95% of the input words. The AT&T Bell Laboratories TTS system is quite similar [10], with an augmented morpheme lexicon of 43,000 morphemes [11]. Oflazer and Inkelas [14] have implemented a full scale pronounciation lexicon for Turkish using finite state technology. The system outputs a word's morphological and pronounciation forms.

Rule based solutions transcribe most of the words into phonological forms by applying the phonological rules. Only words that are pronounced in one particular way are stored in an exception dictionary. Notice that, since many exceptions are found in the most frequent words, a reasonably small exceptions dictionary can account for a large fraction of the words in a running text. In English, for instance, 2000 words typically suffice to cover 70% of the words in text [22].

2.3 Digital Signal Processing Component

Figure 2.4 shows the skeleton of a general digital signal processing module for TTS systems [37]. The module consists of *Prosodic Analysis* and *Speech Synthesis* components:

2.3.1 Prosodic Analysis

Prosodic features have specific functions in speech communication. Most important function of the prosodic features is to focus a single or a group of words. For instance, the pitch of a syllable may stress a certain word and this may change the whole meaning of the utterance. In other words, certain words must be highlighted for a certain meaning of a sentence utterance. The term prosody refers to certain

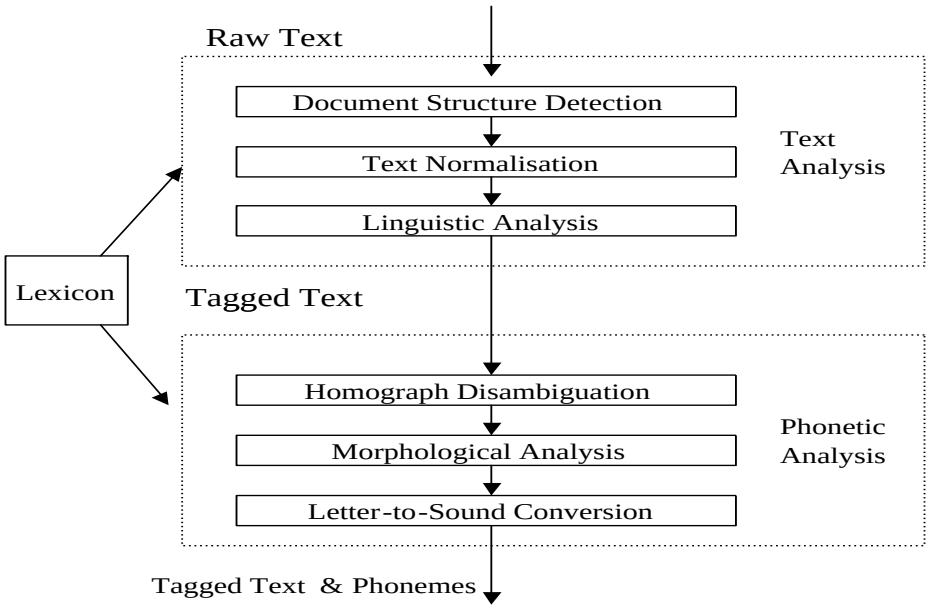

Figure 2.4: Digital signal processing component of a general TTS conversion system

properties of the speech signal which are related to audible changes in pitch, loudness, syllable length [1]. It expresses linguistic information such as sentence type and phrasing as well as paralinguistic[3] information such as emotion. It is characterized by three main parameters: fundamental frequency, duration and intensity. Figure 2.5 shows the components of the prosodic analysis module [37]. Enriched prosody representation (as seen in Figure 2.6) consists of lines of information where each line specifies the phoneme name, the phoneme duration in milliseconds, and a number of prosody points specifying pitch and sometimes volume [37]. For instance, the fourth line specifies values for phoneme /k/ which lasts 80 ms. It also has three prosodic targets. The first is located at 25% of the phoneme duration, i.e., placed at the 20 ms of the phoneme duration and has a pitch value of 178 Hz.

The duration of segments is difficult to describe. It correlates with many pragmatic categories, like sentence focus, emphasis and stress [13]. One can synthesize each phoneme with its average duration. Alternatively, duration can be specified by hand-written rules [15] or numerical models [16]. Although duration can be described categorically, it is essentially a continuous variable, and should therefore be well-suited to methods like non-linear regression [16], regression trees [17], or neural

[3]Pitch-range variation that is correlated with emotion or other aspects of the speech event is called paralinguistic feature.

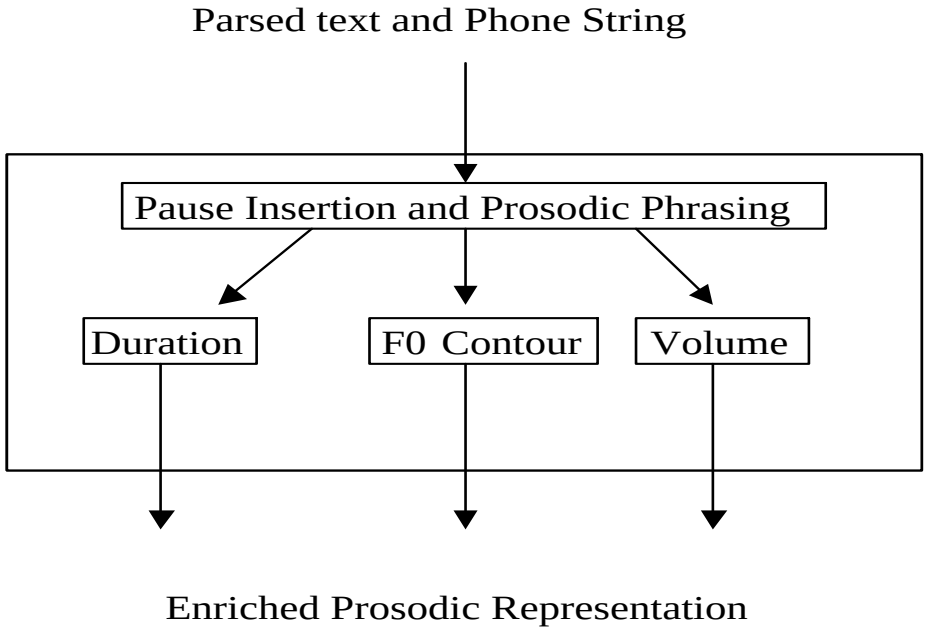

Figure 2.5: Block Diagram of a prosody generation system

networks. However, statistical models or rule-based models require large amounts of labelled training data, which are difficult to produce. Therefore, using average phoneme durations appears to be the only viable solution.

Intonation is represented in two different ways. One is based on the pitch contour of an utterance. Pitch contour is a sequence of rises and falls of F0 [18]. The other way of representation [19] describes intonation as a sequence of pitch targets. There are two levels of targets or tones : high (H,maximum) and low (L,minimum). These targets mark either a pitch accent, that is, extremum in the pitch contour, or a prosodic boundary (boundary tone). The target that bears a nuclear accent is starred, boundary tones are marked by the sign %. TOBI (Tones and Break Indices) [21] is one of the most popular phonological labelling system, which is used widely for annotating prosodic corpora. In TOBI (Tones and Break Indices) labelling, the tones are coupled with break indices [23] for marking prosodic phrase boundaries.

Both approaches are on a phonological level: they are used to describe the rough pitch contour of an utterance and have to be transformed into phonetic descriptions. If one wants to synthesize a pitch contour from a TOBI (Tones and Break Indices) labelling, one needs to know to what values the tones correspond, how to interpolate between them, how to model phrase boundaries, the phonological processes which

```
DH,   24          (0,178);
AH0, 104          ;
#;
K,    80          (25,178)    (50,184)    (75,201);
AE1, 152          (0,214)     (25,213)    (50,204)    (75,193);
T,    40          (0,175)     (25,175)    (50,174)    (75,172);
#;
K,    104         (0,171)     (25,172)    (50,180)    (75,189);
AE1, 104          (0,198)     (25,196)    (50,168)    (75,137);
T,    112         (0,120)     (100,200);
#;
```

Figure 2.6: Enriched Prosody Representation

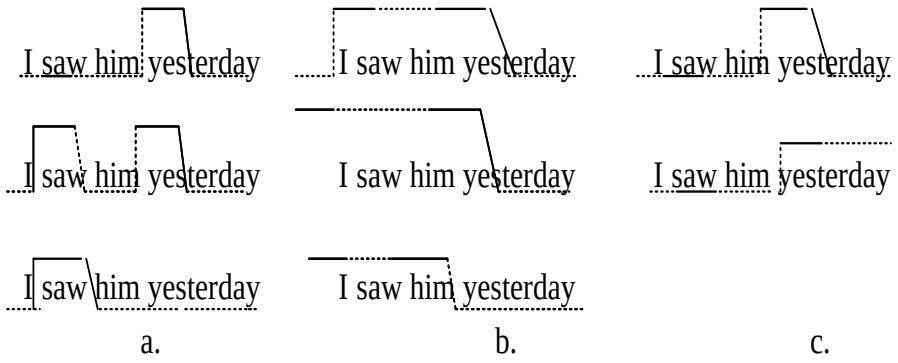

Figure 2.7: Different kinds of information provided by intonation (lines indicate pitch movements; solid lines indicate stress [1]. a. Focus or given/new information; b. Relationships between words (saw-yesterday; I-yesterday; I-him) c. Finality (top) or continuation (bottom), as it appears on the last syllable;

are blocked by those boundaries. Furthermore, we need to set global parameters for the pitch contour like a top line and a base line, which bound pitch excursions. During speaking, fundamental frequency declines slowly, and this fact has to be considered too.

Stress is an important information carrier. Word stress determines which syllable in a word is stressed, phrase stress determines the words in a phrase that receive stress. Stress can be signalled by all prosodic parameters *pitch, intensity, and duration*, as well as by *segment quality*. Stressed syllables tend to be longer than unstressed ones, and they are usually further marked by a local maximum or minimum in the fundamental frequency contour.

2.3.2 Phonetics

Human speech is produced by vocal organs as presented in Figure 2.8. The main energy source is the lungs with the diaphragm. When speaking, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract, the pharynx and the oral and nasal cavities. From the oral and nasal cavities the air flow exits through the nose and mouth, respectively. The V-shaped opening between the vocal cords, called the glottis, is the most important sound source in the vocal system. The most important function of vocal cords is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which vowels and voiced consonants are produced. The frequency of vocal fold vibration is called *fundamental frequency*. The fundamental frequency of vibration depends on gender and physical properties of the mouth and nose cavity and is about 110 Hz, 200 Hz, and 300 Hz with men, women, and children, respectively. With stop consonants the vocal cords act suddenly from a completely closed position in which they cut the air flow completely, to totally open position producing a light cough or a glottal stop. On the other hand, with unvoiced consonants, such as /s/ or /f/, they may be completely open. An intermediate position may also occur with for example phonemes like /h/.

2.3.3 Speech Synthesis

Intuitively, the operations involved in the digital signal processing module are the computer analogue of dynamically controlling the articulatory muscles and vibratory frequency of the vocal folds so that the output signal matches the input requirements [1]. It has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech [24]. This can be achieved in two ways:

Rule Based/Formant Synthesizers:

Rule-based synthesizers have two major components: the first is a generator for the excitation signal and the second is a filter that simulates the effect of the vocal tract. The parameters of the filter are derived from acoustic specifications. In other words, a vowel can be synthesized by passing a glottal periodic waveform through a

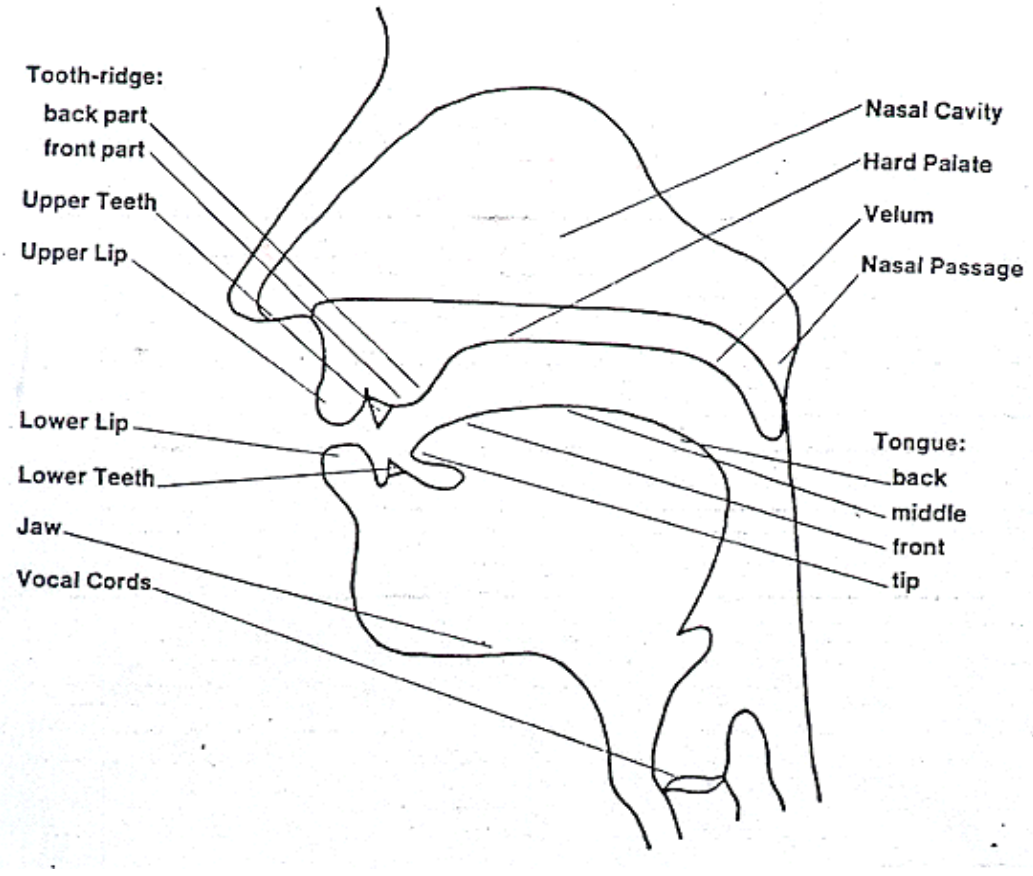

Figure 2.8: The human vocal organs

filter with the formant frequencies of the vocal tract. When synthesizing unvoiced speech white random noise [4] can be used for the source instead. Since speech signals are not stationary the pitch of the glottal source and the formant frequencies change over time. Synthesis-by-rule refers to a set of rules on how to modify pitch, formant frequencies and other parameters from one sound to another while maintaining the continuity present in physical systems like the human production system. Figure 2.9 is a model of such a system. Rule-based synthesizers provide great assistance

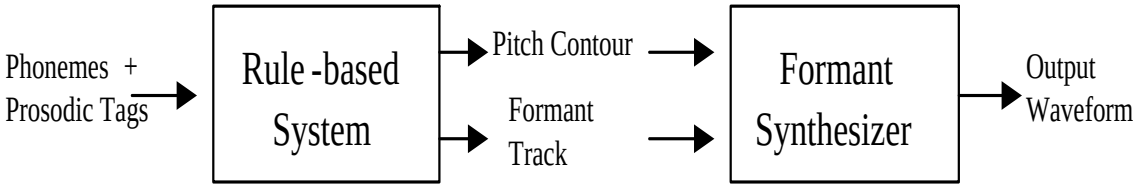

Figure 2.9: Block diagram of a synthesis-by-rule system

[4]Noise is called white noise if, and only if, its samples are uncorrelated.

to phonologists since they constitute a cognitive approach of the phonation mechanism. The spread of the Klatt synthesizer [6] is due to its invaluable assistance in the study of the characteristics of natural speech. The relationships between articulatory parameters and the inputs of the Klatt model make it a practical tool for investigating physiological constraints [25].

Concatenative Synthesizers:

Concatenative speech synthesis systems generate speech by concatenating and manipulating prerecorded units of speech. Choice and storage of units and concatenation method are important in this method. Coarticulatory effects have to be modelled using a potentially limited unit inventory. A balance has to be found between the quality and the size of the inventory. As the units get larger, quality increases but on the other hand more units have to be recorded and segmented. A compilation of unit types for English is shown in Table 2.1.

Unit Length	**Unit Type**	**Number of Units**	**Quality**
Short	Phoneme	42	Low
	Diphone	1500	
	Triphone	30K	
	Demisyllable	2000	
	Syllable	15K	
	Word	100K-1.5M	
	Phrase	∞	
Long	Sentence	∞	High

Table 2.1: Unit types in English assuming a phone set of 42 phonemes. Longer Units produce higher quality at the expense of more storage.

The first approach used in concatenative synthesis was concatenating single phonemes [26]. Having one instance of each phoneme, independent of the neigbouring context, is very generalizable. It allows us to generate every word/sentence. This method yielded rather poor quality since context-independent phones result in many audible discontinuities.

Quality improves dramatically when diphones are used. Diphone units consist

of the transition between two phonemes p1,p2. For instance, while synthesizing the word *hello* /hh ax l ow/, we have to concatenate the diphones /sil-hh/ , /hh-ax/ , /ax-l/ , /l-ow/ , and /ow-sil/. While using the diphone concatenation, we must assume that the transition between the two phones is sufficient to model all neccessary coarticulatory effects. The other assumption is that the spectra of the steady states of the phones are consistent enough to avoid spectral discontinuties.

Comparison of Rule-Based and Concatenative Synthesis:

With rule based systems, one can adjust a high number of parameters and obtain high-quality speech. Discontinuties arising from concatenating pre-recorded speech units is not a problem with this method. However, setting parameters and devising rule sets such that the resulting speech is both intelligible and natural is very difficult.

On the other hand, a concatenative approach only needs the phonetic layout of the language, a good concatenation algorithm and a patient speaker. A concatenative synthesizer is also more natural when compared with rule based synthesizers, since the parameters are already built in.

Chapter 3

The Festival Speech Synthesis System

We used the Festival [31] system for the work presented in this thesis. This chapter describes the relevant details of this system.

3.1 Introduction To Festival Speech Synthesis System

Festival offers a general framework for building speech synthesis systems and also includes examples of various modules. It provides us with a full TTS system through API's[1]: using Scheme command interpreter, or C++ library. It is a multi-lingual system. Voices in English (UK and US), Spanish, Welsh have been developed with this system. The system is written in C++ and mainly uses the Edinburgh Speech Tools for low level architecture and has a Scheme based command interpreter for control.

3.2 Festival Text To Speech

Festival supports text-to-speech for raw text files. At command line,

```
festival -tts textfile
```

renders input text file into speech waveform. In other words this command will say the contents of the text file.

3.3 Utterance Structure

The basic building block for Festival is the *utterance*. Utterance structure consists of a set of relations over a set of items. Items represent objects such as words,

[1] Application Programming Interfaces

segments, syllables, etc. A relation is a set of named links connecting a set of nodes. Relations take any graph structure but they are commonly lists or trees. Relations relate the items. Items can belong to multiple relations. For example a *segment item* can belong both to *segment relation* and a *sylstructure relation*. Relations form an ordered structure over the items within them. Items consist of a bundle of features and a set of named links to nodes in relation.

Figure 3.1 shows an example utterance with a syntax relation and a word relation. The *word relation* contains nodes that have next and previous connections whereas the syntax relation has up and down connections. Every node in the syntax tree is connected to an item. *Word items* are linked to two relations.

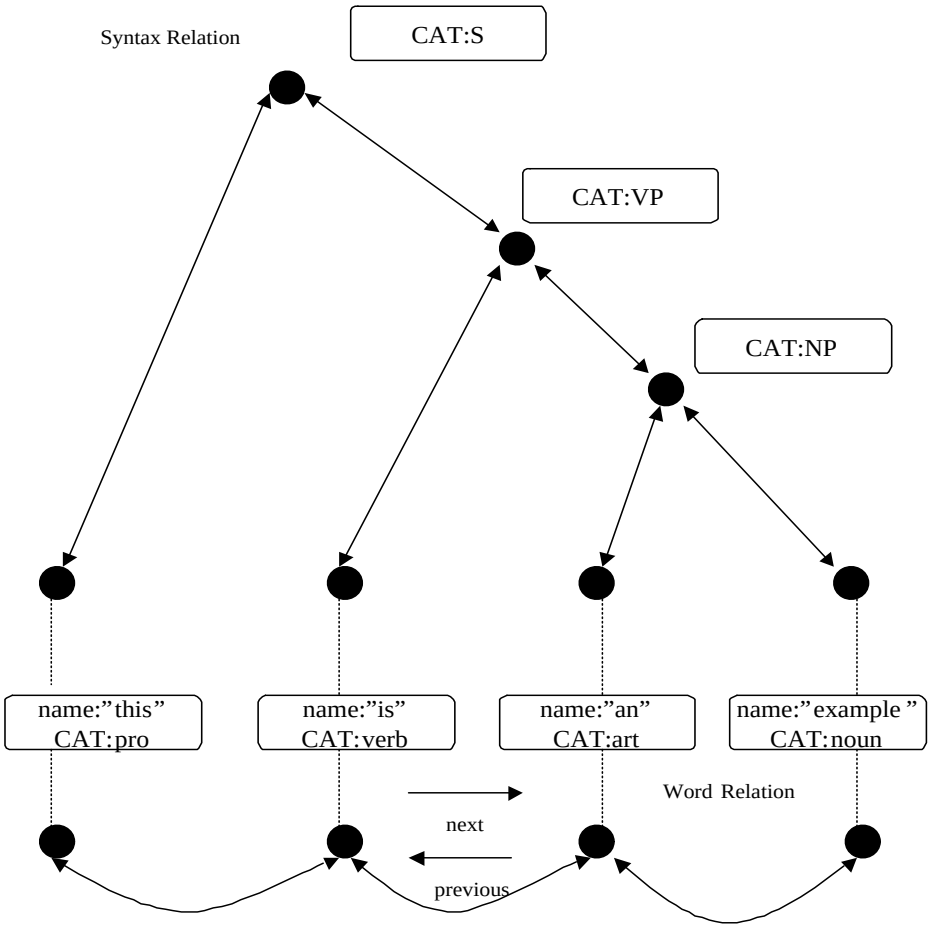

Figure 3.1: An example representation of an utterance structure. This example shows the word relation and the syntax relation. The syntax relation (shown on top) is a tree with links connecting the nodes, shown as black circles. The word relation (shown on the bottom) is a list. The items contain the actual linguistic information and are shown in the rounded boxes. The dotted lines show the connections between the nodes and items.

3.4 Relations

The relations for the basic English TTS are listed below.

- *Text Relation* contains a single item which contains a feature with the input character string that will be synthesized.
- *Token Relation* is a list of trees where root of each tree contains tokenized object obtained from the input character string. Punctuation and white space are stripped and placed on features on these token items.

- *Word Relation* Words represent the words in the utterance. Words are leaves of the *Token Relation*, leaves of the *Phrase Relation* and roots of the *Sylstructure Relation*.
- *Phrase Relation* represents the words in the utterance. Words are leaves of the *Token Relation*, leaves of the *Phrase Relation* and roots of the *Sylstructure Relation*.
- *Syllable Relation* represents a simple list of syllable items which are the intermediate nodes in the *Sylstructure Relation*.
- *Segment Relation* represents a simple list of segment(phoneme) items, which form the leaves of the *Sylstructure Relation* through which we can find where each segment is placed.
- *Sylstructure Relation* represents a list of tree structures over the items in the Word, Syllable and Segment items.
- *IntEvent Relation* represents a simple list of intonation events (accents and boundaries) which are related to syllables through the Intonation relation.
- *Intonation Relation* represents a list of trees whose roots are items in the *Syllable Relation*.
- *Wave Relation* consists of a single item that has a feature with the synthesized waveform.
- *Target Relation* is a list of trees whose roots are segments and daughters are *F0* target points.

3.5 Modules

The synthesis process in Festival consists of applying a number of modules to an utterance. Each module will access various relations and items and generate new features, items and relations. After the modules are applied, the utterance structure will be filled in and the waveform will be generated. The execution order of the modules depends on the utterance type. Most of the time this is defined as tokens. Below you can see definition for utterances of type tokens where utt is the argument.

```
(defUttType Tokens
      (Token_POS utt)
      (Token utt)
      (POS utt)
      (Phrasify utt)
      (Word utt)
      (Pauses utt)
      (Intonation utt)
      (PostLex utt)
      (Duration utt)
      (Int_Targets utt)
      (Wave_Synth utt)
)
```

The modules used for TTS have the following functions

- `Token_Pos`

  Identifies basic tokens, mainly for homograph disambiguation.

- `Token`

  Applies the token-to-word rules, and builds the *Word Relation*.

- `POS`

  If desired, used as a standard part-of-speech tagger.

- `Phrasify`

  Builds the *phrase relation* using the specified method.

- `Word`

  Works as a lexical look up and builds the *syllable* and *segment* relations.

- `Pauses`

  Predicts pauses and inserts silence into the *Segment Relation*. (using prediction mechanisms)

- `Intonation`

  Predicts accents and boundaries, builds the *Intevent* and *Intonation* relations that links IntEvents to syllables.

- `PostLex`

  Organizes rules that can modify segments based on their context. (Used for vowel reduction, contraction)

- `Duration`

  Predicts duration of segments.

- `Int_Targets`

  Creates the *Target Relation* representing the desired *F0* contour.

- `Wave_Synth`

  Calls the appropriate method to generate the waveform.

3.6 Utterance Building

Utterance structures are usually used in the runtime process of converting text to speech. However, one can use them also in database representation. The idea behind this database representation is that we want to build utterance structures for each utterance in a speech database. After obtaining the utterance structures, as if they had been correctly synthesized, one can use these structures for training various models. For instance given the actual durations for the segments in a speech database and utterance structures for these one can save the actual durations and features (phonetic, prosodic context) which influence the durations and training models for that data.

In order to build an utterance, we need label files for the following relations.

- *Segment Labels:* Segments must be labelled with correct boundaries considering the phone set of the language.

- *Syllable Labels:* Syllables must be labelled with stress marking and boundaries should be aligned with the segment boundaries.

- *Word Labels:* Words must be labelled with boundaries aligned close to the syllable and segment labels.

- *IntEvent Labels:* Intonation labels must be aligned to a syllable.

- *Phrase Labels:* Phrase labels must contain a name and marking for the end of each prosodic phrase.

- *Target Labels:* Target labels are the mean *F0* values in Hertz at the mid-point of each segment.

Among these labellings, segment labelling is the hardest to generate since any automatic method will have to make low level phonetic classification. Computers are not very good at this. Autoaligning is another alternative but afterwards hand correction is necessary.

3.7 Diphone Databases

A diphone is a subword unit type which comprise two phones. However, the duration of a diphone is on the average one phoneme long since the beginning of a diphone starts from the middle of the first phone and the end of the diphone is at the middle of the second phone. The word *hello* can be mapped into the diphone sequence : /sil-hh/, /hh-ax/, /ax-l/, /l-ow/, /ow-sil/.

For diphone synthesis, nearly all possible phone-phone transitions in a language must be listed. In general, the number of diphones in a language is the square of the number of phones. However due to phonotactic constraints some phone-phone pairs may not occur at all. However people can often generate the non-existent diphones if they try. Moreover, one must think about phone pairs that cross over word boundaries as well. But even then certain combinations can not exist; for example, /hh-ng/ diphone in English is probably impossible. The /ng/ phoneme may only appear after the vowel in a syllable-initial position. The /hh/ phoneme can not appear at the end of a syllable, though sometimes it may be pronounced when trying to add aspiration to open vowels.

In fact, co-articulatory effects may go over more than two phones. However, the diphone method assumes that this is not true. Unlike unit selection, which will be explained later, only one occurrence of a diphone is recorded. This makes selection easier but collection task is laborious.

When humans are given a context that carries an unusual phoneme, they try to produce it even it falls outside their phonetic vocabulary. Concatenative syn-

thesizers, however can not produce anything outside their pre-defined vocabulary. Formant and articulatory synthesizers have advantage here. Since diphones need to be cleanly articulated, various techniques have been proposed. One technique is to use target words embedded carrier sentences to ensure that the diphones are pronounced with acceptable duration and prosody (i.e consistently). One other technique is using nonsense words that iterate through all possible diphone combinations. The advantage of using nonsense words is that the presentation is less prone to pronounciation errors. For best results, the words should be pronounced with consistent vocal effort, with as little prosodic variation as possible. Pronouncing in a monotone way is ideal. Nonsense words consist of a carrier where the diphone is usually taken from a middle syllable. Classes of diphones, for instance: vowel-consonant, consonant-vowel, vowel-vowel and consonant-consonant must be extracted. Then, carrier contexts have to be defined for these groups.

The following pseudo code lists nonsense words for all possible list of vowel-vowel diphones.

```
for v1 in vowels
    for v2 in vowels
          print pause t aa t $v1 $v2 t aa pause
```

One must consider how easy it is for the speaker to pronounce these nonsense words.

3.7.1 Extracting the Pitchmarks

Festival supports residual excited Linear-Predictive-Coding (LPC) resynthesis [27]. It does support PSOLA [29] but this is not distributed in the public version. Both of these techniques are pitch synchronous, meaning they require information about where pitch periods occur in the acoustic signal. If it is possible, recording with an electroglottograph (EGG, also known as laryngograph) is better. The EGG records electrical activity in the glottis during speech, which makes it easier to get the pitch moments.

Although extracting pitch periods from the EGG signal is not very easy, it is fairly straightforward in practice as the Edinburgh Speech Tools include a program which processes the EGG signal and gives a set of pitchmarks. This program filters the incoming waveform (with a low and a high band filter), then uses autocorrelation

to find the pitch mark peaks with the minimum and maximum specified. Finally, it fills in the unvoiced section with the default pitchmarks. However it is not fully automatic and requires someone to inspect the result and play with the parameters so to improve the results.

If the signal is inverted -inv should be added to the arguments to `pitchmark`. The object is to produce a single mark at the peak of each pitch period and phantom periods during unvoiced regions.

The command is as follows: Notice that -min and -max arguments are speaker dependent.

```
pitchmark lar/file001.lar -o pm/file001.pm -otype est
-min 0.005 -max 0.012 -fill -def 0.01 -wave_end
```

If EGG signals do not exist for the diphones, another alternative is extracting the pitch periods using some other signal processing function. Finding the pitch periods is similar to finding the *F0* contour. It is harder than extracting from the EGG signals but still possible with clean laboratory recorded speech.

The following script is a modification of the above script above for extracting waveforms from a raw waveform signal. It is not as good as extracting the EGG signal but it works. It is more computationally expensive since it requires rather high order filters. The value should be changed according to the speaker's pitch range.

```
for i in $
do
  fname='basename $i. wav'
  echo $i
  $ESTDIR/bin/ch_wave -scaleN 0.9 $i -F 16000 -o /tmp/tmp$$.wav
  $ESTDIR/bin/pitchmark /tmp/tmp$$.wav -o pm/$fname.pm
  -otype  est -min 0.005 -max 0.012 -fill -def 0.01
  -wave_end -lx_lf 200 -lx_lo 71 -lx_hf 80 -lx_ho 71 -med_o 0
done
```

If the pitch periods are extracted automatically, it is worth taking more care to check the signal. Recording consistency and bad pitch extraction are the two most

common causes of poor quality synthesis. Pitchmarks can be displayed using the emulabel tool. Figure 3.2 shows the pitchmarks extracted with the 'pitchmark' command. The pitchmarks (vertical lines) should be aligned to the largest peak (circles) in each pitch period. In this figure this aligning is satisfactory for the pitchmarks.

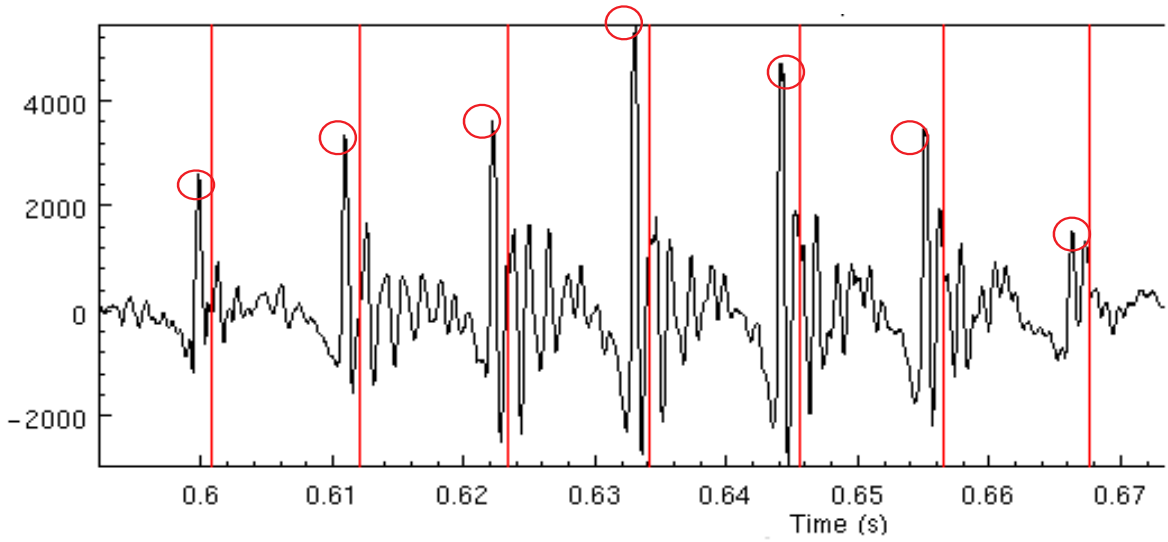

Figure 3.2: Close-up pitchmarks in waveform signal

3.8 Unit Selection Databases

Unit selection is the selection of unit of speech which may be anything from a whole phrase down to a diphone (or even smaller). Technically, diphone selection is a simple case of this. In unit selection, there is usually more than one example of the unit and some mechanism is used to select between them at run-time. Unit selection starts with a phonetic and prosodic specification for a desired utterance. Each phone has a feature vector, including at least pitch, duration, and stress and also carries the phonetic context from its preceding and following phones. ATR's CHATR system [28] is an excellent example for the method of selecting between multiple examples of a phone within a database. Diphone method is not ideal since only a fixed view of the possible space of speech units are made. In some words, there are

articulatory effects which go over more than one phone. For instance, in words *spout* and *spit* the roundness of the following vowel, affect the pronounciation of the letter *s* although there is an intermediate stop. It is not only obvious segmental effects that cause variation in pronounciation, syllable position, word/phrase initial and final position have different level of articulations. Inter-syllable or intra-syllable or word-initial or word-internal positions affect articulation. Stressing and accents also cause differences. Rather than listing all of these events and recording all of them, an alternative is to take a natural distribution of speech and (semi-)automatically find the distinctions that exist rather than predefining them. The success of such systems vary. They can produce very high quality, natural sounding synthesis. However when the database has unexpected holes or the selection costs fail, they can produce very bad synthesis too.

3.8.1 Cluster Unit Selection

This part is a reimplementation of the techniques described in Black and Taylor's work [30]. Unit selection is based on taking a database of general speech and trying to cluster each phone type into groups of acoustically similar units based on the (non-acoustic) information available at synthesis time. These non-acoustic information include phonetic context, prosodic features (*F0* and duration), stressing, word position and accents. This work is similar to previous works like CHATR selection algorithm [28] and the work of Donovan [32]. But this work differs from Hunt' work [28] since it builds CART (Classification and Regression Tree) trees to select the appropriate cluster of candidate phones and as a result it does not calculate target costs (through linear regression) at selection time. As the clusters are built directly from the acoustic figures and target features, a target estimation function is not required. This clustering method differs from Donovan's work since it uses a different acoustic cost function (Donovan uses HMM's). Donovan selects one candidate while in this technique a group of candidates are selected. The basic processes involved in building a waveform synthesizer for clustering algorithm is as follows:

- *Collect the database of general speech.*
- *Build utterance structures for your database.*

- *Build coefficients for acoustic distances, Cepstrum plus F0 or some pitch synchronous analysis (LPC)*
- *Build distance tables*
- *Dump selection features (phone context, prosodic, positional) for each unit type.*
- *Build cluster trees using 'wagon' with the features and acoustic distances dumped by the previous two stages.*
- *Build the voice description itself.*

3.8.2 Diphones from general databases

In this method, we use the diphones as a unit. We should have a general database that is labelled with utterances as described above. We can extract a standard diphone database from this general database. However, we may be unable to cover all phoneme-phoneme transitions. Even in phonetically rich databases like Timit[2], some vowel-vowel diphones does not exist. We can extract a diphone database from the general databases but there may be some holes.

3.9 Building prosodic models

3.9.1 Phrasing

Prosodic phrasing in speech synthesis makes the speech more understandable. Due to our lungs, there is a finite length of of time we can talk without taking a new breath. This defines the upper bound on prosodic phrases. However, we usually take breath before this upper bound. We use phrasing to mark groups within the speech.

For English and many other languages, simple rules that are based on punctuation are very good predictors of prosodic phrase boundaries. If there is a punctuation than there usually exists a prosodic boundary. But sometimes a prosodic boundary exists although there is no punctuation mark. Thus a phrosodic phrasing algorithm solely based on punctuation will typically under predict but rarely make a false

[2]http://www.mpi.nl/world/tg/corpora/timit/timit.html

insertion. However, depending on the application it may be the case that explicitly adding punctuation at desired phrase breaks is possible and adequate.

Festival supports two methods for predicting prosodic phrases. The first basic method is by CART (Classification and Regression Tree). A test is made on each word to predict if it is at the end of a prosodic phrase. The CART (Classification and Regression Tree) tree returns B(short break) or BB(long break). BB denotes end of utterance.

The following tree adds a break after the last word of a token that has the following punctuation.

```
(set! simple_phrase_cart_tree
'
((lisp_token_end_punc in ("?" "." ":"))
  ((BB))
  ((lisp_token_end_punc in ("'" "\"" ";" ","))
   ((B))
   ((n.name is 0)  ;; end of utterance
    ((BB))
    ((NB))))))
```

As the basic punctuation model underpredicts, we need information that will find reasonable boundaries within strings of words. In English, boundaries are more likely between content words. If we have no data to train from, then written rules in a CART tree can give a phrasing model better than pure punctuation rules.

To implement such a scheme we need three basic functions:

- *determining if the current word is a function or content word*
- *determining number of words since previous punctuation*
- *determining number of words to next punctuation*

A much better method for predicting phrase breaks is using a full statistical model trained from data. But the problem with this method is that you need a lot of training data to train phrase break models.

Wagon tool[3] can be used to produce CART (Classification and Regression Tree)

[3]A program developed within Festival Speech Tools.

trees based on the features we identify. Possible features may be

- *lisp_token_end_punc*
- *lisp_until_punctuation*
- *lisp_since_punctuation*
- *p.gpos*
- *gpos*
- *n.gpos*

However without a good intonation and duration model spending time on producing good phrasing is probably not worth it.

3.9.2 Accent/Boundary Assignment

Content words are the key words of a sentence. They are the important words that carry the meaning or sense. For example in the following sentence, capitalized words are content words.

Will you SELL my CAR because I've GONE to FRANCE

The rest of the words are structure words. They make the sentence grammatically correct. For English, the placements of accents on stressed syllables in all content words is quite a reasonable approximation and achives about 80% accuracy on typical databases. Using this method achieving simple, in other words `discourse neutral intonation` is relatively easy. But achieving realistic, natural accent placement is still beyond this method. Here is a simple CART tree that predicts accents of stressed syllables in content words.

```
(set! simple_accent_cart_xtree
  '
  (
   (R:SylStructure.parent.gpos is content)
    ( (stress is 1)
       ((Accented))
```

```
        ((NONE))
    )
  )
)
```

3.9.3 F0 Generation

Based on the place of the accents, an *F0* contour must be built. Accent positions influence durations and the *F0* contour can not be generated without knowing the durations of the segments the contour is to be generated over. There are three basic *F0* generation modules in Festival: by general rule, by linear regression/CART, and by TILT [33] [4].

3.9.4 F0 by rule

This is the most general *F0* generation method. This method allows target points to be programmatically created for each syllable in the utterance. The simple idea behind this general method is that a LISP function is called for each syllable in the utterance. This LISP function returns a list of target *F0* points that lie within that syllable. This method allows the user to program any *F0* value. The idea behind this technique extends from the implementation of TOBI type accents, where a number of points are predicted for each accent. The baseline is the average *F0* of the speaker. To the end of the phrase the *F0* declines slowly. This technique and TOBI place *F0* target points above and below that baseline depending on the accent type and position in phrase. For example a simple accent can be generated using this technique as follows.

```
(define (targ_func1 utt syl)
  "(targ_func1 UTT STREAMITEM)
Returns a list of targets for the given syllable."
  (let ((start (item.feat syl 'syllable_start))
        (end (item.feat syl 'syllable_end)))
    (if (equal? (item.feat syl "R:Intonation.daughter1.name") "Accented")
        (list
```

[4] TILT is a model for prosodic annotation.

```
(list start 110)
(list (/ (+ start end) 2.0) 140)
(list end 100 )))))
```

This method checks if the current syllable is accented and returns a list of target pairs. It assigns 110 Hz at the start point, 140 Hz at the mid-point of the syllable and finally 100 Hz at the end of the syllable.

This technique can be expanded with other rules as neccesary. Festival includes an implementation of TOBI using this technique.

3.9.5 F0 by linear regression

This technique finds the appropriate *F0* target value for each syllable based on available features obtained from training data. A set of features are collected for each syllable and a linear regression model is used to model three points on each syllable. This technique provides reasonable synthesis and requires less analysis of intonation models. In most synthesizers, the task of generating a prosodic tune consists of two sub-tasks, the prediction of intonation labels (accents, tones, etc) from text and the generation of a contour from those labels [17]. *F0* contours can be generated from TOBI labelled utterances. The TOBI labelling system [17] offers a method for labelling pertinent aspects of intonation in speech. Although, there are recognized limitations with the system, it has been used to hand-label large speech databases and is being used in a number of synthesis systems. TOBI labelling for an utterance consists of three tiers each related (through time) to a speech waveform [17]. The tiers are: labels, break indices and miscellaneous. The label tier marks pitch accents, phrase accents and boundary tones. The break index tier marks one of four levels of prosodic breaks. The miscellaneous tier may contain any other labelling, such as background noise, coughing, laughing, disfluencies[5] or anything else that might be labelled. One method of generating an *F0* contour from such labels and breaks is described in [20], which is called the APL method. The APL method predicts a number of target points for each syllable marked with a pitch accent, phrase accent or boundary tone. A number of specific rules deal with each case [17]. For example consider the following Figure 3.3 (from Black Hunt's work[17]).

[5]Stalls, hesitations and self-repairs, in normal spontaneous speech

The horizontal axis shows time, and the vertical axis shows frequency, *F0*. An H*

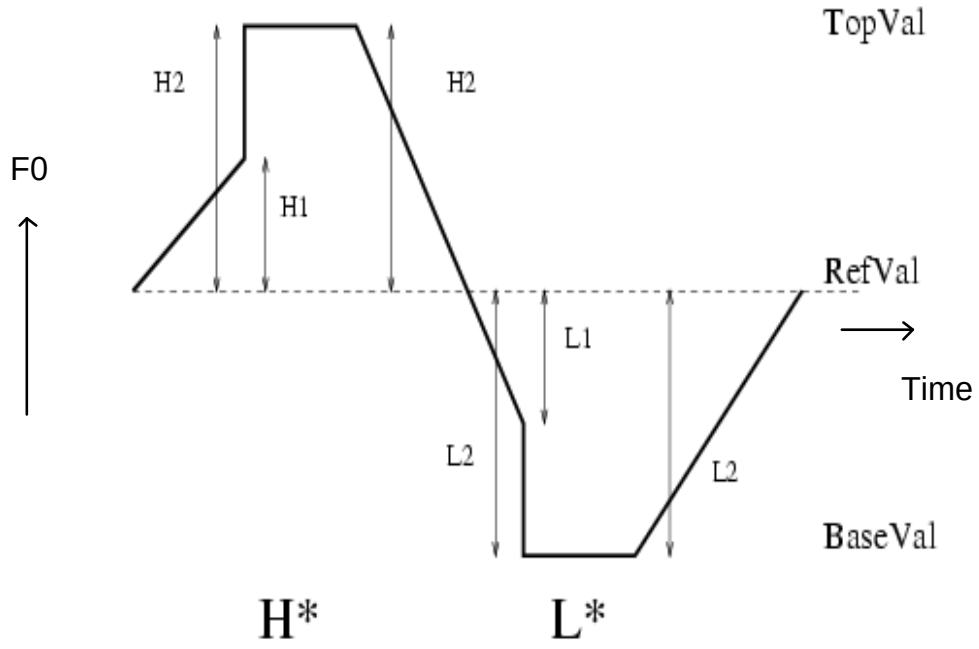

Figure 3.3: TOBI Parameters

accent introduces three target points, the first at height H1 above the reference line at the start of the syllable, the second at height H2 at the start, and the third at H2 at the end of the syllable, there is similar target points for L*. The parameters H1, H2 etc. are given as fractions of *TopVal*[6] and *BaseVal*[7] above or below *RefVal*[8] , so there is some independence from absolute pitch range. Independently, *RefVal*, *TopVal* and *BaseVal* may decrease over time to represent declination. However, this technique depends solely on training data (TOBI labellings), some issues for instance multiple accents on syllables, accent placement with respect to the vowel are not captured. The previous technique allows specification of structure without explicit training from data, on the other hand this technique imposes no structure and depends on data. Tilt modelling, which will be described next, tries to balance these two extremes.

[6]Size in Hertz above refval for maximum sized accents (speaker-dependent).

[7]Size in Hertz below refval for minimum sized accents (speaker-dependent).

[8]Size in Hertz of mid-value. For most speakers this is best set to the mean F0 of the speaker.

3.9.6 Tilt Modelling

A Tilt labelling for an utterance consists of an assignment of one of four basic intonational events: pitch accents, boundary tones, connections and silence. An intonational event is a general term for phonologically significant intonational effect. Connections represent the parts of contours where there is nothing of intonational significance. Tilt modelling is still under development and not as mature as the other methods. A tilt parameterization of a natural *F0* contour can be automatically derived from a waveform and a labelling of accent placements. An *a* label is used for accents, *b* for boundaries, *c* for connections, and *sil* for silence. For each *a* label four continous parameters are found: height, duration, peak position, with respect to the vowel start, and tilt. This method gives better results when compared with linear regression models but has not been tried on new languages other than English.

The automatic parameterization of a pitch event on a syllable is in terms of:

- *starting F0 value(Hz)*
- *duration*
- *amplitude of rise (Arise, in Hz)*
- *amplitude of rise (Arise, in Hz)*
- *starting point, time aligned with the signal and with the vowel onset*

Figure 3.4 shows the Tilt paramters.

Tilt parameter is the difference of the amplitude divided by their sum.

$$tilt = \frac{|Arise| - |Afall|}{|Arise| + |Afall|}$$

The tilt parameter has a range of -1 to 1 where -1 is pure fall, 1 is pure rise and 0 contains equal portions of rise and fall.

3.9.7 Duration

Similar to the prosody generation, simple solutions for predicting durations of segments work surprisingly well, but very good solutions are extremely difficult to achieve.

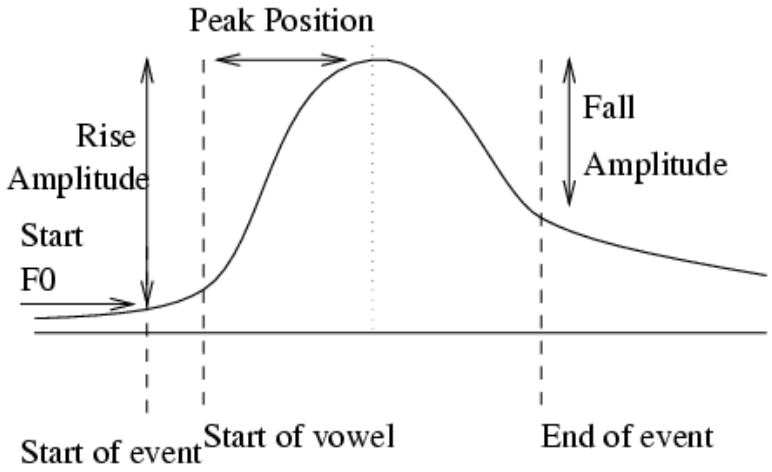

Figure 3.4: Tilt Parameters

The simplest model for duration is a fixed duration for each phone. 100 milliseconds is a reasonable value for phone duration. But this fixed duration sounds too artificial. The next step is to use average durations for the phones. Even when real data is not available to calculate averages, writing values by hand to the CART tree can be acceptable. Usually vowels are longer than the consonants and stops are the shortest. In most languages, phones are longer at the phrase final and shorter at phrase initial positions. Thus, we can define a set of rules that modify the basic average based on the context they occur in. Here we define a simple decision tree that returns a multiplication factor for the duration of a segment.

```
(set! simple_dur_tree
 '
   ((R:SylStructure.parent.R:Syllable.p.syl_break > 1 ) ;; clause initial
    ((R:SylStructure.parent.stress is 1)
     ((1.5))
     ((1.2)))
    ((R:SylStructure.parent.syl_break > 1)   ;; clause final
     ((R:SylStructure.parent.stress is 1)
      ((1.5))
      ((1.2)))
```

```
    ((R:SylStructure.parent.stress is 1)
     ((ph_vc is +)
      ((1.2))
      ((1.0)))
     ((1.0))))))
```

Below we set the average durations for each phoneme (segment) as follows. The format of this information is `segment name average duration`.

```
(set! simple_phone_data
 '(
  (#  0.250)
  (a  0.080)
  (e  0.080)
  (i   0.070)
  (o  0.080)
  (u  0.040)
...
   ))
```

Training from data is better when building duration models. We can save durations and features for each segment in the traning database. Then, we can train a model using these samples.

Chapter 4

A Prosodic Turkish Text-To-Speech System

This chapter describes the implementation of a Prosodic Text-To-Speech system for Turkish using the Festival Speech Synthesis System. First, we describe the aspects of Turkish Language.

4.1 Turkish Phonetization

Phonetics is the study of speech sounds and their production, classification and transcription. Similar to fingerprints, every speaker has different vocal anatomy which makes him unique. Yet, there is a generality at the perceptual level. Researchers have studied to capture these common generalities. Phonological representation and rule formalisms are important for TTS systems. For the following discussions, we use the SAMPA[1] notation to show pronounciations in text. Turkish has an eight vowel inventory : /i, y, e, 2, a, o, 1, u/ which correspond to i, ü, e, ö, a, o, and u in Turkish ortography. Table 4.1 groups the vowels according to their backness, roundness and height. Vowels are classified as *Front* or *Back* according to whether it is the front or back of the tongue which interrupts the flow of breath. Similarly, *High* or *Low* contrast the amount of space left between tongue and palate; alternative terms are *Open* and *Close*. Finally, *Rounded* and *Unrounded* describe the shape of the lips.

Turkish has also long vowels, which usually come from mainly Arabic and Persian loans. We denote such vowels as /a:, e:, i:, u:/, for instance the word *sakin* /s a: k i n/ is an example of such words, whereas the word *sakın*=/s a k ı n/ has the normal length /a/. One other occurence of long vowels is when voiced velar fricative /ğ/

[1]See http://www.phon.ucl.ac.uk/home/sampa/turkish.htm.

Vowel	Backness	Roundness	Height
a	Back	Non-Round	Low
e	Front	Non-Round	Low
ı	Back	Non-Round	High
i	Front	Non-Round	High
o	Back	Round	Low
ö	Front	Round	Low
u	Back	Round	High
ü	Front	Round	High

Table 4.1: Turkish Vowel Inventory

(yumuşak g) is lost and the vowel preceding it is read longer than the usual (this does not always happen). *Ağaç* /a a tS/ (*tree*) and *eğer* /e e r/ (*if*) are examples of such combinations. Similarly in syllable-final positions the loss of this phoneme lengthens the vowel preceding it. The word *Dağ* /d a:/ (*mountain*) is such an example. In our TTS system we have included all short, normal and long read vowels.

In Turkish consonantal inventory some consonants, like /k, 5, g/ have two allophones where one is palatal and the other is non-palatal /c, l, gj/. Turkish Language has 26 consonants /p, t, tS, k, c, b, d, dZ, g, gj, f, s, S, v, w, z, Z, m, n, N, l, 5, r,d j, h, G/. Velar fricative /G/ is lost in standard Turkish. On the other hand, ortography uses only 21 letters for consonants. We extracted the contextual allophones[2] of Turkish. Turkish has 42 allaphonic phonemes for the vowels. Tables 4.2 and 4.3 show the phoneme list for the vowels. Turkish has 29 phonemes for the consonants. Table 4.4 indicates phoneme list for the consonants. These tables contain the ortography, SAMPA and TTS representations and samples for the phonemes.

Overviews of Turkish phonology can be found in Clement & Sezer's work [34]. In this TTS system, I have used most of the phonemes of this allaphonic phoneme inventory as seen in Tables 4.5 and 4.6.

[2]The process by which neighbouring sounds influence one another is called coarticulation. When the variations resulting from coarticulatory processes can be consciously perceived, the modified phonemes are called allophones.

ORTH.	SAMPA.	TTS-VAR.	SAMPLES.
a	a	A1	**a**b**a**
a	a:	A3	**a**b**a**di
a	a	A5	hay**a**t
a	a	A2	abl**a**tif ; k**a**lp
a	a:	A4	adil**a**ne
a	a	A6	davetk**a**r
e	e	E1	ac**e**m
e	e	E2	ab**e**s ; abid**e**
e	e:	E3	d**eğ**er ; lin**ee**r
e	e:	E4	memur
e	e	E5	**e**ciş
e	e	E6	al**e**yhtar
ı	1	I7	abacı
ı	1	I8	hakkıyla ; babıali
ı	1:	I9	adabalığı

Table 4.2: TURKISH PHONETIC ENCODING FOR VOWELS

4.2 Stress in Turkish

4.2.1 Role of Stress In Turkish Words

Stress is an important information carrier in Turkish, particularly because it interacts with syntactic phenomena like focus, backgrounding and question formation. Word stress determines which syllable in a word is stressed whereas phrase stress determines the words in a phrase that receive stress.

ORTH.	SAMPA.	TTS-VAR.	SAMPLES.
i	i	I5	af**i**f ; as**i**l
i	i	I2	abad**i**
i	i:	I3	amud**i**
i	i:	I6	abidev**i**
i	i:	I1	alen**i**
i	i:	I4	f**iğ**
o	o	O1	albin**o**
o	o:	O2	adem**oğ**lu
o	o	O3	adas**oğ**anı ; alk**o**lit
o	o	O4	brav**o** ; çik**o**lata
ö	2	O5	g**ö**z
ö	2:	O6	**ö**ğretmen
ö	2:	O7	s**öğ**üt
ö	2	O8	k**ö**y
ö	2	O10	**ö**ykü
u	u	U1	d**u**dak
u	u	U3	m**u**aşeret , ah**u**
u	u	U4	am**u**di , ar**u**z
u	u	U5	kaf**u**r
u	u:	U2	b**u**se
u	u:	U6	mesk**u**n
ü	Y	U7	gözl**ü**, y**ü**rek
ü	y	U9	gözl**ük**
ü	y	U12	homoseks**ü**el
ü	y	U10	akt**ü**alite, deval**ü**asyon
ü	y:	U8	d**üğ**me
ü	y:	U11	köp**üğ**ü

Table 4.3: TURKISH PHONETIC ENCODING FOR VOWELS CONTINUED

ORTH.	SAMPA.	TTS-VAR.	SAMPLES.
b	b	B1	baba
c	dZ	C1	can
ç	tS	C2	ağaç
ç	tS	C3	alçak
d	d	D1	dev
f	f	F1	fal
g	g	G1	gaz
g	gj	G2	göz
h	h	H1	han
j	Z	J1	ajan
k	k	K1	kar (snow)
k	c	K2	kar/ kâr (profit)
l	5	L2	sal
l	l	L1	hilal
m	m	M1	mal
n	n	N1	nal
n	N	N2	renk
p	p	P1	pul
r	r	R1	ray
r	r	R2	pazarı
r	r	R3	acar
s	s	S1	ses
ş	S	S2	şal
t	t	T1	tut
v	v	V1	alev
v	w	V2	kavun
y	j	Y1	yay
z	z	Z1	nazik
z	z	Z2	zor

Table 4.4: TURKISH PHONETIC ENCODING FOR CONSONANTS

ORTH.	SAMPA.	TTS-VAR.	SAMPLES.
a	a	A1	**a**b**a**
a	a:	A3	**a**b**a**di
a	a	A5	hay**a**t
a	a	A2	abl**a**tif ; k**a**lp
a	a:	A4	adil**a**ne
a	a	A6	davetk**a**r
e	e	E2	ab**e**s ; abid**e**
e	e:	E3	d**eğ**er ; lin**ee**r
e	e:	E4	memur
ı	1	I7	abacı
ı	1:	I9	adabalı**ğ**ı
i	i	I2	abad**i**
i	i:	I3	amud**i**
o	o	O1	albin**o**
o	o:	O2	adem**oğ**lu
o	o	O3	adas**oğ**anı ; alk**o**lit
ö	2	O5	g**ö**z
ö	2:	O7	s**öğ**üt
u	u	U1	d**u**dak
u:	u	U2	b**u**se
u:	u	U6	mesk**u**n
ü	y	U7	gözl**ü**, y**ü**rek
ü	y	U9	gözl**ük**
ü	y:	U8	d**üğ**me
ü	y:	U11	köp**üğ**ü

Table 4.5: ALLOPHONES USED IN TTS FOR VOWELS

ORTH.	SAMPA.	TTS-VAR.	SAMPLES.
b	b	B1	baba
c	dZ	C1	can
ç	tS	C2	ağac
ç	tS	C3	alçak
d	d	D1	dev
f	f	F1	fal
g	g	G1	gaz
g	gj	G2	göz
h	h	H1	han
j	Z	J1	ajan
k	k	K1	kar (snow)
k	c	K2	kar/ kâr (profit)
l	5	L2	sal
l	l	L1	hilal
m	m	M1	mal
n	n	N1	nal
n	N	N2	renk
p	p	P1	pul
r	r	R1	ray
r	r	R2	pazarı
r	r	R3	acar
s	s	S1	ses
ş	S	S2	şal
t	t	T1	tut
v	v	V1	alev
v	w	V2	kavun
y	j	Y1	yay
z	z	Z1	nazik
z	z	Z2	zor

Table 4.6: ALLOPHONES USED IN TTS FOR CONSONANTS

4.2.2 Phonetic Correlates of Stress

The phonetic correlates of stress appear to be loudness and high pitch. Vowel length does not always appear to be linked to stress perceptibly [35]. It is possible in Turkish words to have a long unstressed vowel and a short, stressed one:

taze *fresh* /t **a:** z é/

4.2.3 Distinctions between different levels of stress

In addition to primary stress, there can be secondary stress. Secondary stress exhibits less loudness, but still prominent in pitch than nonstress and lower in pitch than primary stress.

Secondary stress is found within a phrase or a compound, where the modifier word bears primary phrasal stress, and the head exhibits secondary stress. In such instances both primary stress and secondary stress are located on those syllables where word level stress would occur, if those words were found in isolation ([35], page 504).

4.2.4 Word-accent

Turkish words are usually oxytone (Lewis), i.e accented on the last syllable; when an oxytone word is extended by suffixes the accent is on the last syllable of the word formed :

çocük	*child*	/tS o - ”dZ u k/
çocuklär	*children*	/tS o - dZ u k- ”5 a r/
çocuklarımïz	*our children*	/tS o - dZ u k- 5 a- r 1 -”m 1 z/
çocuklarımızïn	*of our children*	/tS o - dZ u k- 5 a- r 1 -m 1 z-”1 n/

odä	*room*	/”o d a/
odadä	*in the room*	/o - d a-”d a/
odadakï	*that which is in the room*	/o - d a-d a-”k i/
odadakilër	*those who are in the room*	/o - d a-d a-k i-”5 e r/
odadakilerdën	*from those who are in the room*	/o - d a-d a-k i-5 e r-”d e n/

Non-oxytone (exceptional) words keep the accent on the same syllable. These words are usually borrowings or place names:

tëyze	*aunt*	/" t e j - z e/
tëyzeniz	*your aunt*	/" t e j - z e - n i z/
tëyzenize	*to your aunt*	/" t e j - z e - n i - z e/

Some non-oxytone place-names :

Anädolu	*Anatolia*	/a - " n a - d o - 5 u/
İstänbul		/i s - " t a n - b u 5 /
Änkara		/" a n - k a - r a /

Many adverbs are stressed on the first syllable

şimdi	*now*	/" S i m - d i/
änsızın	*suddenly*	/a n- s 1 - z 1 n/

Polysyllabic suffixes (except the adverbial pre-stressing suffixes **-lAyIn** 'time adv.' and **-CAsInA** 'manner adv.') are stressed on the first syllable:

oku+yärak	*by reading*	/o - k u - " j a - r a k/
oku+yünca	*having read*	/o - k u - j u n - dZ a/

Interjections and vocatives are stressed on the first syllable:

garsön	*waiter*	/g a r - " s o n/
as contrasted with:		
gärson!	*waiter!*	/" g a r - s o n/

The shift of the stressed syllable can be clearly seen when the place names and common nouns have the same spelling:

mısïr	*maize*	/m 1 - "s 1 r/
Mïsır	*Egypt*	/"m 1 - s 1 r/
sirkecï	*vinegar seller*	/s i r - c e - "Z i/
S̈irkeci	*a district of İstanbul*	/"s i r - c e - Z i/
bebëk	*dolly, baby*	/b e - "b e c /
Bëbek	*a village of the Bosphorous*	/"b e - b e c /

4.3 Sentence intonation

In a regular sentence, the intonation peak is on the preverbal constituent of the sentence ([35], page 505). Intonation peak is located on the syllable that carries primary word stress for that preverbal constituent.

*Hasán bugűn istak**ó**z ye -di*
Hasan today lobster eat -Past
"Hasan ate (a) lobster(s) today"

A secondary, much lower intonation peak will be located on the subject, more specifically, on whichever syllable bears primary word level stress for the subject. Pitch drops immediately after the intonation peak [35]. The boldfaced vowel above carries the primary intonation peak, and the simple accent signed vowel, without boldface, bears the secondary intonation peak.

4.4 Designing and Recording of a Diphone Corpus

Diphones are the speech units that cover two sounds and the transition between them (Section 3.7). To collect the diphone database, carrier words for the diphones have to be designed. Carrier words can be chosen from sentences, pseudo-words or

real words. The advantage of choosing pseudo-words is that they can be constructed so as to minimize both the articulatory effort of the speaker and the coarticulation effects. While extracting a diphone set we have used nonsense words for the standard phonemes. We have collected all Vowel-Vowel, Consonant-Vowel, Consonant-Consonant, Vowel-Consonant, Consonant-Silence, Silence-Consonant, Vowel-Silence, Silence-Vowel diphone combinations. The extraction of the carrier words in a pseudo code is explained below.

- Carrier words for all possible vowel-vowel combinations

```
for v1 in vowels
  for v2 in vowels
     print t v1 v2 t
```

- Carrier words for all possible vowel-consonant and consonant-vowel combinations

```
for c1 in consonants
  for v1 in vowels
     print t v1 c1 v1 c1 v1
```

- Carrier words for all possible silence-consonant and consonant-silence combinations

```
for c1 in consonants
    print c1 a k a c1
```

- Carrier words for all possible silence-vowel and vowel-silence combinations

```
for v1 in vowels
    print v1 t v1
```

- Carrier words for all possible consonant-consonant combinations

```
for c1 in consonants
    for c2 in consonants
        print t a c1 a c2 a
```

However for the remaining allophonic phonemes we used real words so as to reflect the natural sounds. Diphones were recorded in a sound-proof recording studio. Many researchers recommend to record all diphones in the same day in order to minimize the variation in voice quality. However speaking all day strains the voice. Another alternative may be to spread recordings to several consecutive days scheduled at the same time. We have made our recordings in four distinct morning sessions. By recording at separate times, the speaker's performance could keep its quality.

4.5 Text Normalization

Text Normalization is one of first tasks of any TTS system. Text Normalization is the conversion of input text into linguistic representation. In Turkish, words are delimited by whitespaces. Therefore tokenization can be easily performed by using the word boundary information. After tokenization, every token must be tranformed into its correct linguistic representation.
Digit sequences need to be expanded into words:

```
1027 in Turkish would generally be expanded into:
bin     yirmi    yedi
thousand  twenty   seven
```

Abbreviations need to be expanded into full words:

```
PTT can be expanded to:
Pe Te Te
   or
Posta Telefon Telgraf
```

4.6 Designing the Lexicon

In this TTS system pronounciation lexicons are used for the transcription of the ortographic form of a word to pronounciation form. The pronounciations are encoded using the encoding in Tables 4.5 and 4.6, and also includes the marking of the position of the stress. Pronounciation lexicons consist of a list of phones and a

syllabic structure. Stressmark is typed in one of the syllabic positions. An example entry can be given as

```
(lex.add.entry
'("evin" nil ( ((E2)0) ((V1 I2 N1)1) )))
```

In addition to explicitly marking of syllables a stress value is given (`0,1 or 2`). `0` shows that there is no stress value for the corresponding syllable. In word level intonation, `1` value indicates that the associated syllable is stressed. Both `1` and `2` can be used in phrase level intonation where `1` is secondary stress and `2` is primary stress. Usually the basic assumption is that we will have a large lexicon, with tens of thousands of entries.

When a word is not listed, letter-to-sound rules are used to transform the ortographic form into a phonetic representation. But this has two disadvantages, first the syllabic stress is not shown (although we can stress the last syllable automatically). Second, the letters are tranformed into standard phones. For instance the *a* letter in the word *cahil* may be transformed into normal *a* letter and cause it to be synthesized in a wrong form. Also the syllabic stress information may get lost. Table 4.7 shows the mapping of the letters to phonemes.

ORTH. FORM	STANDARD PHONEME
a	A1
b	B1
c	C1
ç	C3
d	D1
e	E2
f	F1
g	G2
ğ	G3
h	H1
ı	I7
i	I2
j	J1
k	K1
l	L1
m	M1
n	N1
o	O1
ö	O5
p	P1
r	R2
s	S1
ş	S2
t	T1
u	U1
ü	U9
v	V1
y	Y1
z	Z2

Table 4.7: Letter to Sound Conversion Table

4.7 Designing the Intonation

Intonation in Festival is generated in two steps, prediction of accents and prediction of *F0*. Accents are placed on stressed syllables for Turkish. They are indicated in the lexicon. Constant *F0* values are set for the beginning, mid point and the end point of the accented syllable. A simple CART (Classification and Regression) tree is used for the *F0* prediction.

```
(set! int_accent_cart_tree
  '
  ((R:SylStructure.parent.gpos is content)
   ((stress is 1)
     ((Accented-Secondary))
      ((stress is 2)
       ((Accented-Primary))
    ((position_type is single)
     ((Accented-Secondary))
      ((NONE))))
       )
      ((NONE))))
```

The schematic view of the above Scheme Code is seen in Figure 4.1.

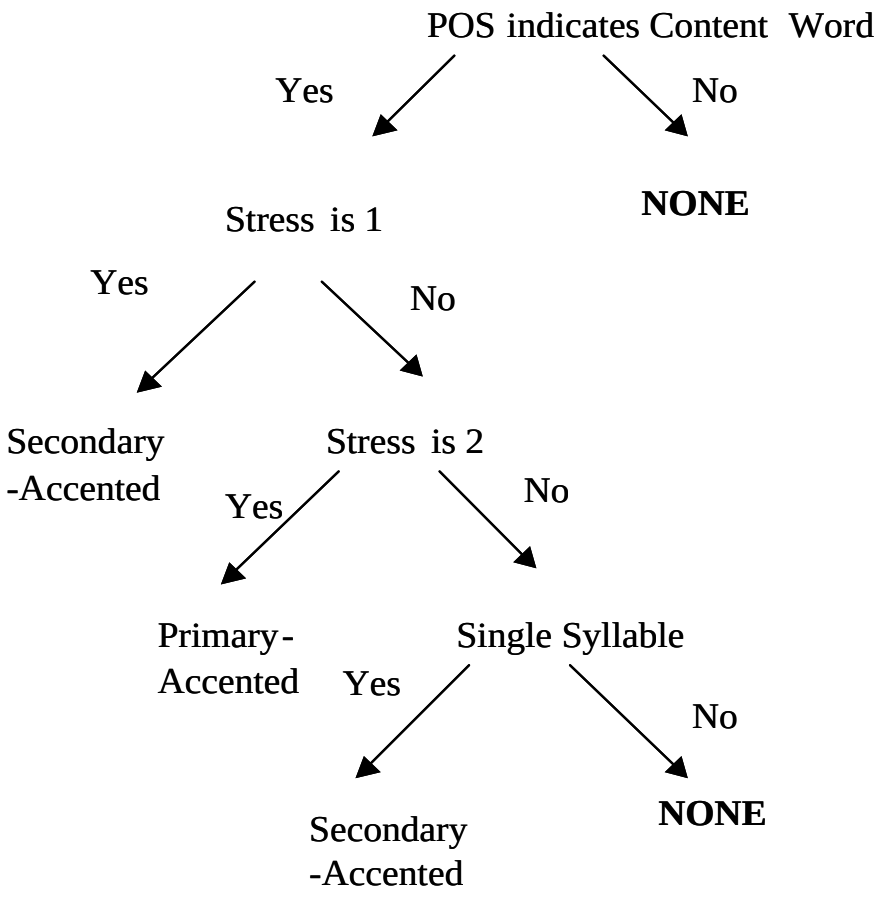

Figure 4.1: The schematic view of the CART tree for accent prediction

A LISP method that returns a list of target points for a syllable is called.

```
(define (targ_func1 utt syl )
 "F0 target points"
  (let ((start (item.feat syl 'syllable_start))
        (end (item.feat syl 'syllable_end))
        (ulen (item.feat (utt.relation.last utt 'Segment ) 'segment_end))
        nstart nend fustart fuend fuend fstart fend)
        (set! nstart (/ start ulen))
        (set! nend (/ end ulen))
        (set! fustart '130)
        (set! fuend   '110)
        (set! fstart  (+ (* (- fuend fustart) nstart) fustart))
       (set! fend    (+ (* (- fuend fustart) nend) fustart))

     (cond
  ((equal? (item.feat syl "R:Intonation.daughter1.name") "AccentedSecondary")
       (list
        (list start fstart)
        (list (+ start 0.010) (+ fstart 10 ))
        (list (- end   0.010) (+ fstart 8 ))
        (list end   fend)
   ))

   ((equal? (item.feat syl "R:Intonation.daughter1.name") "AccentedPrimary")
       (list
        (list start (- fstart 10) )
        (list (+ start 0.010) (+ fstart 40 ))
        (list (- end   0.010) (+ fstart 38 ))
        (list end  (- fend 10) )
   ))

      ((not (item.next syl))
```

```
 (list
 (list end fuend)))
 ((not (item.prev syl))
 (list
 (list start fustart)))
(t
 nil))))
```

This LISP function first initializes the speaker-dependent base *F0* values, namely `fustart` and `fuend`. These values 130, 110 Hz respectively model the slow declination of *F0* while speaking. `Start` denotes the start time of the syllable and `end` shows the end time of the syllable. `Ulen` is the duration time of the syllable that the *F0* target point will be attached. Four target points are attached for an accented syllable.

Nstart parameter is calculated as follows
$nstart = start/ulen$

Nend parameter is calculated as follows
$nend = end/ulen$

Fstart parameter is calculated as follows
fstart $= fustart + ((fuend - fustart) * nstart)$

Fend parameter is calculated as follows
fend $= fustart + ((fuend - fustart) * nend)$

We attach `fstart` Hz to the beginning of the syllable. And then an increased *F0* value[3] is attached at 0.01 milliseconds away from the start of the syllable. We attach approximately the same *F0* value[4] to the 0.01 milliseconds away of the end of the syllable. Lastly we attach `fend` value to the end of the syllable. The *F0* values should be changed for different speakers. This method allows some form of *F0* generation. This general technique may be expanded with other rules in the future.

4.8 Designing the Duration

Duration is very important for the quality and naturalness of a TTS system. In this TTS system, we have used the average duration model for the phones. Real data was not available for calculating the averages. Saylı has done a duration analysis for Turkish Text To Speech Systems [36]. He estimated the mean duration for vowels and consonants in the sentence environment. We used these values for the standard

[3] The increase is 10 Hz for AccentedSecondary syllables and 40 Hz for AccentedPrimary syllables.
[4] 2 Hz less.

phonemes and entered the remaining phoneme durations by hand. This method gave quite satisfactory results. Tables 4.8 and 4.9 show the durations of the allaphones in milliseconds.

PHONEME	DURATION (ms)
#	0.080
A2	0.102
A3	0.200
A5	0.132
A6	0.112
B1	0.054
C1	0.067
C2	0.104
C3	0.104
D1	0.047
E1	0.104
E2	0.114
E3	0.200
E4	0.140
F1	0.070
G2	0.100
G3	0.080
H1	0.062
I7	0.081
I2	0.092
I1	0.081
J1	0.073
K1	0.082
K2	0.082

Table 4.8: Duration of the allaphones in milliseconds [36].

PHONEME	DURATION (ms)
L1	0.060
L2	0.055
M1	0.071
N1	0.072
N1	0.072
N2	0.082
O1	0.109
O5	0.159
P1	0.085
R2	0.060
R3	0.060
R1	0.060
S1	0.111
S2	0.123
T1	0.078
U1	0.100
U3	0.085
U7	0.083
U9	0.103
V1	0.052
V2	0.060
Y1	0.070
Z2	0.080
Z1	0.080

Table 4.9: Duration of the allaphones in milliseconds [36].

Chapter 5

Conclusion and Further Research

5.1 Conclusion

A prosodic TTS system has been implemented for Turkish Language using the Festival Speech Synthesis Tool. This system differs from other TTS systems in a number of ways.

Firstly the number and type of phones used is extended. The set of allophones used in this system consists of 25 vowels and 29 consonants. There is a tradeoff between storage and increased naturalness. As the number of diphones increase, we need more memory, but naturalness of the system increases substantially. Despite this tradeoff, the system's performance was very good. Moreover, the use of contextual phones helped the system's output to be perceived more natural.

Secondly, duration model was useful in generating a more natural sounding system. Long and short vowels could be better distinguished. The durations are stretched for stressed syllables and this affect also helped the listener to realize where the word stress resides.

Lastly, prosody module helped to create an effective word stress. Word stress also contributed to an improved human perception of sentence stress. For the scope of the thesis some sentence and phrasal stress are also studied in less detail. However this part needs to be studied on extensively since there are many other elements affecting sentence prosody. But for now, primary stress and secondary stress for a sentence can be given. The result is quite satisfactory.

5.2 Further Research

In further work, the number of contextual phonemes may be increased. Triphones may be used for units.

As a further research, the sentence and phrasal prosody can be further studied. The target F0 function can be expanded with other rules. TOBI and TILT labellings can be exploited. Training-based F0 generation could also be tried.

Duration modelling could be performed on a contextual basis. Other than giving fixed average duration for the phonemes. The duration of the phonemes could change depending on their context.

Bibliography

[1] Dutoit, T., High-Quality Text-to-Speech Synthesis : an Overview, Journal of Electrical & Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis, vol. 17 n1, pp. 25-37.

[2] Dutoit, T., Pagel V., Pierret N., Bataille F., Vrecken O. 1996. The Mbrola Project: Towards a Set of High Quality Speech Synthesizers Free of Use for NonCommercial Purposes. Proceedings of ICSLP 96 (3).

[3] Veronis, J., Hirst, D., Espesser, R., Ide, N., 1994. NL and speech in the Multext project, AAAI'94 Workshop on Integration of Natural Language and Speech

[4] Bozkurt, B., Dutoit, T., 2001. An implementation and evaluation of two-diphone based synthesizers for Turkish", Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis,pp.247-250, Blair Atholl, Scotland

[5] Sproat, R., Olive, J., 1995. An Approach to Text-to-Speech Synthesis," in Speech Coding and Synthesis, W.B. Kleijn and K.K. Paliwal, Amsterdam, pp. 611-634, Elsevier Science

[6] Kupiec, J., 1992. Robust part-of-speech tagging using a Hidden Markov Model, Computer Speech and Language, n6, pp. 225-242.

[7] Sauerbrey, G.Z., 1959. Z, Phys.,155,206-222

[8] Hunnicut, S., Grapheme-to-Phoneme rules : a Review, Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR 2-3, pp. 38-60.

[9] Allen, J., Hunnicut, S., Klatt, D., 1987. From Text To Speech, The MITTALK System, Cambridge University Press, 213 pp.

[10] Levinson, S.E., Olive, J.P., Tschirgi, J.S, Speech Synthesis in Telecommunications, IEEE Communications Magazine, November 1993, pp 510-513.

[11] Coker, C.H., Church, K.W., Liberman, M.Y., 1990. "Morphology and rhyming : Two powerful alternatives to letter-to-sound rules for speech synthesis", Proc. of the ESCA Workshop on Speech Synthesis, Autrans (France), pp 83-86.

[12] Sproat, R., Olive, J. 1995. "An approach to Text-To-Speech Synthesis," in Speech Coding and Synthesis, W.B. Kleijn and K.K. Paliwal, eds., Amsterdam, pp. 611-634, Elsevier Science

[13] Laver, J. 1994. Principles Of Phonetics, Cambridge University Press, Cambridge.

[14] Oflazer, K., Inkelas, S., A Finite State Pronounciation Lexicon for Turkish, in Proceedings of the EACL Workshop on Finite State Methods in NLP, April 13-14, 2003, Budapest, Hungary

[15] Klatt, D. 1979. Synthesis by rule of segmental durations in English sentences, in B. Lindblom & S. hmann, eds, 'Frontiers of Speech Communication Research', pp. 287-300.

[16] Van Santen, J. 1993. Timing in text-to-speech synthesis, in 'Proc. Eurospeech'.

[17] Black, A. & Hunt, A. 1996. Generating F0 Contours from TOBI labels using linear regression, in 'Proc. ICSLP'.

[18] Cruttenden, A. 1987. Intonation, Cambridge University Press.

[19] Pierrehumbert, J. 1980. The Phonology and Phonetics of English Intonation, PHD thesis, Department of Linguistics, MIT

[20] Anderson, M., 1984. Pierrehumbert, J., and Liberman, M. Synthesis by rule of English intonation patterns. In Proceedings of ICASSP 84, pages 2.8.1-2.8.4

[21] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M.,Wightman, C., Price, P., Pierrehumbert, J. & Hirschberg, J., 1992. TOBI: A standard for labelling English prosody, in 'Proc. ICSLP'.

[22] Hunnicut, S., Grapheme-to-Phoneme rules : a Review , Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR 2-3, pp. 38-60.

[23] Price, P., Ostendorf, M., Shattuck-Hufnagel, S. & Fong, C., 1991. The use of prosody in syntactic disambiguation, JASA 90, 2956-2969

[24] Libermann, M.J., Church, K.W., Text analysis and word pronounciation in text-to-speech synthesis system, Proc. Eurospeech 89, Paris, pp. 510-513

[25] Stevens, K.N. , Control Parameters for synthesis by rule, Proceedings of the ESCA tutorial day for speech synthesis, Autrans, 25 sept 90, pp. 27-37.

[26] Harris, C. , 1953. A study of the building blocks of speech, JASA 25, 962-969

[27] Hunt, M. Zwierynski, D., Carr R. ,1989. Issues in high quality LPC analysis and synthesis. In Eurospeech 89, volume 2, pages 348-351, Paris, France

[28] Hunt, A., Black, A., 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In ICASSP-96, volume 1, pages 373-376, Atlanta, Georgia

[29] Moulines, E. , Charpentier, F. 1990. , Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication, 9(5/6):453-467

[30] Black,A., Taylor,P., 1997. Automatically clustering similar units for unit selection in speech synthesis. In Eurospeech97, volume2, pages 601-604, Rhodes, Greece

[31] Black,A., Lenzo,K., Building voices in the Festival Speech Synthesis System, Carnegie Mellon University

[32] Donovan,R., Woodland,P. 1995., Improvements in an HMM-based speech synthesiser. In Eurospeech95, volume 1, pages 573-576, Madrid, Spain,

[33] Taylor, P.A. ,1998. The Tilt Intonation Model, Proc. Int. Conf. on Spoken Language Processing, Sydney, Australia

[34] Clements, G.N., Sezer, E., 1982. Vowel and Consonant Disharmony in Turkish. In: H. van der Hulst & N. Smith (eds.), The Structure of Phonological Representation. Part ——. Dordrecht: Foris 213-255

[35] Kornfilt, J., 1997. Turkish. (Descriptive Grammars) London: Routledge, 1999. Turkic Languages 3, 131-142.

[36] Saylı, Ö., Duration Modelling, Bosphorous University, MS Thesis

[37] Huang X., Acero A., Hon H.W, Spoken Language Processing, Prentice Hall, Upper Saddle River, NJ, 2001

Bibliography

[1] Dutoit, T., High-Quality Text-to-Speech Synthesis : an Overview, Journal of Electrical & Electronics Engineering, Australia: Special Issue on Speech Recognition and Synthesis, vol. 17 n1, pp. 25-37.

[2] Dutoit, T., Pagel V., Pierret N., Bataille F., Vrecken O. 1996. The Mbrola Project: Towards a Set of High Quality Speech Synthesizers Free of Use for NonCommercial Purposes. Proceedings of ICSLP 96 (3).

[3] Veronis, J., Hirst, D., Espesser, R., Ide, N., 1994. NL and speech in the Multext project, AAAI'94 Workshop on Integration of Natural Language and Speech

[4] Bozkurt, B., Dutoit, T., 2001. An implementation and evaluation of two-diphone based synthesizers for Turkish", Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis,pp.247-250, Blair Atholl, Scotland

[5] Sproat, R., Olive, J., 1995. An Approach to Text-to-Speech Synthesis," in Speech Coding and Synthesis, W.B. Kleijn and K.K. Paliwal, Amsterdam, pp. 611-634, Elsevier Science

[6] Kupiec, J., 1992. Robust part-of-speech tagging using a Hidden Markov Model, Computer Speech and Language, n6, pp. 225-242.

[7] Sauerbrey, G.Z., 1959. Z, Phys.,155,206-222

[8] Hunnicut, S., Grapheme-to-Phoneme rules : a Review, Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR 2-3, pp. 38-60.

[9] Allen, J., Hunnicut, S., Klatt, D., 1987. From Text To Speech, The MITTALK System, Cambridge University Press, 213 pp.

[10] Levinson, S.E., Olive, J.P., Tschirgi, J.S, Speech Synthesis in Telecommunications, IEEE Communications Magazine, November 1993, pp 510-513.

[11] Coker, C.H., Church, K.W., Liberman, M.Y., 1990. "Morphology and rhyming : Two powerful alternatives to letter-to-sound rules for speech synthesis", Proc. of the ESCA Workshop on Speech Synthesis, Autrans (France), pp 83-86.

[12] Sproat, R., Olive, J. 1995. "An approach to Text-To-Speech Synthesis," in Speech Coding and Synthesis, W.B. Kleijn and K.K. Paliwal, eds., Amsterdam, pp. 611-634, Elsevier Science

[13] Laver, J. 1994. Principles Of Phonetics, Cambridge University Press, Cambridge.

[14] Oflazer, K., Inkelas, S., A Finite State Pronounciation Lexicon for Turkish, in Proceedings of the EACL Workshop on Finite State Methods in NLP, April 13-14, 2003, Budapest, Hungary

[15] Klatt, D. 1979. Synthesis by rule of segmental durations in English sentences, in B. Lindblom & S. hmann, eds, 'Frontiers of Speech Communication Research', pp. 287-300.

[16] Van Santen, J. 1993. Timing in text-to-speech synthesis, in 'Proc. Eurospeech'.

[17] Black, A. & Hunt, A. 1996. Generating F0 Contours from TOBI labels using linear regression, in 'Proc. ICSLP'.

[18] Cruttenden, A. 1987. Intonation, Cambridge University Press.

[19] Pierrehumbert, J. 1980. The Phonology and Phonetics of English Intonation, PHD thesis, Department of Linguistics, MIT

[20] Anderson, M., 1984. Pierrehumbert, J., and Liberman, M. Synthesis by rule of English intonation patterns. In Proceedings of ICASSP 84, pages 2.8.1-2.8.4

[21] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M.,Wightman, C., Price, P., Pierrehumbert, J. & Hirschberg, J., 1992. TOBI: A standard for labelling English prosody, in 'Proc. ICSLP'.

[22] Hunnicut, S., Grapheme-to-Phoneme rules : a Review , Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR 2-3, pp. 38-60.

[23] Price, P., Ostendorf, M., Shattuck-Hufnagel, S. & Fong, C., 1991. The use of prosody in syntactic disambiguation, JASA 90, 2956-2969

[24] Libermann, M.J., Church, K.W., Text analysis and word pronounciation in text-to-speech synthesis system, Proc. Eurospeech 89, Paris, pp. 510-513

[25] Stevens, K.N. , Control Parameters for synthesis by rule, Proceedings of the ESCA tutorial day for speech synthesis, Autrans, 25 sept 90, pp. 27-37.

[26] Harris, C. , 1953. A study of the building blocks of speech, JASA 25, 962-969

[27] Hunt, M. Zwierynski, D., Carr R. ,1989. Issues in high quality LPC analysis and synthesis. In Eurospeech 89, volume 2, pages 348-351, Paris, France

[28] Hunt, A., Black, A., 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In ICASSP-96, volume 1, pages 373-376, Atlanta, Georgia

[29] Moulines, E. , Charpentier, F. 1990. , Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication, 9(5/6):453-467

[30] Black,A., Taylor,P., 1997. Automatically clustering similar units for unit selection in speech synthesis. In Eurospeech97, volume2, pages 601-604, Rhodes, Greece

[31] Black,A., Lenzo,K., Building voices in the Festival Speech Synthesis System, Carnegie Mellon University

[32] Donovan,R., Woodland,P. 1995., Improvements in an HMM-based speech synthesiser. In Eurospeech95, volume 1, pages 573-576, Madrid, Spain,

[33] Taylor, P.A. ,1998. The Tilt Intonation Model, Proc. Int. Conf. on Spoken Language Processing, Sydney, Australia

[34] Clements, G.N., Sezer, E., 1982. Vowel and Consonant Disharmony in Turkish. In: H. van der Hulst & N. Smith (eds.), The Structure of Phonological Representation. Part ——. Dordrecht: Foris 213-255

[35] Kornfilt, J., 1997. Turkish. (Descriptive Grammars) London: Routledge, 1999. Turkic Languages 3, 131-142.

[36] Saylı, Ö., Duration Modelling, Bosphorous University, MS Thesis

[37] Huang X., Acero A., Hon H.W, Spoken Language Processing, Prentice Hall, Upper Saddle River, NJ, 2001