IMPLEMENTATION OF A TEXT-TO-SPEECH SYSTEM WITH MACHINE LEARNING ALGORITHMS IN TURKISH

by

Zeliha GÖRMEZ

July 2009

IMPLEMENTATION OF A TEXT-TO-SPEECH SYSTEM WITH MACHINE LEARNING ALGORITHMS IN TURKISH

by

Zeliha GÖRMEZ

A thesis submitted to

The Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

July 2009
Istanbul, Turkey

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Assist. Prof. Tuğrul YANIK
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Assist. Prof. Zeynep ORHAN
Supervisor

Examining Committee Members

Assist. Prof. Zeynep ORHAN __________________________

Assist. Prof. Tuğrul YANIK __________________________

Assist. Prof. Nurgül ÖZCAN __________________________

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

____

Assoc. Prof. Nurullah ARSLAN
Director

IMPLEMENTATION OF A TEXT-TO-SPEECH SYSTEM WITH MACHINE LEARNING ALGORITHMS IN TURKISH

Zeliha GÖRMEZ

M. S. Thesis - Computer Engineering
July 2009

Supervisor: Assist. Prof. Zeynep ORHAN

ABSTRACT

This study is intended to build the framework of a concatenative TTS (Text to Speech) system for Turkish. Turkish TTS system is based on concatenative, unit selection approach. System contains two different speech databases comprised of units which are directly recorded and cut from a continuous speech. The units have been cut from speech manually and automatically. Some digital signal features such as zero crossing rate and energy of speech have been used for automatic cutting. While concatenating the units, PSOLA (Pitch Synchronous Overlap and Add) algorithm has been used for smoothing.

Some subjective tests are used to measure the system success. The quality of the synthesized speech is measured depending on two criteria: Intelligibility and naturallness. For naturalness defined as closeness to human speech, Mean Opinion Score (MOS), for intelligibility defined as the ability to be understood, Diagnostic Rhyme Test (DRT) and Comprehension Test (CT) have been applied.

Although the system uses simple techniques, it provides promising results for Turkish TTS, since the selected concatenative method is very well suited for Turkish language structure.

Keywords: Turkish TTS System, Concatenative Synthesis, Unit Selection, MOS, DRT, CT, PSOLA.

MAKİNE ÖĞRENME ALGORİTMALARYLA TÜRKÇE METİN SESLENDİRME SİSTEMİ YAZILIMI

Zeliha GÖRMEZ

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Temmuz 2009

Tez Yöneticisi Yrd. Doç.Dr. Zeynep ORHAN

ÖZ

Bu çalışma Türkçe için eklemeli metin seslendirme sistemi oluşturmak amacındadır. Türkçe metin seslendirme sistemi eklemeli birim seçme yaklaşımına dayanmaktadır. Sistem direk kaydedilen ve sürekli bir konuşmadan kesilen birimlerden oluşan iki farklı ses veritabanına sahiptir. Birimler konuşmadan elle ve otomatik olarak kesilmiştir. Ses sinyalinin sıfırdan geçiş sayısı ve sesin enerjisi gibi sinyal özellikleri otomatik kesme işlemi için kullanılmıştır. Birleştirme işleminde sesler arasında yumuşak geçişler için Perde Senkronize Üstüste Ekleme (PSOLA-Pitch Synchronous Overlap-Add) algoritması kullanılmıştır.

Sistem başarısının ölçülmesi için bir takım öznel testler kullanılmaktadır. Sistemde üretilen seslerin kalitesi iki noktaya bağlı olarak ölçülmüştür: Anlaşılabilirlik ve doğallık. İnsan sesine yakınlık olarak tanımlanan doğallık için MOS testi, anlaşılabilirlik için ise ahenk testi (DRT) ve kavrama testi (CT) uygulanmıştır.

Sistem basit teknikler kullanıyor olmasına rağmen, seçilen eklemeli method Türkçe'nin yapısına çok uygun olduğu için ümit verici sonuçlar üretmektedir.

Anahtar Kelimeler: Türkçe Metin Seslendirme, Eklemeli Sentezleme, MOS, DRT, CT, PSOLA.

DEDICATION

To my family and dear husband

ACKNOWLEDGEMENT

I owe a special debt of gratitude to Assist. Prof. Dr. Zeynep ORHAN, whose guidance and critical advice, as my thesis advisor, greatly contributed to the making of this study.

I gratefully would like to acknowledge my deepest appreciation to my dear friend Emine DOĞANAY for her invaluable help in the appilications with digital signal processing and to Rüya ŞAMLI and Saime AKDEMİR for their checking the quality of English. I especially would like to thank to my colleague Vildan ÖZDEMİR for her moral support and my chair Günay KARLI for his facilitative attitutes and trust.

I would like to express my special thanks to Assist. Prof. Dr Tuğrul YANIK and Assist. Prof. Dr. Nurgül ÖZCAN for their kindly accepting to be a member of the committee.

Lastly, but certainly not least, I would like to thank to my husband, Birol GÖRMEZ, who helped me immeasurably with his encouragement, and kept me going at the most desperate moments.

TABLE OF CONTENTS

ABSTRACT III

ÖZ IV

DEDICATION V

ACKNOWLEDGEMENT VI

TABLE OF CONTENTS VII

LIST OF TABLES X

LIST OF FIGURES XII

LIST OF SYSMBOLS AND ABBREVIATIONS XIII

CHAPTER 1 1

INTRODUCTION 1

1.1 GENERAL PURPOSE 1

1.2 NATURAL LANGUAGE PROCESSING 1

1.2.1 Natural Language Processing Applications 2

1.2.2 Approaches to Natural Language Processing 4

1.3 TEXT TO SPEECH 5

1.3.1 Speech 6

1.3.2 Modules of Text-to-Speech 7

1.3.2.1 NLP Module 7

1.3.2.2 DSP Module 8

1.3.3 Technologies of Speech Synthesis 11

1.3.3.1 Concatenative Synthesis 11

1.3.3.2 Formant Synthesis 13

1.3.3.3 Articulatory Synthesis....13
1.3.4 Application Area of Text-to-Speech....14
1.3.5 Major Text-to-Speech Applications....15
1.3.5.1 ASY 16
1.3.5.2 MITalk and DECTalk 16
1.3.5.3 INFOVOX 16
1.3.5.4 Bell Labs TTS....17
1.3.5.5 CNET PSOLA 17
1.3.5.6 CHATR....18
1.3.5.7 FESTIVAL....18
1.3.5.8 MBROLA 18
1.3.5.9 WHISTLER 19
1.3.5.10 Text-to-Speech for Turkish....19
1.3.6 Problems of Text-to-Speech 21
1.3.6.1 Text Processing Problems....21
1.3.6.2 Speech Processing Problems 24
CHAPTER 2 27
TURKISH TEXT-TO-SPEECH SYSTEM ARCHITECTURE....27
2.1 TURKISH LANGUAGE PROPERTIES....28
2.1.1 Turkish Phoneme Set....28
2.1.2 Structure of Turkish Syllables 31
2.2 SPEECH DATABASE 32
2.3 RECOGNIZE AND CLASSIFY PHONEMES AUTOMATICALLY 34
CHAPTER 3 40
IMPLEMENTATION OF SPEECH SYNTHESIS 40
3.1 TEXT-TO-PHONEME / NLP MODULE....40

3.2 PHONEME-TO-SPEECH / DSP MODULE 46
3.3 PROGRAM INTERFACE...47
CHAPTER 4 49
EVALUATION TESTS...49
4.1 MEAN OPINION SCORE...49
4.2 COMPREHENSION TEST 52
4.3 DIAGNOSTIC RHYME TEST 54
CHAPTER 5 56
CONCLUSION...56
REFERENCES 59

LIST OF TABLES

TABLE

1.1 Major TTS Applications 15
1.2 Example of Semiotic Classes 22
1.3 Semiotic Class of Tokens and Translation 23
2.1 Letters and Phonemes in Turkish 30
2.2 The Structure of Turkish Syllables 31
2.3 Diphones Kept in Turkish Speech Database 33
2.4 CV-VC Syllable Combinations 33
2.5 Synthesized Form Example 34
2.6 Confusion Matrix 38
2.7 Train and Test Accuracy Result 39
2.8 Presicion and Recall Results 39
3.1 Pseudocode for Main Function of Text-to-Phoneme 41
3.2 Pseudocode of Sub-Functions 41
3.3 Vocalized and Not Vocalized Special Characters 43
3.4 Possible Combinations of Two Consecutive Consonants at the End and Beginning of a Syllable in Turkish (1=Possible, 0=Impossible or Ignored) 44
3.5 Examples of Possible Combinations of Two Consecutive Consonants at the End and Beginning of a Syllable in Turkish(N/A=Not Applicable) 45
4.1 Mean Opinion Score 50
4.2 Test Sentences 50
4.3 MOS Range and Scores for Each Sentence 51
4.4 Listening Comprehension 53
4.5 Answering Common Questions 53

4.6 Filling in the Blanks..54
4.7 Words for DRT 55
4.8 Comprehension Tests and DRT Accuracy...55

LIST OF FIGURES

FIGURE

1.1 Positions of NLP and TTS in the Computer Science Taxonomy 2
1.2 Processes of an NLP application 4
1.3 Position of TTS and ASR in Computer Human Communication....................................5
1.4 Human Speech Production Process 6
1.5 Waveform of Speech Signal 7
1.6 Basic Block Diagram of TTS...8
1.7 SOLA Time Scale 10
1.8 PSOLA Synthesis 10
1.9 Block Diagram of the Concanative TTS System...11
1.10 Flexibility and Intelligibility Variation According to Unit Length 13
1.11 A screenshot of OKU 4.0..20
1.12 A screenshot of GVZ 21
1.13 Prosodic Dependencies...26
2.1 Zemberek Screenshot..28
2.2 Turkish TTS System Architecture 29
2.3 The Percentages of Various Syllable Lengths in Turkish...32
2.4 Autocorrelation Voiced and Unvoiced Speech...36
2.5 Correlation of Energy and ZCR Value of Each Frame in The Word *kardeş*.................37
3.1 First Program Interface 48
3.2 Second Program Interface...48
4.1 Distribution of MOS Values for Test Sentences Assigned by Testers 51
4.2 Average MOS Values for Each Sentence and System Average 52

LIST OF SYSMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

Acc	Accuracy
ANSI	American National Standards Institute
ARPA	Advanced Research Projects Agency
ASR	Automatic Speech Recognition
C	Consonant
CCend	Two Consecutive Consonants at the End of a Syllable
CCbegin	Two Consecutive Consonants at the Beginning of a Syllable
CT	Comprehension Test
DM	Dialog Manager
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processing
N	Noise
ITU	International Telecommunication Union
MOS	Mean Opinion Score
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
P	Presicion
PSOLA	Pitch Synchronous Overlap and Add
R	Recall
S	Silence
SOLA	Synchronous Overlap-Add method
TD-PSOLA	Time Domain- Pitch Synchronous Overlap and Add

TIMIT	Texas Instruments/Massachusetts Institute of Technology
TTS	Text-to-Speech
WEKA	Waikato Environment for Knowledge Analysis
V	Vowel
ZCR	Zero Crossing Rate

CHAPTER 1

INTRODUCTION

1.1 GENERAL PURPOSE

Text-to-Speech (TTS) is the technology which lets computer speak to you. The TTS gets the text as an input and then a computer algorithm called TTS engine analyzes the text, preprocesses the text and synthesizes the speech with some mathematical models. The TTS engine usually gives kind of sound data like wave, mp3 etc as an output.

TTS is under Natural Language Processing (NLP) in computer science taxonomy. To understand TTS better, first NLP should be understood. So, in the following two sections of introduction, NLP and TTS will be investigated extensively.

1.2 NATURAL LANGUAGE PROCESSING

NLP is a field of computer science concerned with the interactions between computers and human (natural) languages.

NLP is a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages[1]. To summarize, NLP is to teach computers how humans learn and use language and how to speak to humans in their natural language.

[1] NLP: http://www.webopedia.com/TERM/N/NLP.html

NLP is a branch of artificial intelligence. Position of NLP in computer science and position of TTS in NLP are shown in Figure 1.1

NLP is also known as Computational Linguistics[2] , Human Language Technology[3] , and Natural Language Engineering.

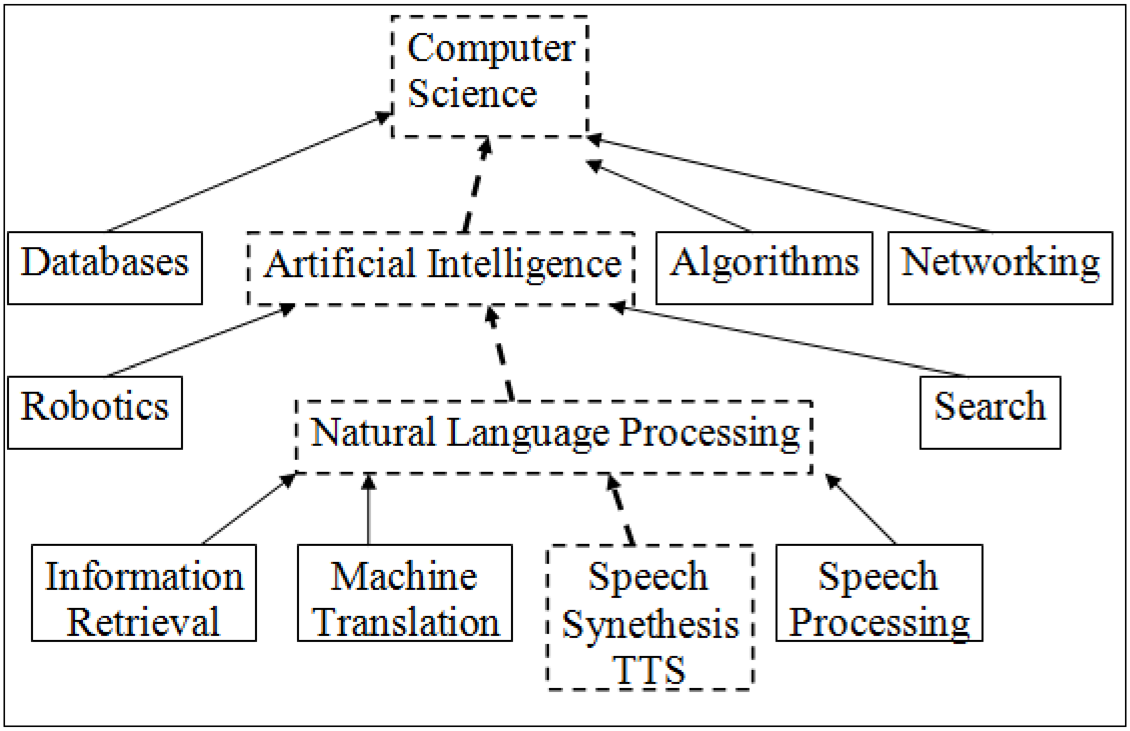


Figure 1.1 Positions of NLP and TTS in the Computer Science Taxonomy

1.2.1 Natural Language Processing Applications

There are many NLP applications. Some of these applications are text-to-speech, speech recognition, information retrieval, information extraction, document classification, automatic summarization, text proofreading – spelling & grammar, machine translation, story understanding systems, plagiarism detection etc.

An NLP application and the processes of the NLP are illustrated in Figure 1.2. In this figure, there are two main processeses: Speech processing and text processing in NLP.

[2] Computational Linguistics: http://en.wikipedia.org/wiki/Computational_linguistics

[3] Human Language Technology: http://en.wikipedia.org/wiki/Human_Language_Technology

In the speech processing there are two parts:

- Automatic Speech Recognition (ASR)
- Text-to-Speech (TTS)

In the text processing there are three parts:

- Natural Language Understanding (NLU)
- Dialog Manager (DM)
- Natural Language Generation (NLG)

Description of Figure 1.2 is belowIt is assumed that there is a human on one side and a computer on the other side.

Firstly, the human says "*what exercise should I do now*". This speech is taken as an input by the computer. It is also the input of ASR part. ASR system tokenizes the speech and converts it into text and this text is the output of this step. This output is also the input of text processing part.

In the text processing part, firstly, NLU part takes this input and tries to understand what it says. Secondly, the computer decides how to reply to this input with the help of knowledge representation. The computer asks itself some questions such as *"what does the exercise mean?"* What do I now know about the types of exercise? And so on. By making use of its stored sources such as dictionary or a text corpus, it tries to give response to the questions. At the end of DM part, computer decides to suggest the human run. Thirdly, in the NLG part, the computer generates a grammatically and structurally correct sentence to give response the human. The output of text processing part is naturally a text.

Finally, in the TTS part text processing output is converted into speech. And this speech is also evaluated as the output of computer. The computer says: '*I suggest you run five kilometers'*.

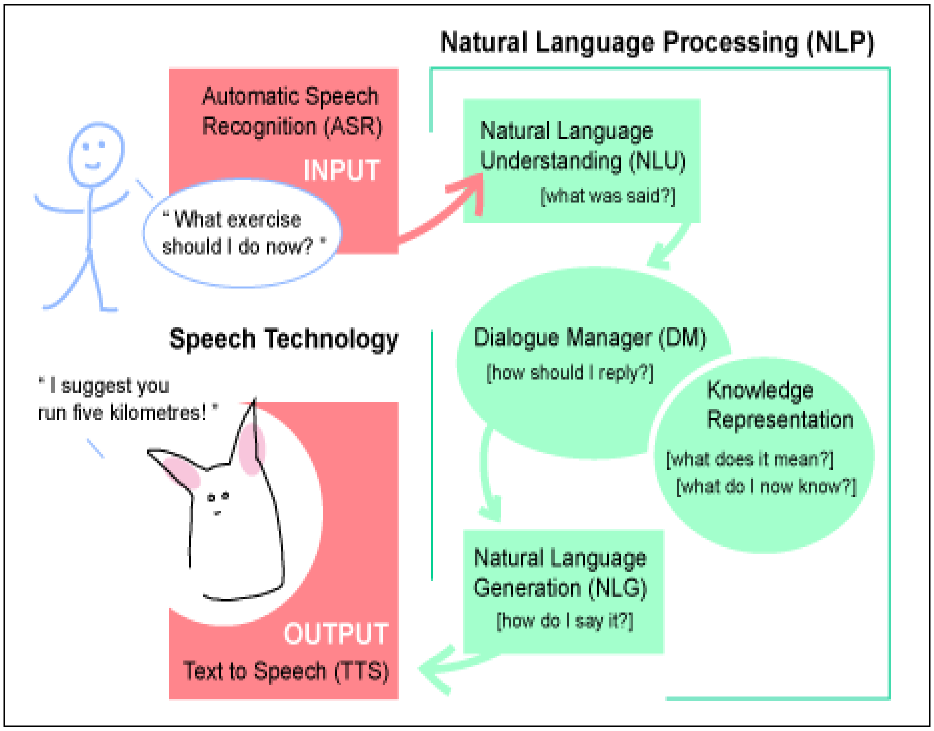

Natural Language Processing (NLP)
Automatic Speech Recognition (ASR)
INPUT
" What exercise should I do now? "
Natural Language Understanding (NLU)
[what was said?]
Speech Technology
Dialogue Manager (DM)
[how should I reply?]
Knowledge Representation
[what does it mean?]
[what do I now know?]
" I suggest you run five kilometres! "
Natural Language Generation (NLG)
[how do I say it?]
OUTPUT
Text to Speech (TTS)

Figure 1.2 Processes of an NLP application[4]

1.2.2 Approaches to Natural Language Processing

NLP approaches are divided into two categories: symbolic (rule-based) and statistical. Rule-based systems usually consist of a set of rules, an inference engine, and a workspace or working memory. Knowledge is represented as facts or rules in the rule-base. The inference engine repeatedly selects a rule whose condition is satisfied and executes the rule (Liddy, 2003). Symbolic approaches have been used for information extraction, text categorization and ambiguity resolution. Statistical approaches employ various mathematical techniques and often use large text corpora to solve problems in tasks such as speech recognition, speech synthesis, parsing and part-of-speech tagging.

[4] Companions Research: http://www.companions-project.eu/research/

1.3 TEXT TO SPEECH

Speech synthesis technology refers to the knowledge of producing the artificial sounds that will be interpreted as speech that can be possibly strange but yet understandable (Maxey, 2002). The human speech can be produced artificially either in software or in hardware as the ultimate goal of the speech synthesis. The natural language text is converted into speech by TTS systems as a subbranch of speech synthesis. Building a system that clearly gets across the message and achieving this by using a human-like voice are the fundamental concerns of a computer system capable of speaking.

This study is about getting computers to read out loudly. It is therefore about three things: The process of reading, the process of speaking, and the issues involved in getting computers (as opposed to humans) to do this. This field of study is known both as speech synthesis that is the “synthetic” (computer) generation of speech, and text-to-speech or TTS; the process of converting written text into speech (Taylor, 2007). The aim of the TTS is that the system converts all digital texts and printed texts via optical character recognition into speech automatically.

TTS contains speech processing but not speech recognition. Someone can think TTS as *talking* and ASR as *listening*[5]. Figure 1.3 shows the position of TTS and ASR in computer human communication.

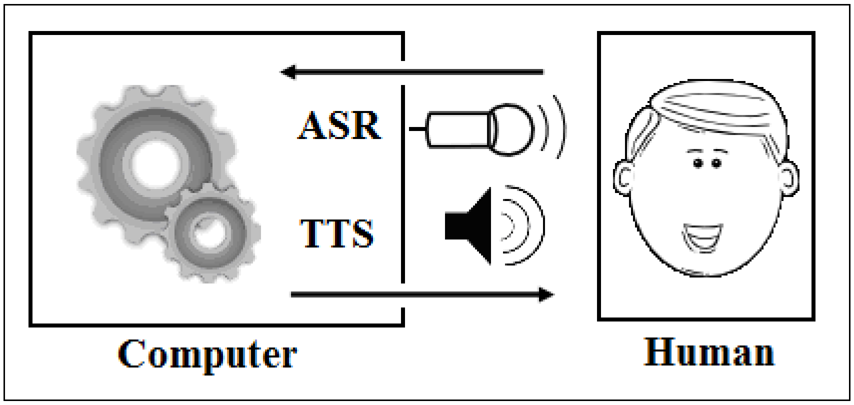


Figure 1.3 Position of TTS and ASR in Computer Human Communication

[5] AT&T Labs, FQA: http://www.research.att.com/~ttsweb/tts/faq.php#TechWhat

1.3.1 Speech

Speech is the vocalization form of human communication. It is based upon the syntactic combination of lexicals and names that are drawn from very large (usually>10,000 different words) vocabularies. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech sound units[6].

The process of human speech production is shown by Figure 1.4. The mechanism of speech is composed of four processes: Language processing, in which the content of an utterance is converted into phonemic symbols in the brain's language center; generation of motor commands to the vocal organs in the brain's motor center; articulatory movement for the production of speech by the vocal organs based on these motor commands; and the emission of air sent from the lungs in the form of speech (Honda 2003). The air formed by the vocal organs after emitting from the lungs constitutes the speech signal, which is a continuous, acoustic waveform. It is created by the operation of the vocal organs in response to motor control commands from the brain (Taylor, 2007). In Figure 1.5 speech signal waveform is shown.

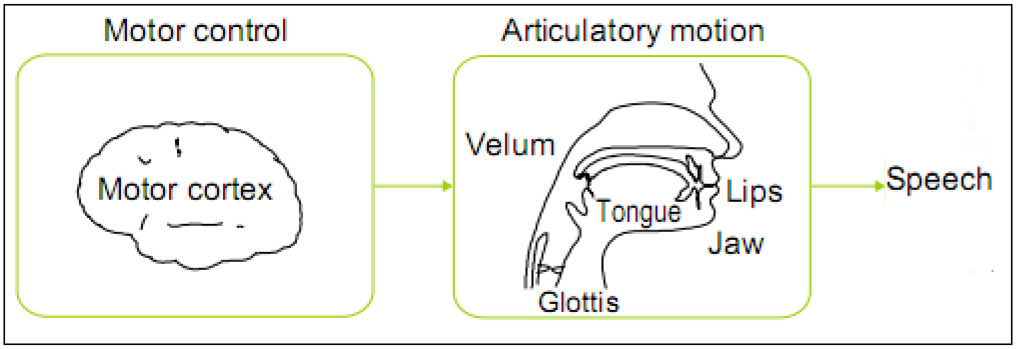


Figure 1.4 Human Speech Production Process (Honda, 2003)

[6] Speech: http://en.wikipedia.org/wiki/Speech

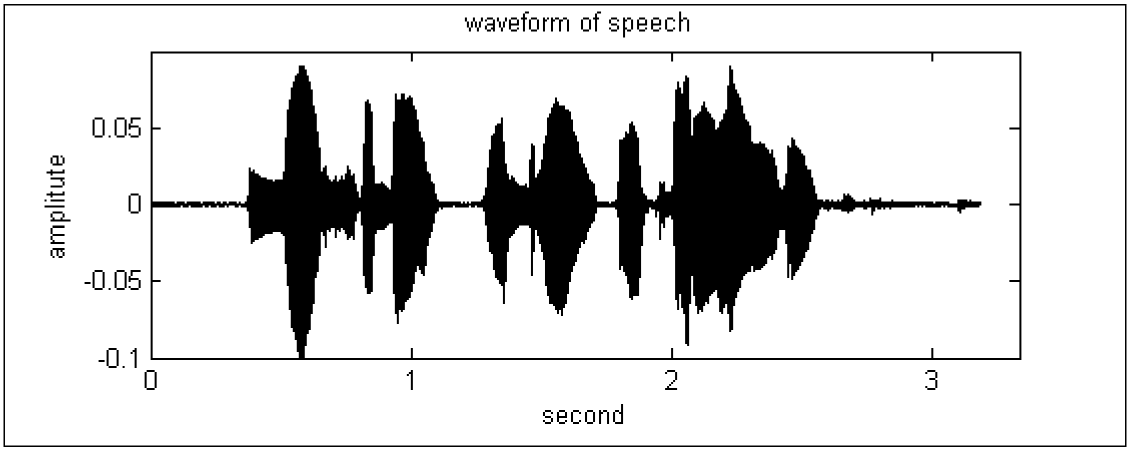


Figure 1.5 Waveform of Speech Signal

1.3.2 Modules of Text-to-Speech

There are two major modules in a TTS system. One of them is Natural Language Processing (NLP) Module that converts the written text input in phonetic transcription. The other one is Digital Signal Processing (DSP) Module that transforms the symbolic information into speech (Cerrato, 2005) (Taylor 2007). Block diagram of TTS is shown in Figure 1.6. TTS software has little or no *understanding* of the text being read. It uses rules, lists, dictionaries, etc. to make very sophisticated guesses about how a piece of text should be read[7]

1.3.2.1 NLP Module

This step called *high-level*, *front-end or text-to-phoneme*. This module contains text normalization, text analysis and language analysis.

With high-level synthesis the input text or information is transcribed in such format that low-level voice synthesizer is capable to produce the acoustic output (Lemmety, 1999).

Text normalization step subsumes sentence segmentation, tokenizing, and normalization of nonstandard words such as date, number etc. Sentence segmentation is

[7] AT&T Labs, FQA: http://www.research.att.com/~ttsweb/tts/faq.php#TechWhat

end-of-sentence detection by period or other sentence end punctuation marks. Tokenizing is achieved by splitting the text at white spaces and at punctuation marks (Reichel and Pfitzinger, 2006). NLP module processes are detailed in the section of text processing problems extensively.

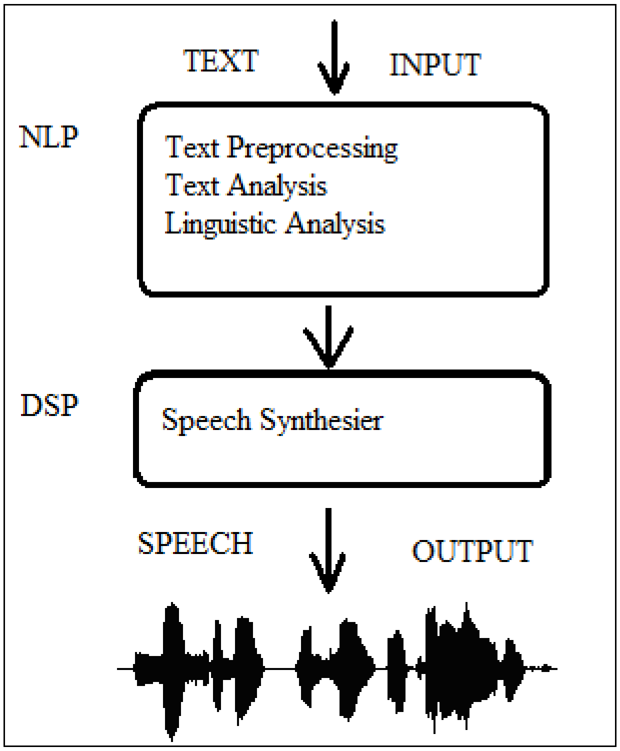


Figure 1.6 Basic Block Diagram of TTS

1.3.2.2 DSP Module

In literature, the terms of *low-level*, *back-end and phoneme-to-speech* are also commonly used to refer to this step in speech synthesis. A low-level synthesizer is the actual device which generates the output sound from information provided by high-level device in some format, for example in phonetic representation (Lemetty, 1999).

The process in this step changes according to methods. The formant synthesis is based on the modeling of the resonances in the vocal tract. So for formant synthesizer, at least fundamental frequency, formant frequencies, duration and amplitude of each sound segment is necessary (Lemmety, 1999). For concatenative synthesizer, in this module, selected units are concatenated.

During concatenating two diphones if the waveforms of the two diphones edges across the juncture are very different, a perceptible click will result. Thus a windowing function to the edge of both diphones is required to be applied so that the samples at the juncture have low or zero amplitude. Furthermore, if both diphones are voiced, the two diphones are joined pitch-synchronously. This means that the pitch periods at the end of the first diphone must line up with the pitch periods at the beginning of the second diphone; otherwise the resulting single irregular pitch period at the juncture is perceptible as well (Jurafsky and Martin, 2008).

After concatenation, these units are modified in duration and “melody” to smoothly join each other and achieve the prosody of a natural utterance. In order to perform these modifications without introducing unnatural-sounding artifacts, signal modeling techniques, Synchronous Overlap-Add method (SOLA) such as PSOLA (Pitch Synchronous Overlap and Add), TD-PSOLA (Time Domain- Pitch Synchronous Overlap and Add) technique, must be employed (Moulines and Charpentier, 1990)

The input signal is divided into overlapping blocks of a fixed length and each block is shifted according to the time scaling factor α. Then the discrete-time log Δt_n of highest cross-correlation is searched in the area of the overlap interval. At this point of maximum similarity the overlapping blocks are weighted by a fade-in and fade-out function. SOLA block processing algorithm is represented as follows (Sanjaume, 2002).

PSOLA is a method used to manipulate the pitch of a speech signal to match it to that of the target speaker. The algorithm can be decomposed in two phases. The first one analyzes and segments the input signal, and the second phase overlaps and adds the extracted segments to synthesize the time stretched signal (Patton, 2007) (Sanjaume, 2002)

In the analysis phase, first step is to determine the pitch period. Second, the speech signal is divided into separate but overlapping smaller signals. This is accomplished by hanning windowing segments around each *pitch marks* or *peak amplitude* in the original signal (Upperman, 2004).

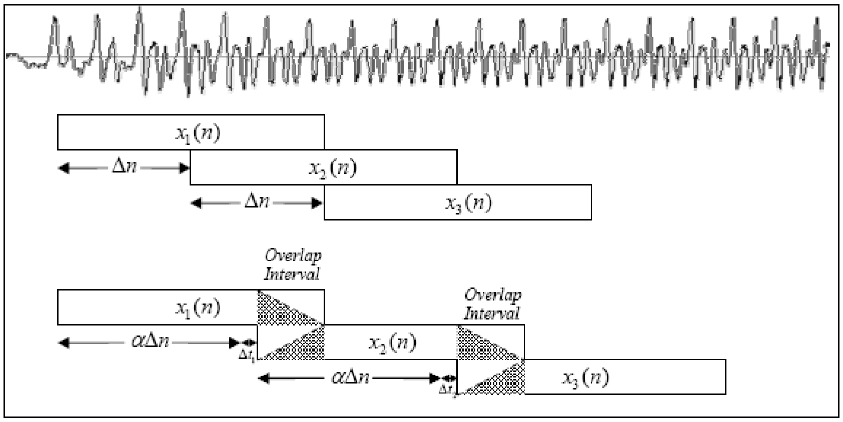


Figure 1.7 SOLA Time Scale (Sanjaume, 2002)

In the synthesis phase, the smaller signals are modified by either repeating or leaving out speech segments, depending on whether the pitch of the target speaker is higher or lower than the pitch of the source speaker. Lastly, the remaining smaller segments are recombined through overlapping and adding (Patton, 2007)

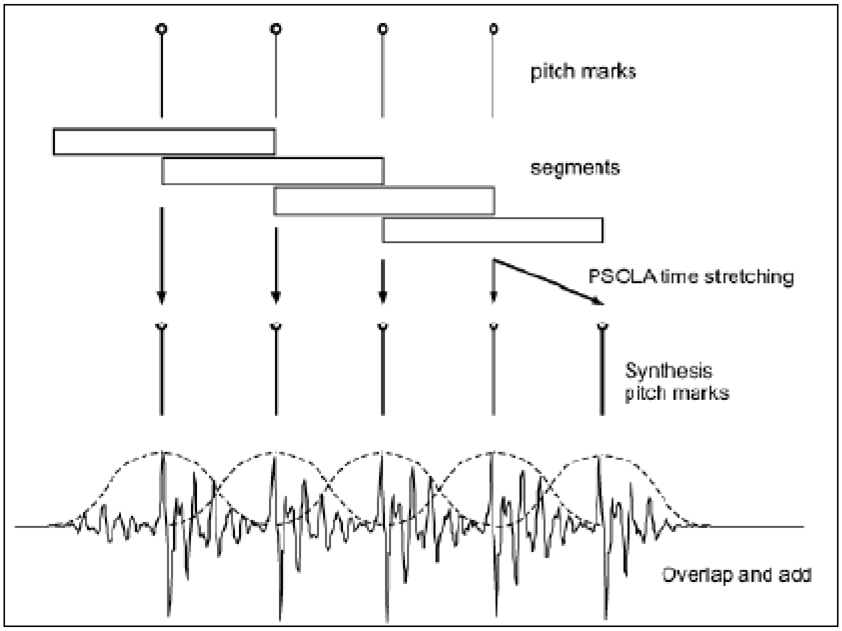


Figure 1.8 PSOLA Synthesis (Zolzer, 2002)

1.3.3 Technologies of Speech Synthesis

In the literature there are two basic categories of methods to produced synthesized speech: *Synthesis-by-rule* and *synthesis-by-concatenation* (Dutoit, 2001). Formant synthesis and articulatory synthesis are rule-based methods. The architecture of most modern commercial TTS systems is based on concatenative synthesis, in which samples of speech are chopped up, stored in a database, and combined and reconfigured to create new sentences (Jurafsky and Martin, 2008)

1.3.3.1 Concatenative Synthesis

Concatenative synthesis is based on the concatenation of segments of recorded speech. It is characterized by storing, selecting, and smoothly concatenating pre-recorded human utterances (phonemes, syllables, or longer units) (Cerrato, 2005).

In this approach, to prepare *speech database*, the small pieces are either cut from the recordings or recorded directly and then stored. Then, at the synthesis phase, units selected from the speech database are concatenated and, the resulting speech signal is synthesized as output. Figure 1.9 shows block diagram of the concatenative TTS system.

Concatenative synthesis has the potential for producing the most natural-sounding synthesized speech. However, differences between natural variations in speech and the automatic segmentation of the waveforms can cause audible glitches in the output[8]

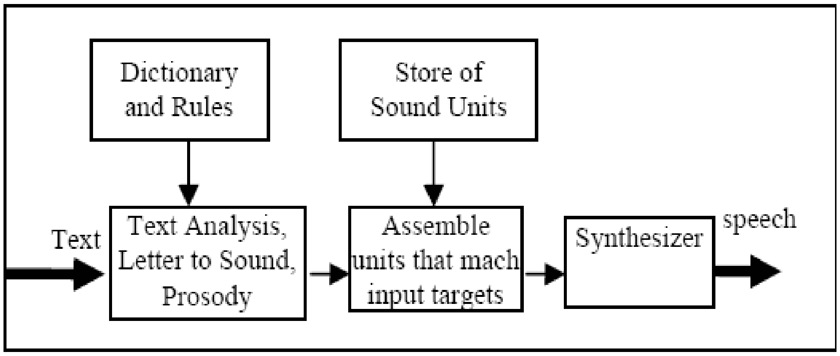

Dictionary and Rules
Store of Sound Units
Text
Text Analysis, Letter to Sound, Prosody
Assemble units that mach input targets
Synthesizer
speech

Figure 1.9 Block Diagram of the Concanative TTS System (Shah at el., 2004)

[8] Speech Synthesis: http://en.wikipedia.org/wiki/Speech_synthesis

There are three main sub-types of concatenative synthesis:

- Unit selection synthesis
- Diphone synthesis
- Domain-specific synthesis

Unit selection is one of the concatenative approaches. It uses large speech database and applies only a small amount of digital signal processing to the recorded speech providing the greatest naturalness. On the other hand, the size of the database required and selecting the appropriate unit from this large database can cause problems in this approach (Zhang, 2004).

Another concatenative approach is called the diphone synthesis that uses a minimal speech database containing all possible diphones (sound-to-sound transitions) in a language. The size of the diphone database may vary depending on the language. These units are combined by DSP techniques in the synthesis process resulting in a quality less than the unit synthesis but generally better than the formant synthesis and keeping the size of the database small[9]

Domain-specific synthesis concatenates long prerecorded sample of natural speech, like words, sentence and phrases. This method provides high quality and naturalness, but has a limited vocabulary. So it is used in limited, a particular domain. It is very suitable announcing (transit schedule announcements) and information systems (price list, the weather forecasting report), digit synthesis is concern only pronunciation of number sequence (Utama and Syrdal, 2006).

In this approach, the longer the phoneme means the more success of the system. For the units with different lengths, intelligibility and flexibility of system are also different. Flexibility is worse but naturalness of speech is better, if big units are used. Besides big units are for limited domain applications. Conversely, small units are for more flexible systems and wider applications.

[9] Speech Synthesis: http://en.wikipedia.org/wiki/Speech_synthesis

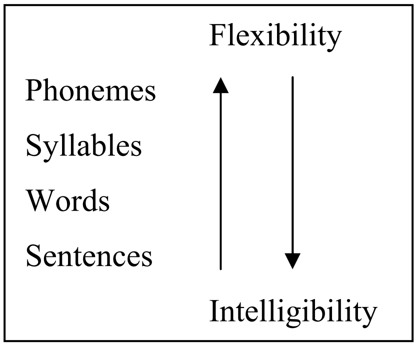


Figure 1.10 Flexibility and Intelligibility Variation According to Unit Length

1.3.3.2 Formant Synthesis

In formant synthesis human speech samples are not used at runtime and the vocal tract transfer function is modeled by simulating formant frequencies and formant amplitudes[10]. Formant synthesis provides intelligibility but naturalness is not assured. The memory and microprocessor power requirement is less than the other techniques; therefore it is suitable for the limited devices (Styger and Keller, 1994).

A formant synthesizer recreates the speech spectrum using a collection of rules and heuristics to control a digital filter model of the vocal tract. Klattalk and DECTalk are examples of formant-based synthesizers (Water and Levergood, 1993).

1.3.3.3 Articulatory Synthesis

Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. Manner of articulation describes how the tongue, lips, jaw, and other speech organs are involved in making a sound make contact[11].The shape of the vocal tract can be controlled in a number of ways which usually involves modifying the position of the speech

[10] Speech Synthesis: http://en.wikipedia.org/wiki/Speech_synthesis

[11] Speech Production: http://en.wikipedia.org/wiki/Speech#Speech_production

articulators, such as the tongue, jaw, and lips. Speech is created by digitally simulating the flow of air through the representation of the vocal tract.

The first software articulatory synthesizer regularly used for laboratory experiments was developed at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer, and Paul Mermelstein. This synthesizer is known as ASY[12].

1.3.4 Application Area of Text-to-Speech

These systems have been widely used as assistive technological tools for a long time. Some of application areas of TTS are below:

- Information Service by Voice
    - price list
    - the weather forecasting report
- Listening daily journal, book, textbook, e-mail, sms while working, doing housework, traveling, driving, exercising. For example spoken web is a web portal, managing a wide range of online data-intensive content like news updates, weather and travel, business articles for computer users who are blind or visually impaired[13].
- Talking pc, doll, dictionary, mobile.
    - mobile rings or speaks to warn, for example, it says “time is 11:15, call the teacher”
- Screen reader[14] is a software that attemts to identify and interpret what is being displayed on the screen, reading all digital screens for blind or visually impaired people, saying the names of tab, buttons and links, describing images on the web (Eagleton, 2006) or saying buttons’ names on ATM screen and computer screen.

[12] Articulatory Synthesis: http://www.haskins.yale.edu/facilities/asy.html

[13] Spoken web: http://www.spoken-web.com/

[14] Screen reader: http://en.wikipedia.org/wiki/screen_reader

- Improving pronunciation: While learning language especially foreign language correct pronunciation is important. There is no teacher every time to teach. TTS applications help to learn pronunciation (González, 2007).
- Spoken dialog system, is a dialog system delivered through voice. It has two essential components that do not exist in a text dialog system: A speech recognizer and a text-to-speech module[15]. PEGASUS is a conversational interface that provides information about flight status; Pegasus enables users to obtain flight status information over a telephone line (Zue at el., 1994).
- Finally, speech synthesis can be used to speak for sufferers of neurological disorders, such as astrophysicist Stephen Hawking who, having lost the use of his voice due to ALS, speaks by typing to a speech synthesizer and having the synthesizer speak out the words (Jurafsky and Martin, 2008).

1.3.5 Major Text-to-Speech Applications

There are several commercial and non-commercial applications. Only a few of most known are explained. In Table 1.1, applications and their based approaches are given.

Table 1.1 Major TTS Applications

Approaches	Products
Articulatory	ASY
	VOCODER
Formant	MITalk
	DECTalk
	INFOVOX (early version)
Formant & Concatenative	CHATR
Concatenative	INFOVOX (last version)
	Next-Gen
	CNET-PSOLA
	FESTIVAL
	MBROLA
	WHISTLER

[15] Spoken Dialog System: http://en.wikipedia.org/wiki/Spoken_dialog_system

1.3.5.1 ASY

ASY is most known articulatory synthesizer. The software implementation ASY provides a kinematic description of speech articulation in terms of the moment-by-moment positions of six major structures; the jaw, velum, tongue body, tongue tip, lips and hyoid bone, all presented graphically for viewing in the midsagittal plane[16]. ASY was a computational model of speech production based on vocal tract models developed at Bell Laboratories in the 1960s and 1970s by Paul Mermelstein, Cecil Coker, and their colleagues[17]

1.3.5.2 MITalk and DECTalk

Most well-known formant synthesizer is the Klatt formant synthesizer and its successor systems, including the MITalk systems (Allen at al., 1987) and Klattalk software used in Digital Equipment Corporationn's DECTalk (Klatt, 1982), (Jurafsky and Martin, 2008).

MITalk has used letter-to-sound rules instead of pronunciation dictionary, since computer memory was too expensive to store large dictionary.

DECTalk system is originally descended from MITalk and Klattalk. System is available for British and American English, German, French and Spanish. DECtalk Software has multithreaded processing and the new text-to-speech application programming interface for UNIX and NT workstations.

1.3.5.3 INFOVOX

The Infovox speech synthesis system is the result of many years of work at the Royal Institute of Technology in Stockholm, Sweden. The system is originally descended formant synthesizer. But latest infovox version unlike earlier systems is based on diphone concatenation of pre-recorded samples of speech (Lemmetty, 1999).

[16] ASY: http://www.haskins.yale.edu/facilities/asy.html

[17] Articulatory Synthesis: http://en.wikipedia.org/wiki/Articulatory_synthesis

Infovox commercial software runs on Windows. It can generate speech in Swedish, German, French, Russian, Spanish and Turkish. The best language is Swedish by far. For Turkish it is unsuccessful.

1.3.5.4 Bell Labs TTS

AT&T Bell Labs (Lucent Technologies) are combined with researches in speech recognition and speech synthesis. In 1937, the VOCODER, the first electronic speech synthesizer was invented and demonstrated by Homer Dudley. This early analog system was the forerunner of Bell Labs work in articulatory synthesis[18]. Present systems are based on concatenation of diphones, triphones. Several commercial products are available such as TrueTalk [19], Flextalk by AT&T.

AT&T's Next-Gen TTS is implemented within the Festival framework. It is based on best-choice components of the AT&T Flextalk TTS, the Festival System from the University of Edinburgh, and ATR's CHATR system. From Flextalk, it employs text normalization, letter-to-sound, and prosody generation. Festival provides a flexible and modular architecture for easy experimentation and competitive evaluation of different algorithms or modules. In addition, CHATR's unit selection algorithms were modified and adopted (Beutnagel at el., 1999).

1.3.5.5 CNET PSOLA

Diphone based synthesizer CNET (Central National d'Etudes Télécommucicaions) was introduced in mid 1980's by France Telecom. Systems used PSOLA algorithm (Lemmetty 1999). Also Psola is a trademark of CNET. A commercial product the ProVerbe Speech Engine from ELAN Informatique produces natural sounding speech from written text. Naturalness is achieved by using the TD-PSOLA process from the CNET which is based on the concatenation of elementary speech units (including diphones). Supported languages are British English, American English, Russian, German, French and Spanish.

[18] Bell Labs TTS: http://www.bell-labs.com/project/tts/tts-overview.html

[19] TrueTalk: http://web.archive.org/web/19980214091103/www.entropic.com/truetalk.html

1.3.5.6 CHATR

CHATR was developed at ATR. CHATR offers a number of synthesis methods: Klatt formant synthesis, LPC based diphone synthesis and a number of concatenative synthesis methods (each with its own internal options to choose between different unit selection strategies) (Black and Taylor, 1994).

1.3.5.7 FESTIVAL

Festival is a general multi-lingual speech synthesis system originally developed at Centre for Speech Technology Research (CSTR) at the University of Edinburgh. Substantial contributions have also been provided by Carnegie Mellon University and other sites. Festival is free software. Festival and the speech tools are distributed under an X11-type license allowing unrestricted commercial and non-commercial use alike.

Festival offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-like command interpreter for general customization and extension.

Festival is designed to support multiple languages, and comes with support for English (British and American pronunciation), Welsh, and Spanish. Voice packages exist for several other languages, such as Castilian Spanish, Czech, Finnish, Hindi, Italian, Marathi, Russian and Telugu (Black at el., 1999).

Festival support for MBROLA already supports a number of diphone sets including French, Spanish, German and Romanian, Hindi, Swedish, Turkish.

1.3.5.8 MBROLA

The aim of the MBROLA project, initiated by the TCTS Lab of the Faculté Polytechnique de Mons (Belgium), is to obtain a set of speech synthesizers for as many languages as possible, and provide them free for non-commercial applications. The ultimate goal is to boost academic research on speech synthesis, and particularly on prosody generation, known as one of the biggest challenges taken up by TTS synthesizers for the years to come (Dutoit at el., 1996).

Central to the MBROLA project is MBROLA[20], a speech synthesizer based on the concatenation of diphones. It takes a list of phonemes as input, together with prosodic information (duration of phonemes and a piecewise linear description of pitch), and produces speech samples on 16 bits (linear), at the sampling frequency of the diphone database used (it is therefore not aTTS synthesizer, since it does not accept raw text as input). This synthesizer is provided for free, for non commercial, non military applications only. Non-commercial TTS systems and components are compatible with MBROLA such as EULAR, MBRDICO, Festival

1.3.5.9 WHISTLER

Whistler (Whisper Highly Intelligent Stochastic TaLkER) is the TTS system being developed at Microsoft. Goal is to Whistler trainable that automatically learns the model parameters from a corpus, scalable and natural. Whistler can produce synthetic speech that sounds very natural and resembles the acoustic and prosodic characteristics of the original speaker. The speech engine is used to construct syllable/word/phrase dependent triphone sequences. Training procedure is on Hidden Markov Models. To segment the speech corpus was used the speech features developed in Whisper speech recognition systems, to align the input waveform with phonetic symbols that are associated with Hidden Markov Models states (Huang at el., 1996-1997).

1.3.5.10 Text-to-Speech for Turkish

Most of the researches on speech synthesis are being improved for English. There are some researches for other language like German, French, Chinese, Spanish and some major English TTS applications support other language and these systems are synthesized speech successfully for a lot of language.

For Turkish there is not enough research to produce high level natural syntactic speech. There are some academic researches. One of the most academic research products is OKU[21] that was developed at Bilkent University. Latest version is OKU 4.0. The project

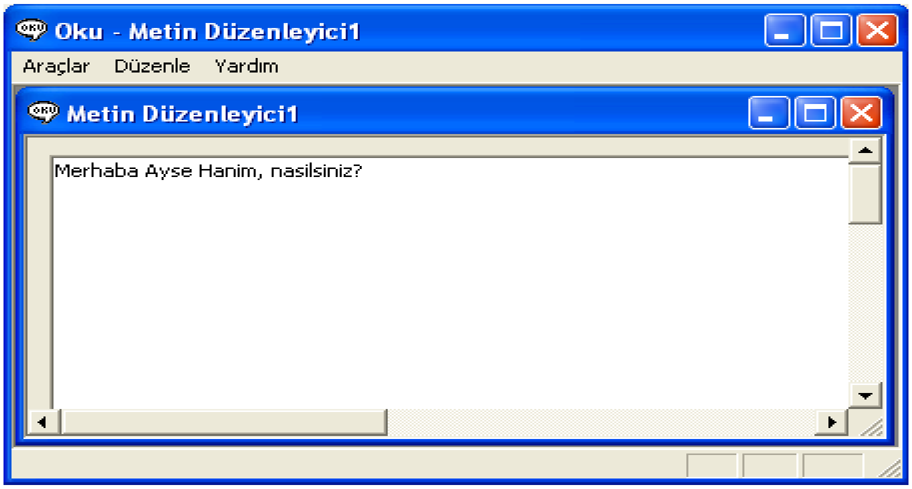
[20] MBROLA homepage: http://tcts.fpms.ac.be/synthesis/mbrola.html

[21] OKU Turkish TTS: www.cs.bilkent.edu.tr/oku

involves the design and implementation of a tool for visually handicapped people speaking Turkish. Using this tool, visually handicapped people can be able to read, edit and print a document, browse in the internet, send and receive email messages. OKU 1, 2 and 4 versions are syllable based but OKU 3.0 is compliant with MBROLA

OKU is syllable based synthesizer. If a word cannot be syllabified, it is spelled out (letters are pronounced) such as TCMB, YTL. Speech database of OKU contains recorded pronunciation of 3537 words and 1819 syllables were extracted from these words. These 1819 syllables do dot cover all Turkish. If the sound for a syllable is not recorded, it is synthesized from the smallest units. For exampe *"cenk"* is not recorded. It is synthesized from *"cen"* and *"k*". Some foreign words need to be pronounced differently such as *Microsoft, George, show*. These are pronounced as m*ikrosoft, corc, şow*. Some abbreviations should be spelled out such as DİE → de-i-e, ABD → a-be-de. These are stored in text files, can be extended by the users (Güvenir, 2005).

A screenshot of OKU 4.0 is shown in Figure 1.11. Oku is vocalized all charecters (also space and comma) during pressing key and then vocalized word when pres space. Whole text can be vocalized at the end. The program is not recognized some special Turkish letter such as *ş,ı*. It is vocalized them *soru işareti (question mark).*



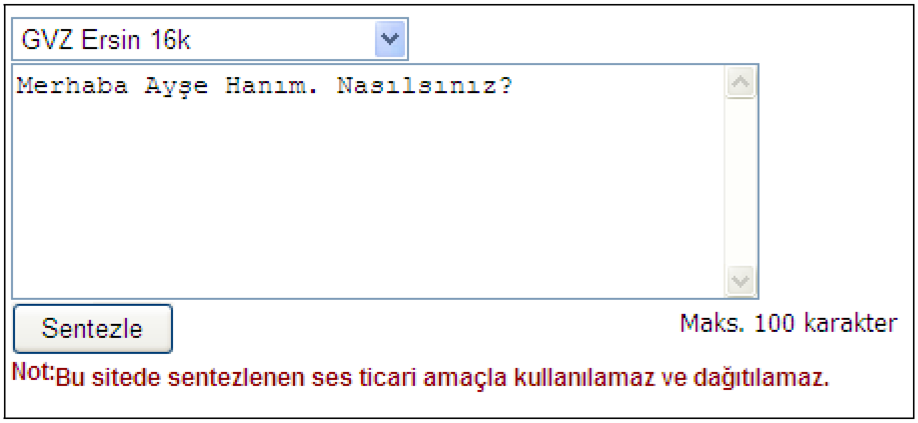
Merhaba Ayse Hanim, nasilsiniz?

Figure 1.11 A screenshot of OKU 4.0

Some commercial TTS systems for Turkish are also available such as TeknoSes[22], GVZ[23]. Both of them use Microsoft SAPI (Speech Applications Program Interface). With

[22] TeknoSes Speech Synthesizer: www.teknoses.com

Teknoses TTS blind users can use computer with all functions, they send and receive mails write documents, and surf up the internet. GVZ speech synthesizer has both female and male speech. The module of GVZ web vocalization is developed to vocalize the textual content of websites. Sabah Newspaper provides free news service at its website, www.sabah.com.tr, by benefiting from this module. An online demo is available its website www.gvz.com.tr. In Figure 1.12, a screenshot of demo is shown.



GVZ Ersin 16k

Merhaba Ayşe Hanım. Nasılsınız?

Maks. 100 karakter

Not:Bu sitede sentezlenen ses ticari amaçla kullanılamaz ve dağıtılamaz.

Figure 1.12 A screenshot of GVZ

1.3.6 Problems of Text-to-Speech

There are several problems at all level in speech synthesis. Some of text processing problems are word-sentence boundaries, identification of numbers, date, time, phone, abbreviations, converting text-to-phonetic, analyzing special characters. Some of signal processing problems are prosody, preparing speech database, synthesized quality (naturalness and intelligibility) speech, evaluation

1.3.6.1 Text Processing Problems

Tokenizing sentence, word boundaries, and sentence boundaries are the problems. Spaces are not enough for tokenizing. It is normally delimited by whitespace should be considered as full words or further broken down, and it have seen that apart from a few

[23] GVZ Speech Synthesizer: www.gvz.com.tr

exceptions, whitespace in conventional writing is actually a quite reliable guide as to word boundaries. It is splited when a contiguous non-whitespace sequence changes from one character grouping to another (e.g. 10cm → 10 cm, 13:15'de → 13:15 de, 3-5 → 3 - 5).

Sentence is divided into token by word boundaries, token sequences are obtained and each token must be classified. Tokens are non-standard words or natural language words. Non-standard words need to be expanded into language (English, Turkish) words before they can be pronounced such as digits, dates, times, abbreviations (Jurafsky and Martin, 2008).

Semiotic classification and verbalization problems are other problems in text processing level. Semiotic classification is therefore a question of assigning one of the correct the known semiotic classes to each of tokens in sentence (Taylor, 2007). Some of semiotic classes are shown in Table 1.2.

Table 1.2 Example of Semiotic Classes

Class	**Example**
Abbreviation	TDK, UNICEF
Date	12.11.2000, March 4
Time	13:10, 11:00 pm
Phone	(0212) 866 3300
number (ordinal)	VI Murat, 1st, 5.
number (cardinal)	123, 0.6, 14/9
card number	1234 5678 5678 9876
word-text	child, oyuncak

Semiotic classification is difficult because there are many types of ambiguities for non-standard words. One such kind of an ambiguity is related to the numbers. For example, if the number in textual form is 8540178, then it is classified and translated into natural language by different ways depending on the context. If it is a phone number, it is pronounced as 854 01 78 (eight hundred fifty four zero one seventy eight), on the other hand, if it is a currency, than is it is converted as 8 540 178 (eight million five hundred forty thousand one hundred seventy eight).

For example, there is a sample sentence: *Toplantı 13/11/2000 tarihinde saat 13:15'de yapılacaktır (The meeting will be on 1:15 PM on 13/11/2000).*

This sentence is divided into token by word boundaries, token sequences are obtained and each token must be classified. Semiotic classes of tokens of the sentence and their translations into word are shown in Table 1.3. Finally, each token must be translated into natural language words.

Table 1.3. Semiotic Class of Tokens and Translation

Token	**Semiotic Class**	**Translation into word**
Toplantı	word-text	toplantı
13/11/2000	date	onüç onbir iki bin(thirteen eleven two hundred)
tarihinde	word-text	tarihinde
13:15	time	onüç onbeş thirteen fifteen
te	word-text	word-text
yapılacaktır	word-text	word-text

To solve semiotic classification, some rules may be used. If the numbers of neighbors contain *sicil numarası (credentials number), vergi numarası (tax number), kart numarası (credit card number), TC kimlik no (identity number),* it can be divided two or four length digits groups and then vocalized.

Identification of date and time may be solved as regular expressions. Dates and times may include patterns such as 10/12/99, 12.09.200 (tree digit groups divided by slashes, dot or hyphens), 13:15 and 12.00 (two digit groups divided by dot or colon.)

If the numbers start with 0 and it is not in a date format (02/02/2000), it may be a phone number such as 02128663300. If the numbers have thousands of separator (dot for Turkish) or decimal point s (comma for Turkish), this may be a currency such as 12.344 and 12,34.

The mean of verbalization is translation from non-standard words into natural words. Also there are many types of ambiguities occurring in the TTS systems in verbalization level.

Ambiguity of time verbalization is another problem. For example 13:15 is translated into a word following different ways: *onüç onbeş (thirteen fifteen) or onüç çeyrek (quarter past thirteen) or bir çeyrek (quarter past one PM).*

Special characters and symbols, such as '@, &, -, /, (,)', may cause another special ambiguity problem. Hyphen is vocalized or not vocalized in between numbers. For example, *3-5* is vocalized as *üç beş (three five)* or *3-5=2* is vocalized as *üç eksi beş (three* *minus five).* Hyphen is between words tokens not vocalized such as *sarı-kırmızı* (yellow red).

There is also ambiguity for slash character vocalization in different context. For example 28/3 is vocalized as *yirmi sekiz taksim üç* in address. If there are some terms about *oran (rate), hisse (allotment), pay (apportionmen)* in context 1/3 will be vocalized as *bir bölü üç (one over three).*

In a numbering text with digits and hyphen, dot or close parenthesis (e.g 1), 1-, 1.), these special characters which are hyphen, dot or close parenthesis, are not vocalized. But digits of ordinal numbers and dot combinations must be vocalized such as *2.bölüm- ikinci bölüm (second chapter)*.

Open and close parentheses and hyphen in a sentence for explanation are not vocalized. For example: *Yarın (çarşamba) geleceğim. –I am going to come tomorrow (Wednesday*). In addition, open and close parentheses in phone numbers are not vocalized, for example *(0212) 866 3300* is vocalized as *zero two one two eight six six three three zero zero.*

Abbreviation verbalization is another problem in text-phoneme level in text analysis. It is vocalized as written (*UNICEF, ATO*) or letter by letter (*DVD, TCK*) or different word (cm → centimeter). In some languages, there is also contextual problem. For example, in 3 cm or 1 cm expressions, cm is vocalized as centimeter(s). In Turkish language there is no plural form of abbreviations such as kg (kilogram), g (gram), l (liter). So, there is no problem about this

1.3.6.2 Speech Processing Problems

There are many problems in speech process module from database preparation to smoothing synthetic speech at all level.

Preparing speech database is the major problem. According to units length of database size and quality of speech are changed. If small units are used, database size is small but cutting small units from continues speech at correct boundaries is very difficult.

To cut manually requires time and brute force. Automatically alignment is more difficult than manual for small units such as syllable, phoneme. If long units such as word or phrase, are used alignment and cutting manual and automatically is easier than small units because boundaries of word and phrase are silence in continuous speech, but in language there are too many words so database size very huge. And some language such as agglutinative language, it is very productive and it is likely to derive plenty of new words by adding affixes and suffixes to words. So it is inconvenient to store the phonemes as word based.

Naturalness means the system should sound with intonation just like a human. It is not desirable to hear robotic sound instead natural human sound. But TTS systems still can't produce perfect sound like natural human speech. One of difficult tasks for naturalness is intonation that may change in context. In below conversation answer sentences are the same but intonation is on different word. In first conversation intonation is on *okula,* in second is *Ayşe* are shown underline.

-*Ayşe nereye gitti? (Where did Ayşe go?)*
-*Ayşe okula gitti. (Ayşe went to school)*

-*Kim okula gitti? (Who went to school?)*
-*Ayşe okula gitti. (Ayşe went to school)*

Finding correct intonation, stress and duration form written text is another problem to produce naturalness speech. The prosody of continues speech that is considered melody, rhythm, depends on many separate aspects, such as meaning, speaker characteristic and emotions (Lemetty, 1999). The prosodic dependencies are shown in Figure 1.13. Unfortunately, written text doesn't contain all these features. Sometimes maybe punctuations help for prosody analysis such as exclamation mark or question mark.

In Turkish there is liasion rule at reading text level. The liaison is the grammatical circumstance in which a usually silent consonant at the end of a word is pronounced together with the vowel at the beginning of the word that follows it. The detail of liaison is described in Section 3.1.

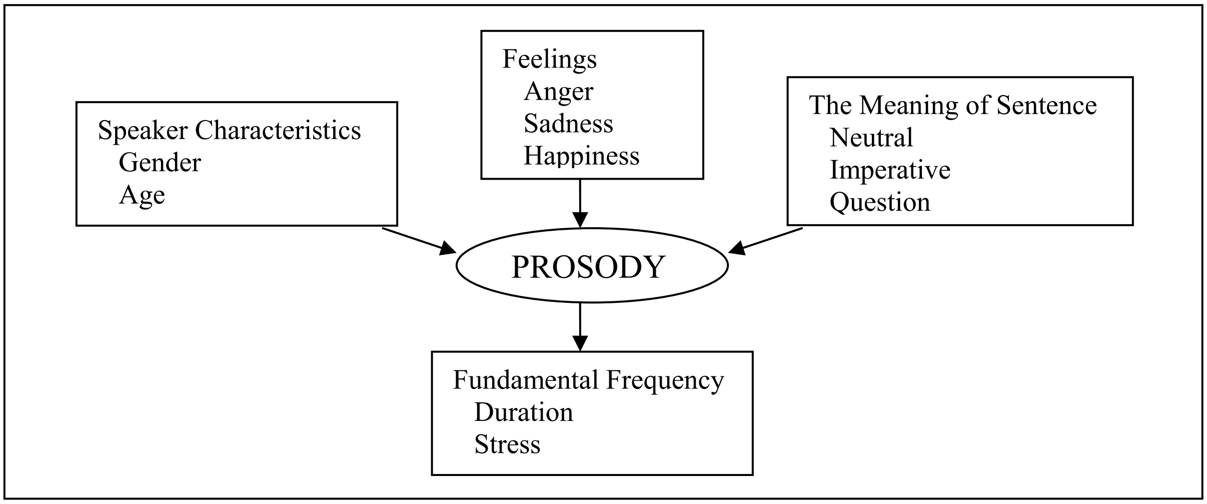


Figure 1.13 Prosodic Dependencies (Lemetty, 1999)

Intelligibility is the ability of a listener to understand the message from the speech. Pronunciation of the letters that have different pronunciation in different word is problem for intelligibility. For example in English *o* letter pronounced in *box* and *book* differently. In Turkish, the letter of "*a"* has tree different pronunciation. *Kar (snow, ka is pronounced deeply in this word), kâr (profit, ka is pronounced softly), aza (member, first "a" is pronounced long vowel).*

CHAPTER 2

TURKISH TEXT-TO-SPEECH SYSTEM ARCHITECTURE

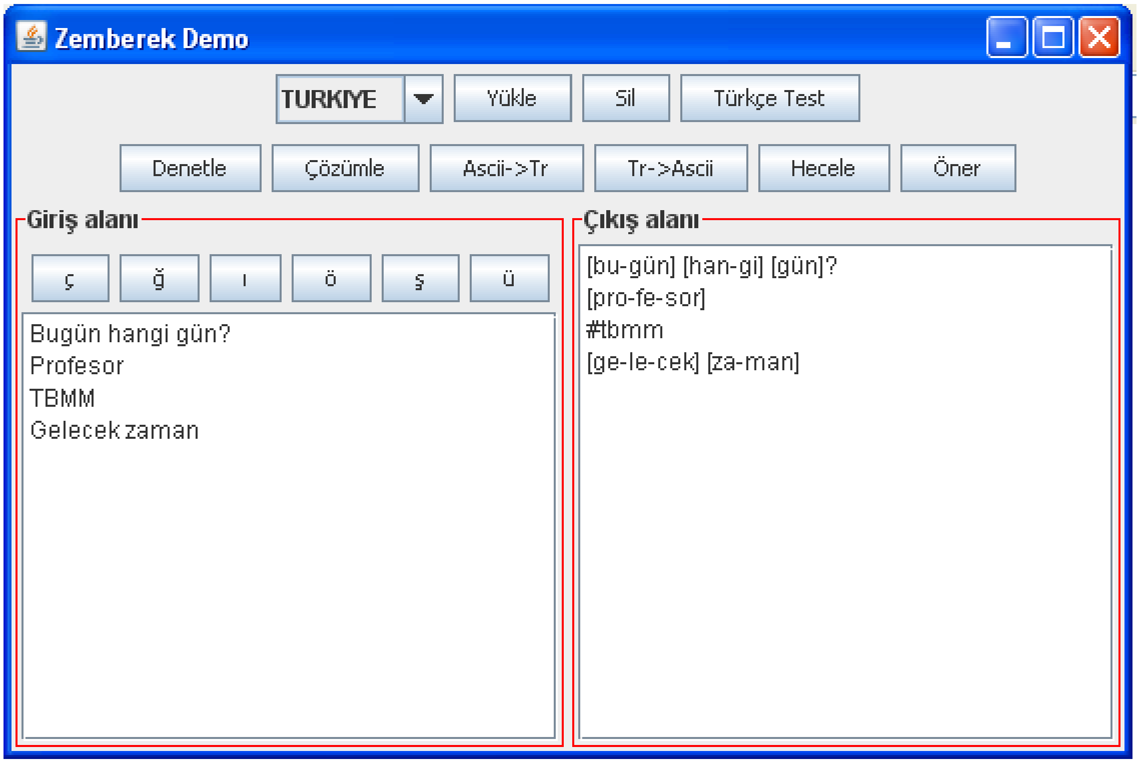
In concatenative approach, the longer the phonemes the more success of the system achieved. It is believed that the concanative system will yield better results, for Turkish which is an agglutinative language. But, it is inconvenient to store the phonemes as word based. It is very productive and it is likely to derive plenty of new words by adding affixes and suffixes to words, for it is an agglutinative language. Furthermore, the appearance of new words and the disappearance of old words in the language will require the updating of the database. Instead, huge databases and updating will be precluded by working on smaller phonemes. So concatenative unit selection synthesis approaches is prefered and for units syllables and phonemes are prefered.

There are two major modules, NLP and DSP, in our TTS system, too. The modules and processes that take place in Turkish TTS systems are shown in Figure 2.2.

For some processes of NLP module, analyzing the text and syllabification, Zemberek project[24] is used. Some processes, preprocessing of the text and analyzing the syllable and non-standart words, were implemented.

Zemberek is an open source, platform independent, general purpose NLP library and toolset designed for Turkic languages, especially Turkish. The framework provides basic NLP operations such as spell checking, morphological parsing, stemming, word construction, word suggestion, converting words written only using ASCII characters (so called *deasciifier*) and extracting syllables (Akın and Akın, 2009). A screenshot of Zemberek which is taken from the example of extracting syllables is shown in Figure 2.1.

[24] Zemberek project: http://code.google.com/p/zemberek/



```
Bugün hangi gün?
Profesor
TBMM
Gelecek zaman
```

```
[bu-gün] [han-gi] [gün]?
[pro-fe-sor]
#tbmm
[ge-le-cek] [za-man]
```

Figure 2.1 Zemberek Screenshot

2.1 TURKISH LANGUAGE PROPERTIES

2.1.1 Turkish Phoneme Set

Phoneme is the term used by linguists to describe distinctive smallest sounds in a language. In most languages the written text does not correspond to its pronunciation so that in order to describe correct pronunciation some kind of symbolic presentation is necessary. Every language has a different phonetic alphabet and a different set of possible phonemes and their combinations.

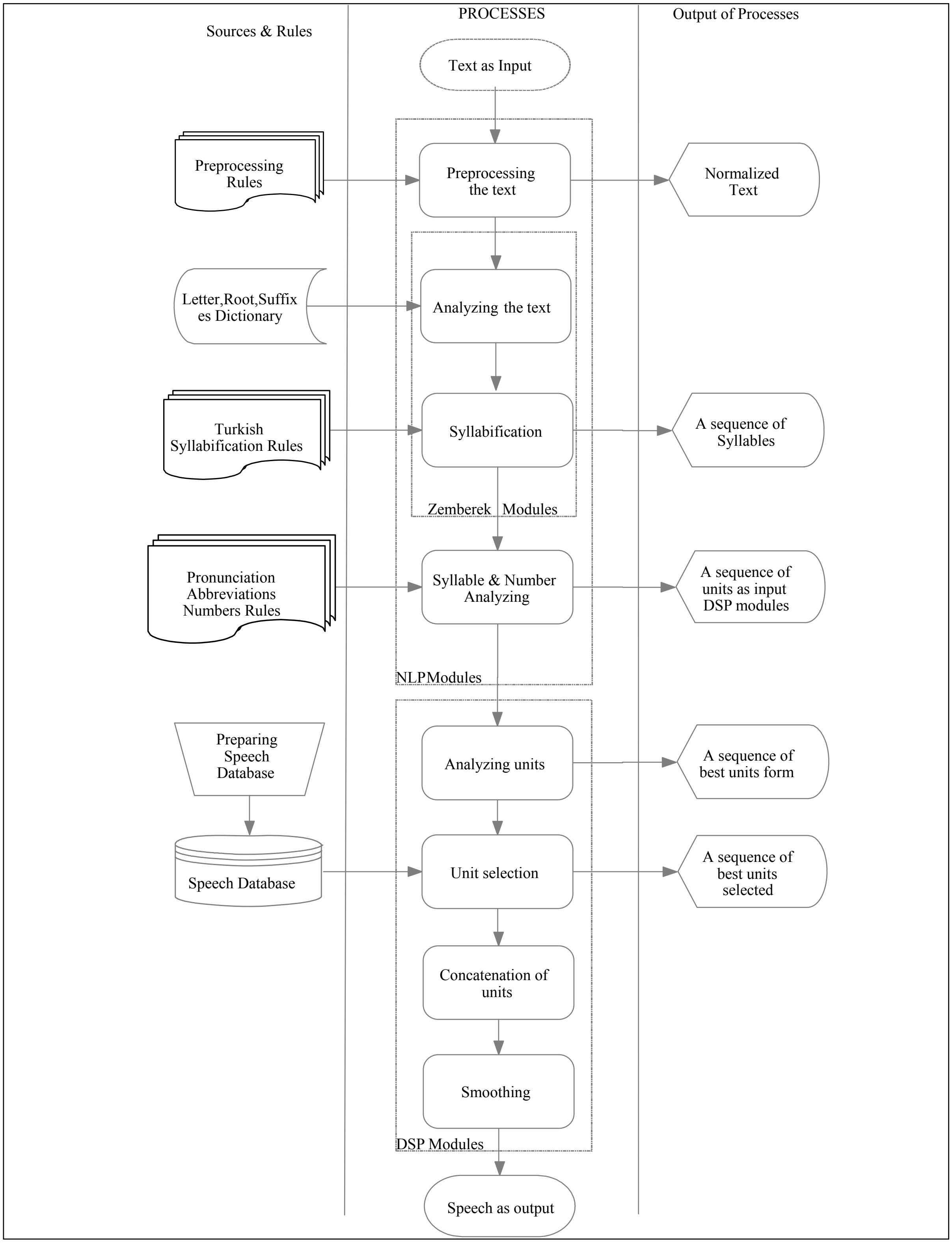


Figure 2.2 Turkish TTS System Architecture

Table 2.1. Letters and Phonemes in Turkish

Letter	Phoneme	Rule	Example in Turkish	English Translation
	a	normal	hala	aunt
	â	circumflex	hâla	still
a	**a:**	long vowel	ada:let	justice
b	**b**	normal	balık	fish
c	**c**	normal	cam	glass
ç	**ç**	normal	çok	a lof of
d	**d**	normal	demir	iron
	e	normal	sekiz	eight
e	**e:**	long vowel	me:zun	grad
f	**f**	normal	fakülte	faculty
g	**g**	normal	gazete	newspapaer
ğ	**ğ**	normal	boğaz	bosporus
h	**h**	normal	hece	syllable
ı	**ı**	normal	ılık	lukewarm
	i	normal	inci	pearl
i	**i:**	long vowel	şi:ve	vernacular
j	**j**	normal	jilet	blade
k	**k**	normal	kedi	cat
l	**l**	normal	lamba	lamb
m	**m**	normal	mucit	inventor
n	**n**	normal	neşe	blitheness
	o	normal	oda	room
o	**ô**	circumflex	lôkum	turkish delight
ö	**ö**	normal	örnek	example
p	**p**	normal	pembe	pink
r	**r**	normal	resim	picture
s	**s**	normal	sis	fog
ş	**ş**	normal	şehir	city
t	**t**	normal	taze	fresh
	u	normal	kuş	bird
	û	circumflex	mahkûm	convict
u	**u:**	long vowel	numu:ne	model
ü	**ü**	normal	üzüm	grape
v	**v**	normal	veri	data
y	**y**	normal	yaz	summer
z	**z**	normal	zil	bell

There are 21 consonants[25] (C) and 8 vowels[26] (V), yielding a total of 29 characters in Turkish alphabet. In Turkish each letter has one pronunciation, but some letters pronounced different in some words that have been barrowed foreign languages such as Arabic and Persian. For example, the letter of *a* has different pronunciation in *ad* (name), *aza (member)* and *adalet* (justice) and also in adalet first *a* and second *a* are pronounced differently[27]. Letter and phonemes of Turkish language and example word that includes letters and phonemes are shown in Table 2.1.

2.1.2 Structure of Turkish Syllables

The syllables in Turkish are formed with the combination of consonants and vowels in many ways. Syllables are generally formed from 1-6 characters and contain a vowel and consonants with some minor exceptions. However, some of these syllables, especially the ones that have 5 or 6 characters are very rare. There are only seven types of syllables of Turkish as shown in Table 2.2.

Table 2.2 The Structure of Turkish Syllables (Görmez and Orhan, 2008)

Syllable Structure	**Sample Syllables**	**The Possible Number of Different Syllables**	
V	a, e, ı, i, o, ö, u, ü	29	29
VC	ab, ac, aç, ad, … ,az, eb, ec, …	8*21	168
CV	ba, be, bı, bi, … , za, ze, zı, zi, …	8*21	168
CVC	bel, gel, köy, tır, …	8*21*20	3.528
VCC	alt, üst, ırk, …	8*21*21	3.528
CCV	bre	21*21*8	3.528
CVCC	kurt, yurt, renk, Türk	21*8*21*21	74,088
		Total	**85037**

The percentages of these syllables are given in Figure 2.3 that is obtained from the Turkish corpora prepared in a research of this domain (Aşlıyan and Günel, 2008).

[25] Consonants in Turkish: b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z

[26] Vowels in Turkish: a, e, ı, i, u, ü, o, ö

[27] TDK, "Uzun Ünlü, Düzeltme İşareti", In Türkçe İmla Kılavuzu, Türk Dil Kurumu, 2000

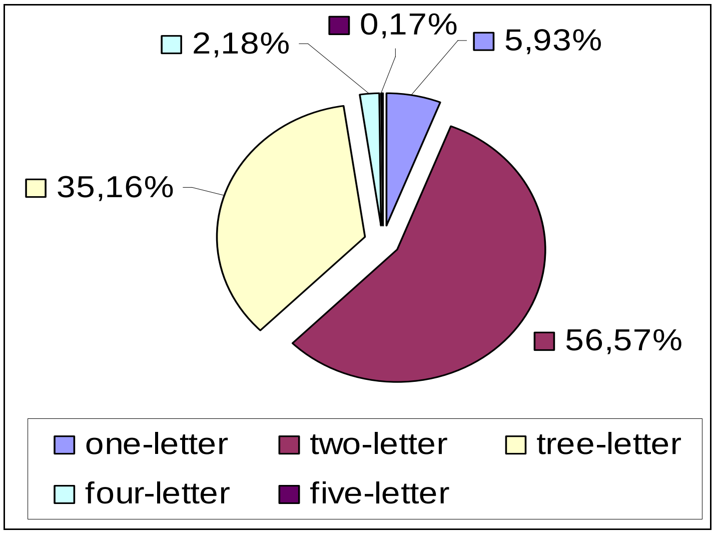


Figure 2.3 The Percentages of Various Syllable Lengths in Turkish

2.2 SPEECH DATABASE

The ratios of the syllables support the claim asserted above. The speech database of Turkish includes single or double letter sounds as the smallest phoneme in our system. The rest of the syllables are formed from the concatenation of these sounds. The longer syllables are synthesized as follows:

- CV+C ➔ CVC
- VC+C ➔ VCC
- CC+V ➔ CCV
- CV+C+C ➔ CVCC

Single letter syllables can have only the vowels and double letter syllables may have one consonant and one vowel and their order may change. The syllables that have 3 or more characters can be obtained by adding a consonant to double letter syllable. The resulting database has 365 recordings as a total and Table 2.3 shows how this is obtained. The possible CV-VC combinations for Turkish are shown in the Table 2.4 . Sound files are kept with the same name given in the table.

The rest of the syllables are formed from the concatenation of these sounds. The longer syllables are synthesized as follows and synthesized form examples are shown Table 2.5

- Target CVC: CV+C → CV+VC (git/go→gi+it)
- Target VCC: VC+C → VCC (üst/top→üs+t)
- Target CCV: CC+V → C(V)CV (bre[28]→ b(i)re)
- Target CVCC: CV+C+C→CV+VC+C(türk→tü+ür+k)

For CVC syllables last consonant is affected by previous vowel so VC is used instead of only C leading to a more natural sound.

Table 2.3 Diphones Kept in Turkish Speech Database

Syllable Combinations	Possible values
CV	21*8
VC	8*21
V	8
C	21
Total	**365**

Table 2.4 CV-VC Syllable Combinations (Görmez and Orhan, 2008)

C\V	a		e		ı		i		u		ü		o		ö	
b	ab	ba	eb	be	ıb	bı	ib	bi	ub	bu	üb	bü	ob	bo	öb	bö
c	ac	ca	ec	ce	ıc	cı	ic	ci	uc	cu	üc	cü	oc	co	öc	cö
ç	aç	ça	eç	çe	ıç	çı	iç	çi	uç	çu	üç	çü	oç	ço	öç	çö
d	ad	da	ed	de	ıd	dı	id	di	ud	du	üd	dü	od	do	öd	dö
f	af	fa	ef	fe	ıf	fı	if	fi	uf	fu	üf	fü	of	fo	öf	fö
g	ag	ga	eg	ge	ıg	gı	ig	gi	ug	gu	üg	gü	og	go	ög	gö
ğ	ağ	ğa	eğ	ğe	ığ	ğı	iğ	ği	uğ	ğu	üğ	ğü	oğ	ğo	öğ	ğö
h	ah	ha	eh	he	ıh	hı	ih	hi	uh	hu	üh	hü	oh	ho	öh	hö
j	aj	ja	ej	je	ıj	jı	ij	ji	uj	ju	üj	jü	oj	jo	öj	jö
k	ak	ka	ek	ke	ık	kı	ik	ki	uk	ku	ük	kü	ok	ko	ök	kö
l	al	la	el	le	ıl	lı	il	li	ul	lu	ül	lü	ol	lo	öl	lö
m	am	ma	em	me	ım	mı	im	mi	um	mu	üm	mü	om	mo	öm	mö
n	an	na	en	ne	ın	nı	in	ni	un	nu	ün	nü	on	no	ön	nö
p	ap	pa	ep	pe	ıp	pı	ip	pi	up	pu	üp	pü	op	po	öp	pö
r	ar	ra	er	re	ır	rı	ir	ri	ur	ru	ür	rü	or	ro	ör	rö
s	as	sa	es	se	ıs	sı	is	si	us	su	üs	sü	os	so	ös	sö
ş	aş	şa	eş	şe	ış	şı	iş	şi	uş	şu	üş	şü	oş	şo	öş	şö
t	at	ta	et	te	ıt	tı	it	ti	ut	tu	üt	tü	ot	to	öt	tö
v	av	va	ev	ve	ıv	vı	iv	vi	uv	vu	üv	vü	ov	vo	öv	vö
y	ay	ya	ey	ye	ıy	yı	iy	yi	uy	yu	üy	yü	oy	yo	öy	yö
z	az	za	ez	ze	ız	zı	iz	zi	uz	zu	üz	zü	oz	zo	öz	zö

[28] Two consonats can not come one after the other at the beginning of a word in Turkish. However, the words borrowed from other languages via cultural interactions may disturb this and suitable vowel insertion is essential for correct pronounciation.

Table 2.5 Synthesized Form Example

Text	Synthesized Form
üniversite (university)	ü+ni+ve+er+si+te
öğrenci (student)	öğ+re+en+ci
p(i)lan (plan)	pi+la+an
film (film)	fi+il+m
çift (double)	çi+if+t
espri (witticism)	es+p+ri

Two speech databases were prepared:

- The units are directly recorded
- The units are cut from large speech signals

Both of databases consist of alternative diphones for phonemes. The syllable *'ka'* in the word *kağıt (paper)* should be pronounced as *kậ-ğıt* by using a caret or circumflex, but on the other hand *'ka'* in the word *ka-lem (pen)* as it is written.

Second speech database contains units that are cut from large speech signals. There are three forms for each unit because phonemes are also affected by their position in the word. For example the 'şe' syllable:

- First form: **şe**-ker (sugar) (beginning)
- Middle form : şi-**şe**-ler (bottles) (middle)
- Last form : kö-**şe** (corner) (end)

Speech records that constitute the database are obtained by cutting them from the continue speech of the Turkish native speaker and 100 sentences that contains exemplars of all the phonemes and their forms are recorded as much as possible.

2.3 RECOGNIZE AND CLASSIFY PHONEMES AUTOMATICALLY

A typical speech sentence signal consists of two main parts: One carries the speech information, and the other includes silent or noise sections that are between the utterances, without any verbal information. The verbal (informative) part of speech can be further divided into two categories: The voiced speech and unvoiced speech (Lavner and Porat, 2002).

In literature (Cui, 2007) (Lavner and Porat, 2005) (Qi and Hunt, 1993) (Atal and Rabiner, 1976) there are some features of speech used to segment speech into smaller units corresponding to phonemes and to and the boundary between voiced, unvoiced and silience portions of speech.

Characteristic features for voiced/unvoiced/silence determination:

a. Short Time Zero Crossing Rate
b. Short Time Energy
c. Cross-correlation

a. Zero Crossing Rate (ZCR): ZCR occurs if successive samples have different algebraic signs (Cui, 2007). The value of ZCR is defined by the following equation (Rabiner and Schafer, 1978):

$$ZCR = \sum_{m=-\infty}^{\infty} \left| sign\left[x(m)\right] - sign\left[x(m-1)\right] \right| \cdot w(n-m) \tag{2.1}$$

where sign function is given by

$$\begin{aligned} &\text{sign}[\text{x(m)}] = 1;\ \text{x(m)} > 0 \\ &\text{sign}[\text{x(m)}] = 0;\ \text{x(m)} = 0 \\ &\text{sign}[\text{x(m)}] = -1;\ \text{x(m)} < 0 \end{aligned} \tag{2.2}$$

and

$$w(n) = \frac{1}{2N} \quad 0 \le n \le N-1 \tag{2.3}$$

Equation 2.1 to Equation 2.3 means that the ZCR is to check samples in pairs and find where the zero-crossings occur. If the zero-crossing rate is high, it indicates the speech is unvoiced, and if the zero-crossing rate is low, it indicates that the speech is voiced speech. So ZCR is quite useful in making the distinction between voiced and unvoiced speech (Cui, 2007).

b. Energy: Short-Time energy is a simple short-time speech measurement. It is defined as (Rabiner and Schafer, 1978):

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m) \cdot h(n-m) \tag{2.4}$$

In order for E_n to reflect the amplitude variations in time (for this a short window is necessary), and considering the need for a low pass filter to provide smoothing, h(n) was chosen to be a hamming window powered by 2. It has been shown to give good results in terms of reflecting amplitude variations (Rabiner and Schafer, 1978).

c. *Cross-correlation*. Cross-correlation is calculated between two consecutive pitch cycles. The cross-correlation values between pitch cycles are higher (close to 1) in voiced speech than in unvoiced speech (Lavner and Porat, 2002). Autocorrelation is handled as a special case of cross-correlation and correlation of signal with itself. The property of the short-time autocorrelation to reveal periodicity in a signal is demonstrated. It is notizable that how the autocorrelation of the voiced speech segment retains the periodicity. On the other hand, the autocorrelation of the unvoiced speech segment looks more like noise. In general, autocorrelation is considered as a robust indicator of periodicity (Nassos and Vassilis, 2004). Autocorrelation of voiced and unvoiced speech are shown in Figure 2.4

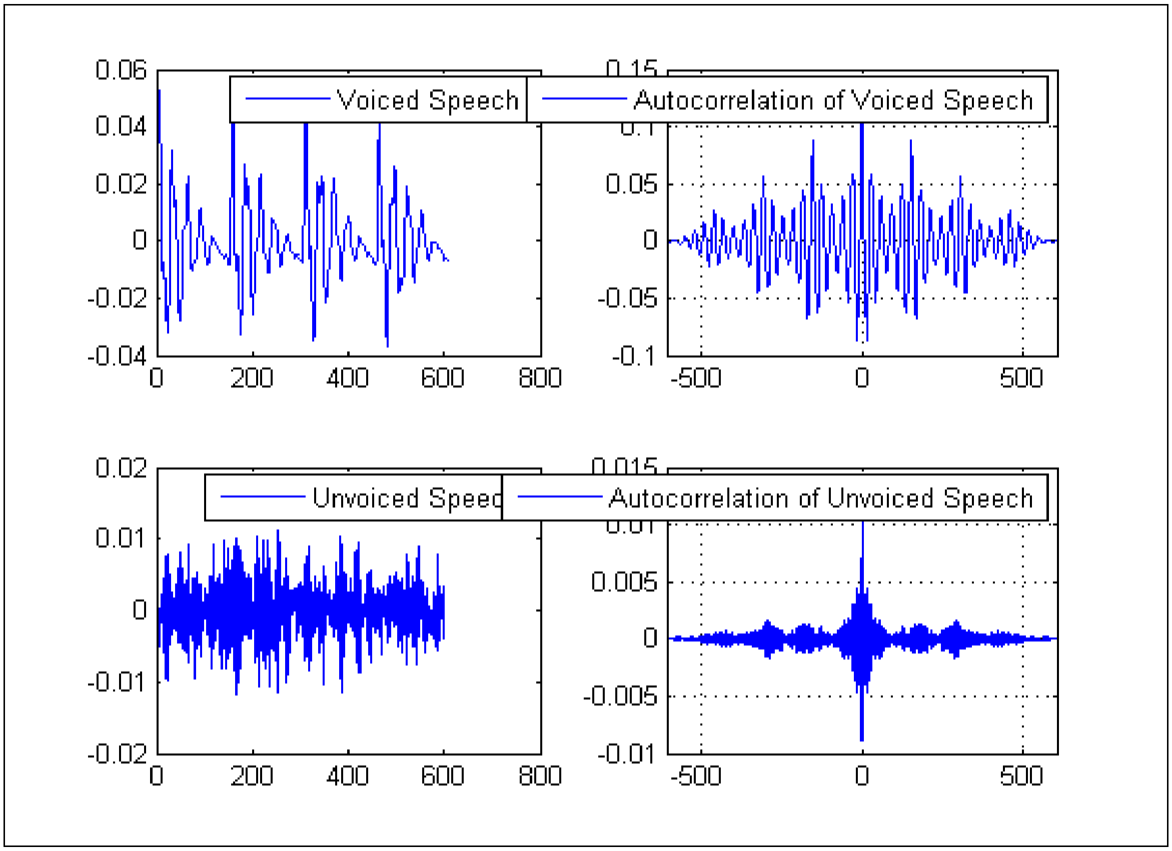


Figure 2.4 Autocorrelation Voiced and Unvoiced Speech (Nassos and Vassilis, 2004)

To decide about which segment is consonant and which segment is vowel, ZCR and energy are used. Basically, high ZCR means that the segment is consonant and high energy means that the segment is vowel phoneme. There is a correlation between zero-crossing rate and energy distribution with frequency; it is illustrated in the Figure 2.5. Classifying in literature used threshold of features. Many researches are rule-based. Benincasa and Savic, 1998 used threshold and Bayesian approach to classify.

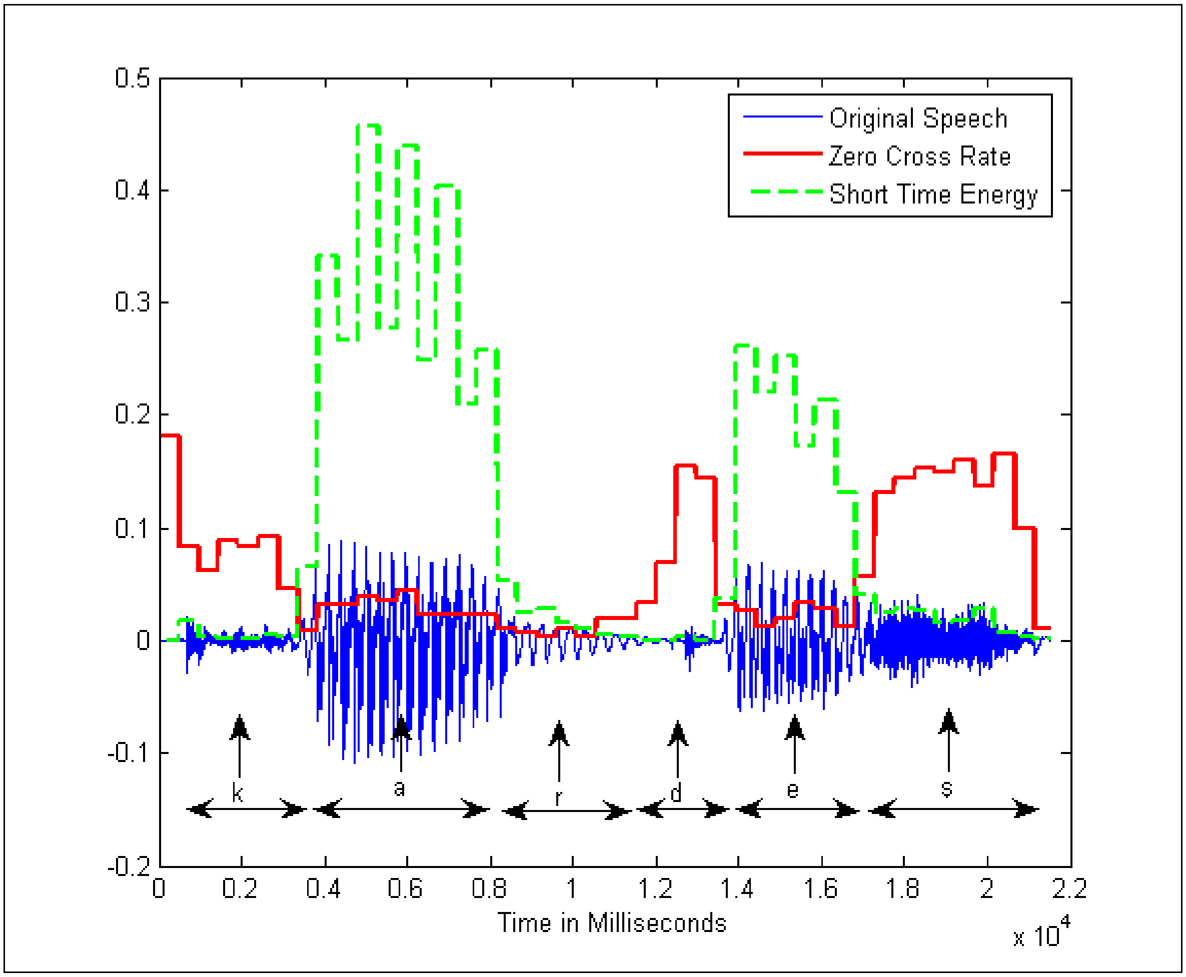


Figure 2.5 Correlation of Energy and ZCR Value of Each Frame in The Word *kardeş*

K* (Kstar) machine learning algorithm which is one of the sample based classificators in WEKA[29], is used in the current system because it was better than the other instance-based learners (Cleary and Trigg, 1995).

[29] WEKA: http://www.cs.waikato.ac.nz/ml/weka/

K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. Instance-based learners classify an instance by comparing it to a database of pre-classified examples. The fundamental assumption is that similar instances will have similar classifications (Cleary and Trigg, 1995).

WEKA (Waikato Environment for Knowledge Analysis) is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes (Witten and Frank, 2005).

Some values are used to mention the performance of results in WEKA. These are accuracy, confision matrix, precision and recall results.

Confusion Matrix; is a table with the true class in columns and the predicted class in rows. The diagonal elements represent correctly classified compounds while the cross-diagonal elements represent misclassified compounds (Kohavi and Provost, 1998).

Table 2.6 Confusion Matrix

		Actual	
		Negative	Positive
Predict	Negative	**A**	**B**
	Positive	**C**	D

Accuracy (Acc) is the degree of closeness of a measured or calculated quantity to its actual (true) value[30].

$$Acc = \frac{A + D}{A + B + C + D} \tag{2.5}$$

Presicion (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation (Kohavi and Provost, 1998):

$$P = \frac{C}{C + D} \tag{2.6}$$

[30] Accuracy: http://en.wikipedia.org/wiki/Accuracy_and_precision

Recall (R) is the proportion of positive cases that were correctly identified, as calculated using the equation(Kohavi and Provost, 1998):

$$R = \frac{D}{B + D} \tag{2.7}$$

There are four classes in train set: C (consonant), V (vowel), S (silence) and N (noise). Percentace of distrubition of classes are C: %43, V: %33, S: %24, N: 0,7. For testing, speech of words of iki (two), bin (thousand) and biraz (a few) are used. There are train and test accuracy results in Table 2.7 and presicion and recall results of each class in train and test steps in Table 2.8.

Table 2.7 Train and Test Accuracy Result

Instance	AC (%)
Train	95,4
iki (two)	77,7
bin (thousand)	76,9
biraz (a few)	81,5

Table 2.8 Presicion and Recall Results

		P				**R**		
Instance	**C**	**V**	**S**	**N**	**C**	**V**	**S**	**N**
Train	0,92	0,96	1	1	0,97	0,91	0,98	0,75
iki (two)	0,69	1	0	0	1	0,55	0	0
bin (thousand)	0,76	0,77	0	0	0,86	0,63	0	0
biraz (a few)	0,75	0,88	1	0	0,93	0,88	0,2	0

Incorrect class assignments generally occur during the transitions of C-V or V-C. This intrinsic problem also observed during the train data preparation process. The incorrect classifications can be improved by assigning different classes to the transition points.

CHAPTER 3

IMPLEMENTATION OF SPEECH SYNTHESIS

There are two major modules in a TTS system: NLP module that converts the written text input in phonetic transcription and DSP module that transforms the symbolic information into speech. Turkish TTS system is also formed from these two components that are explained in the following subsections.

3.1 TEXT-TO-PHONEME / NLP MODULE

The first step of text-to-phoneme is preprocessing. The preprocessing of the text to be converted into speech requires the followings:

- First, all unnecessary format information that does not contribute anything to the pronunciation are cleared and the redundant characters such as extra white spaces (\t, \n, \s) etc. are removed.
- Next, the text is normalized by considering upper/lowercase letters
- Then this step is followed by the syllabification phase. The Zemberek is used as a tool for this step in the system. The syllables that have more than two letters are derived from the smallest units. The words, which are generally borrowed from other languages throughout cultural interactions, present exceptional behaviors and should be handled specifically. The pseudocode for the main function of text-to-phoneme phase is shown in Table 3.1. The pseudocode of sub-functions for syllabification, checking epenthesis and abbreviation function are shown in Table 3.2, part A, B and C, respectively

Table 3.1. Pseudocode for Main Function of Text-to-Phoneme (Orhan and Görmez, 2009b)

```
Check spaces, convert extra spaces{\n,\t\r} into single space
Check liaison, convert ‘CspaceV’ into CV (delete space between C and V)
Check special character, convert character into SpaceCharacterSpace (put character between two spaces)
Check comma, do nothing if comma between two numbers otherwise convert comma into SpaceCommaSpace.
Check spaces again.
Tokenize text, divide text into word array considering space.
  For each token:
if token is a number use as number
else // as a word
   if it has one character
          if the character is a special character convert its vocalization
           else if vowel do nothing
           else if consonant concatenate with ‘e’ vowel
   else if has two letters
          if it  is in the form of  CV or VC syllable send zemberekSyllable
          else add abbreviation function result to syllable
```

Table 3.2. Pseudocode of Sub-Functions (Orhan and Görmez, 2009b)

```
function zemberekSyllable
  declare syllable as text
  convert word into syllable array with zemberek project
  if is not  syllable array add abbreviation function result to syllable
 for each syllable:
if it has less than three characters add it to syllable
else if it has three characters
   if it  is in CVC format add it as CV+C to syllable
   else if CCV format add abbreviation function result to syllable
           else if is VCC format check CCend set
          if is available add it as VC+C to syllable
          else add abbreviation function result to syllable
else if it has four letters(CVCC) check CCend set
   if is available add it as CV+C+C to syllable
   else add abbreviation function result to syllable
                                                                     (A)
```

```
function epenthesis
 if token matches CC[C|V]+ pattern check CCbegin
set
if available insert required vowel
                                          (B)
```

```
function abbreviation
 for each letter:
   if vowel do nothing
      else if consonant add ‘e’  vowel to  consonant
                                          (C)
```

There are some other issues that should be considered during the synthesis of the units. The liaison is the grammatical circumstance in which a usually silent consonant at the end of a word is pronounced together with the vowel at the beginning of the word that follows it. The examples of the liaison are pointed out by → in the following lines of a Turkish poem:

*“Dönülme**z →ak**şamın ufkundayız vakit çok geç*
*Bu son fasıldı**r →ey** ömrüm nasıl geçersen geç”*

In the above lines normal syllabification should result in

Dö-nül-mez ak-şa-mın or *fa-sıl-dır ey*

But due to the liaison they are synthesized as

Dö-nül-me zak-şa-mın or *fa-sıl-dı rey*

However, the following example is not a liaison, since the comma disturbs the rule. Therefore, the punctuations are required for liaison detection. (Liaison is not applicable at → point).

*Anne**m,** ↛ **a**blam geldi*

Special characters that are considered and ignored in synthesis are shown in Table 3.3. Comma is synthesized if between two numbers as in the case of 11,3 and synthesized as *onbir virgül üç (eleven comma tree)*, and ignored if it occurs between words. Exceptionally, the comma is not considered between two numbers when a space follows it. Therefore, 11, 3 is synthesized as two distinct numbers *on bir (eleven) üç (three).* Not as *onbir virgül üç (eleven comma tree)* as in the previous case.

CCend and *CCbegin* sets shown in Table 3.4 are possible combinations of two consecutive consonants that can occur at the end and the beginning of a syllable in Turkish. These sets are used for the detection of the abbreviations and epenthesis. If the syllable includes two consecutive consonants at the end, CCend set is checked for the valid combination. If it is valid, it is synthesized as a normal syllable; otherwise it is considered as an abbreviation. The word *fabl (fable)* includes two consonants at the end and synthesized as fa+b+l because ‘bl’ consonant combination is member of CCend set. On the other hand, *ABS (Anti-lock Braking System)* ending with two consonants which is not a valid combination, since ‘bs’ consonant combination is not member of CCend set, accepted as an abbreviation and synthesized as *A-Be-Se*.

The potential epenthesis is checked especially for foreign words via the CCbegin set. If the syllable includes two consecutive consonants at the beginning, CCbegin set is checked for the valid combination. If it is valid, additional sounds are inserted by using some heuristics and exceptional case rules. Examples are the word *grip (flu),* since *gr* is a member of CCbegin set, the word is converted to *gırip*, by inserting *ı* in between *g* and *r*, or *profesör (professor)* to *purofesör* etc.

Table 3.3 Vocalized and Not Vocalized Special Characters (Orhan and Görmez, 2008)

Character	Vocalization
	(alternatives are seperated by /)
@	Et
%	Yüzde
&	Ve
|	Veya
#	Diyez
*	Çarpı
/	Bölü
-	Eksi/Tire
+	Artı
>	Büyüktür
<	Küçüktür
(	Aç parantez
)	Kapa parantez
=	Eşittir
~	Yaklaşık
€	Yuro
$	Dolar
_	Alt çizgi
,	Virgül (for decimal number)/None
.	None
:	None
;	None
?	None
!	None
'	None
"	None

The input of this step is a string of Turkish and the output is obtained as arrays of words composed of one or two-letter components of the input that are going to be sent to the second step as the input for synthesis. More specifically if the input is *Bugün hangi gün? (What day is it?),* the output is returned as *[Bu, gü, n] [ha, n, gi] [gü, n]*.

Table 3.4 Possible Combinations of Two Consecutive Consonants at the End and Beginning of a Syllable in Turkish (1=Possible, 0=Impossible or Ignored) (Orhan and Görmez, 2008)

Position=End of syllable- Ccend																					
	b	**c**	**ç**	**d**	**f**	**g**	**ğ**	**h**	**j**	**k**	**l**	**m**	**n**	**p**	**r**	**s**	**ş**	**t**	**v**	**y**	**z**
b	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
c	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ç	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
g	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ğ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
l	0	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
m	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
n	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1
p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	1
s	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
ş	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
t	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0
y	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0
z	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Position=Beginning of syllable- Ccbegin																					
	b	**c**	**ç**	**d**	**f**	**g**	**ğ**	**h**	**j**	**k**	**l**	**m**	**n**	**p**	**r**	**s**	**ş**	**t**	**v**	**y**	**z**
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ç	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
ğ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0
r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0
ş	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3.5 Examples of Possible Combinations of Two Consecutive Consonants at the End and Beginning of a Syllable in Turkish(N/A=Not Applicable) (Orhan and Görmez, 2008)

Consec.	Example(Pos.=End)	English	Example(Pos.=Begin)	English
bd	abd	slave,servant	N/A	N/A
bl	fabl	fable	N/A	N/A
br	N/A	N/A	branş	branch
bt	zabt	restrain	N/A	N/A
cd	vecd	entrancement	N/A	N/A
dh	medh	praise	N/A	N/A
dr	N/A	N/A	dram	drama
fl	N/A	N/A	flüt	flute
fr	N/A	N/A	fren	brake
fs	nefs	essence,flesh	N/A	N/A
ft	çift	double,pair	N/A	N/A
gl	N/A	N/A	gladyatör	gladiator
gr	N/A	N/A	gram	gram
hd	ahd	vow	N/A	N/A
hr	N/A	N/A	hristiyan	christian
ht	baht	luck	N/A	N/A
kl	N/A	N/A	kla-sik	classic
kr	N/A	N/A	krem	cream
ks	lüks	luxury	N/A	N/A
kt	dir-rekt	direct	N/A	N/A
lç	felç	apoplexy	N/A	N/A
lf	golf	golf	N/A	N/A
lg	telg-raf	telegram	N/A	N/A
lh	sulh	peace	N/A	N/A
lk	ilk	first	N/A	N/A
lm	film	film	N/A	N/A
lp	kulp	handle	N/A	N/A
ls	vals	waltz	N/A	N/A
lt	alt	bottom	N/A	N/A
mb	amb-lem	emblem	N/A	N/A
mp	komp-leks	complex	N/A	N/A
mt	semt	district	N/A	N/A
nç	genç	young	N/A	N/A
nf	enf-las-yon	inflation	N/A	N/A
ng	mi-ting	meeting	N/A	N/A
nk	renk	color	N/A	N/A
ns	fi-nans	finance	N/A	N/A
nt	hint-li	indian	N/A	N/A
nz	bronz	bronze	N/A	N/A
pl	N/A	N/A	plan	plan
pr	N/A	N/A	pro-fe-sör	professor
ps	N/A	N/A	psi-ko-log	psychologist
rç	borç	debt	N/A	N/A
rd	ard	consecutive	N/A	N/A
rf	harf	letter	N/A	N/A
rg	morg	morgue	N/A	N/A
rh	zırh	armor	N/A	N/A
rj	şarj	charge	N/A	N/A
rk	ırk	race	N/A	N/A
rm	form	form	N/A	N/A
rn	mo-dern	modern	N/A	N/A
rp	sarp	steep	N/A	N/A

Consec.	Example(Pos.=End)	English	Example(Pos.=Begin)	English
rs	ders	lesson	N/A	N/A
rş	marş	march,anthem	N/A	N/A
rt	kort	court	N/A	N/A
rv	re-zerv	reserve	N/A	N/A
rz	arz	earth,supply	N/A	N/A
sf	N/A	N/A	sfenks	sphinx
sk	kask	helmet	skandal	scandal
sl	N/A	N/A	slayt	slide
sm	N/A	N/A	smo-kin	tuxedo
sp	esp-ri	witticism	spor	sports
sr	N/A	N/A	N/A	N/A
st	ast	junior	stres	stress
şk	aşk	amour,love	N/A	N/A
şt	ser-gü-zeşt	adventure	N/A	N/A
tf	lutf	grace	N/A	N/A
tm	ritm	rhythm	N/A	N/A
tr	fötr	felt	tren	train
vk	zevk	enjoyment	N/A	N/A
vr	sevr	Sèvres,ox	N/A	N/A
vs	Dos-to-yevs-ki	Dostoyevsky	N/A	N/A
vt	lo-kavt	lockout	N/A	N/A
yd	N/A	N/A	N/A	N/A
yh	a-leyh	against	N/A	N/A
yl	kok-teyl	cocktail	N/A	N/A
yn	di-zayn	design	N/A	N/A
yp	teyp	tape player	N/A	N/A
yr	seyr	wtach	N/A	N/A
ys	ays-berg	iceberg	N/A	N/A
yş	N/A	N/A	N/A	N/A
yt	la-kayt	uninterested	N/A	N/A
zm	fe-mi-nizm	feminism	N/A	N/A

3.2 PHONEME-TO-SPEECH / DSP MODULE

In this step the input is previous step's output as an array such as [*Bu, gü, n] [ha, n, gi] [gü, n]*. These array elements are mapped to their corresponding wav files. Selection of the appropriate file is achieved depending on the position of the component as in the case of syllable *şe* for the words *<u>şe</u>-ker (sugar)*, *şi-<u>şe</u>-ler (bottles)*, and *kö-<u>şe</u> (corner)*. If the syllable is C and previous syllable CV (gü, n), C convert VC (gü, ün) because for CVC syllables last consonant affected by previous vowel. The duration of words and silence parts are adjusted and the synthesis process is completed

During synthesize, the units are selected two different databases:

- From recorded database
- From cut database

For the 2nd database, there are three different units. First, middle and last. The units are selected according to the following rules:

- if the unit is first syllable; it is selected the first form: *__şe__-ker*
- if the unit is the middle syllable, it is selected the middle form: *şi-__şe__-ler*
- if the unit is the last syllable; it is selected the last form: *kö-__şe__*

In this step firstly, the units are selected as per rules above, secondly, the selected units are concatenated. Lastly, smoothing algorithm TD-PSOLA is applied to concatenated units.

Speech synthesis was implemented in three steps.

In the first step, firstly, the pitch detection was done by using matlab M-file, pitchmarker.m[31]. Secondly, the signal was segmented by using a Hanning window centered on the pitch mark. In the second step, selected units were concatenated. Thirdly, TD-PSOLA[32] algorithm was implemented in accordance with the speech signal, which is comprised of concatenation of units that are selected from two different speech databases. If the signal was comprised of the recorded database, the signal was compressed in time by discarding the segments. The segments were discarded in order to approximate the low speed of concatenated units, the recorded diphones, to the speed of continous speech. If the signal was comprised of the cut database, the signal was expanded in time by repeating the segments. The segments were repeated in order to approximate the high speed of concatenated units, the cut diphones, to the speed of continous speech.

3.3 PROGRAM INTERFACE

The program is able to make two different vocalizations by using two databases of different speed.

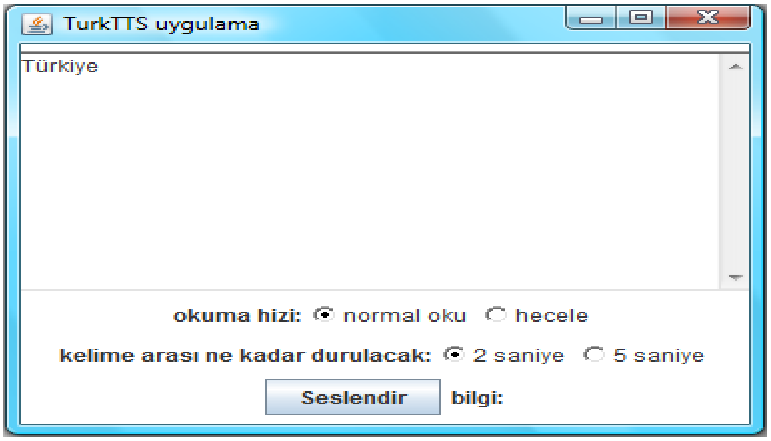
- The first one is to vocalize through syllabification by using recorded database.
- The second one is to vocalize at normal speech speed by using cut database.

[31] pitchmarker.m: http://www.ece.uvic.ca/~jpatton/yeshua1984/Elec484/Elec484_files/pitchmarker.m

[32] psola.m: http://www.ece.uvic.ca/~jpatton/yeshua1984/Elec484/Elec484_files/psola.m

Two different options were given to the user for pausing time between two succeeding words.

The Turkish TTS system was implemented has two user interfaces. The first program processes the text and produces phoneme from the text, and then produces speech from phoneme by using two different speech databases. This program interface was implemented by using JAVA. In Figure 3.1 first program interface is shown. The second program uses first program for text-to-phoneme level. And then produces speech from phoneme by using two different speech databases. After the speech is produced, it is re-synthesized by making use of TD-PSOLA algorithm. This second program interface was implemented by using MATLAB. In Figure 3.2 second program interface is shown



TurkTTS uygulama

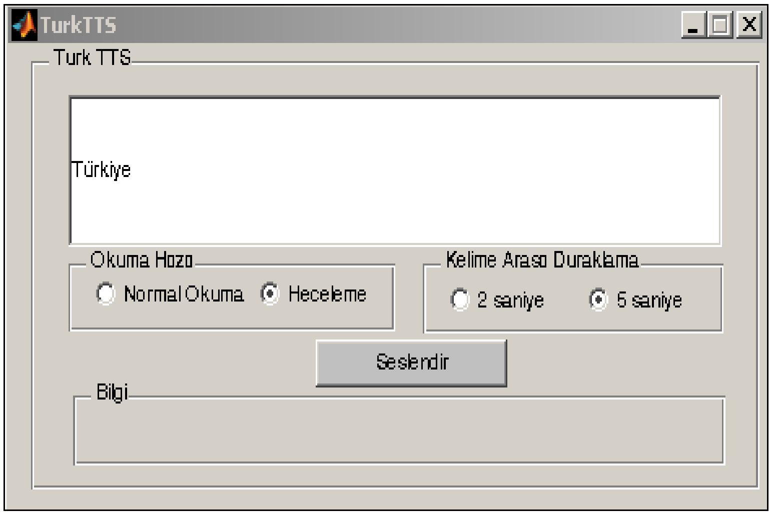
Türkiye

okuma hizi: ● normal oku ○ hecele

kelime arası ne kadar durulacak: ● 2 saniye ○ 5 saniye

Seslendir bilgi:

Figure 3.1 First Program Interface



TurkTTS

Turk TTS

Türkiye

Okuma Hızı ○ Normal Okuma ● Heceleme

Kelime Arası Duraklama ○ 2 saniye ● 5 saniye

Seslendir

Bilgi

Figure 3.2 Second Program Interface

CHAPTER 4

EVALUATION TESTS

The basic criteria for measuring the performance of a TTS system can be listed as the similarity to the human voice (*naturalness*) and the ability to be understood (*intelligibility*). The ideal speech synthesizer is both natural and intelligible, or at least try to maximize both characteristics. Therefore, the aim of TTS is also determined as to synthesize the speeches in accordance with natural human speech and clarify the sounds as much as possible. For overall quality evaluation, the International Telecommunication Union (ITU) recommends a specific method that, in this author's opinion, are suitable for also testing *naturalness* (ITU-T Recommendation P.85 1994).

Several methods have been developed to evaluate the overall quality or acceptability of synthetic speech (Lemmety 1999, Podsiadlo 2007). Diagnostic Rhyme Test (DRT), Comprehension Test (CT) and Mean Opinion Score (MOS) (Goldstein 1995) are the most frequently used techniques for the evaluation of the naturalness and the intelligibility of TTS systems. Naturalness and intelligibility of the Turkish TTS system is tested by MOS and CT-DRT respectively.

4.1 MEAN OPINION SCORE

The study was tested by making use of the MOS. The MOS that is expressed as a single number in the range 1 to 5, where 1 is lowest perceived quality and 5 is the highest perceived quality[33]. MOS tests for voice are specified by ITU-T recommendation. The

[33] MOS: http://en.wikipedia.org/wiki/Mean_Opinion_Score

MOS is generated by averaging the results of a set of standard, subjective tests where a number of listeners rate the perceived audio quality of test sentences read aloud by both male and female speakers over the communications medium being tested. A listener is required to give each sentence a rating using the rating scheme in Table 4.1. The perceptual score of the method MOS is calculated by taking the mean of the all scores of each sentence.

Table 4.1 Mean Opinion Score

MOS	**Quality**	**Impairment**
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

In the context of this study, 16 sentences that are provided in Table 4.2 are used for tests and 27 native Turkish speakers employed in the evaluation. For each sentence, MOS ranges that are assigned by the listeners are shown in Table 4.3. Distribution of MOS values for test sentences by testers and average MOS values for each sentence are given in Figure 4.1 and Figure 4.2 respectively.

Table 4.2 Test Sentences (Orhan and Görmez, 2009a)

SentN.	**Turkish**	**English**
S1	Bugün hangi gün?	What day is it ?
S2	2+8=	Two plus eight equals to
S3	Senin adın nedir?	What is your name?
S4	Beyazın zıddı nedir?	What is opposite of white?
S5	Türkiye'nin başkenti neresidir?	Where is the capital city of turkey?
S6	Çalıkuşunun yazarı kimdir?	Who is the author of çalıkuşu?
S7	Türkiye'nin, enbüyük gölü hangisidir?	Which is the biggest lake in turkey?
S8	Oniki eksi beş kaç eder?	What does twelve minus five amount to?
S9	Onbeş yaşındayım, orta okula gidiyorum	I am fifteen years old and a secondary school student.
S10	Seni dün çok aradım, evdemiydin.	I called you yesterday, were you home?
S11	Gelemem çok hastayım, yatıyorum.	I am too ill to come.
S12	Yemekte makarna var, yaşasın!	There is macaroni at lunch, yipee!
S13	Oda arkadaşın geldi ama hemen geri gitti	Your room mate came but he went back immediately.
S14	Ali erzuruma gitti, yarın gelecek.	Ali went to erzurum, he will come tomorrow.
S15	Eve gidelim hava fazla sıcak.	Let us go home, the weather is too hot.
S16	İngilizce çalışmadım matematik çalıştım.	I did'nt study english, ı studied mathematics.

Table 4.3 MOS Range and Scores for Each Sentence

SentN.	Excellent	Good	Fair	Poor	Bad			
	5	**4**	**3**	**2**	**1**	**sum listener**	**point sum**	**avg**
	2	11	12	2	-	27	**94**	**3,48**
S2	3	13	10	1	-	27	**99**	**3,67**
S3	2	13	11	1	-	27	**97**	**3,59**
S4	1	5	15	5	1	27	**81**	**3,00**
S5	3	14	9	1	-	27	**100**	**3,70**
S6	3	12	7	5	-	27	**94**	**3,48**
S7	3	11	12	1	-	27	**97**	**3,59**
S8	6	12	9	-	-	27	**105**	**3,89**
S9	3	8	12	4	-	27	**91**	**3,37**
S10	2	4	14	4	3	27	**79**	**2,93**
S11	4	6	13	4	-	27	**91**	**3,37**
S12	5	11	9	2	-	27	**100**	**3,70**
S13	1	9	10	7	-	27	**85**	**3,15**
S14	4	14	6	3	-	27	**100**	**3,70**
S15	1	4	10	9	3	27	**72**	**2,67**
S16	3	11	7	5	1	27	**91**	**3,37**
SUM	**46**	**158**	**166**	**54**	**8**	**432**	**1476**	**3,42**

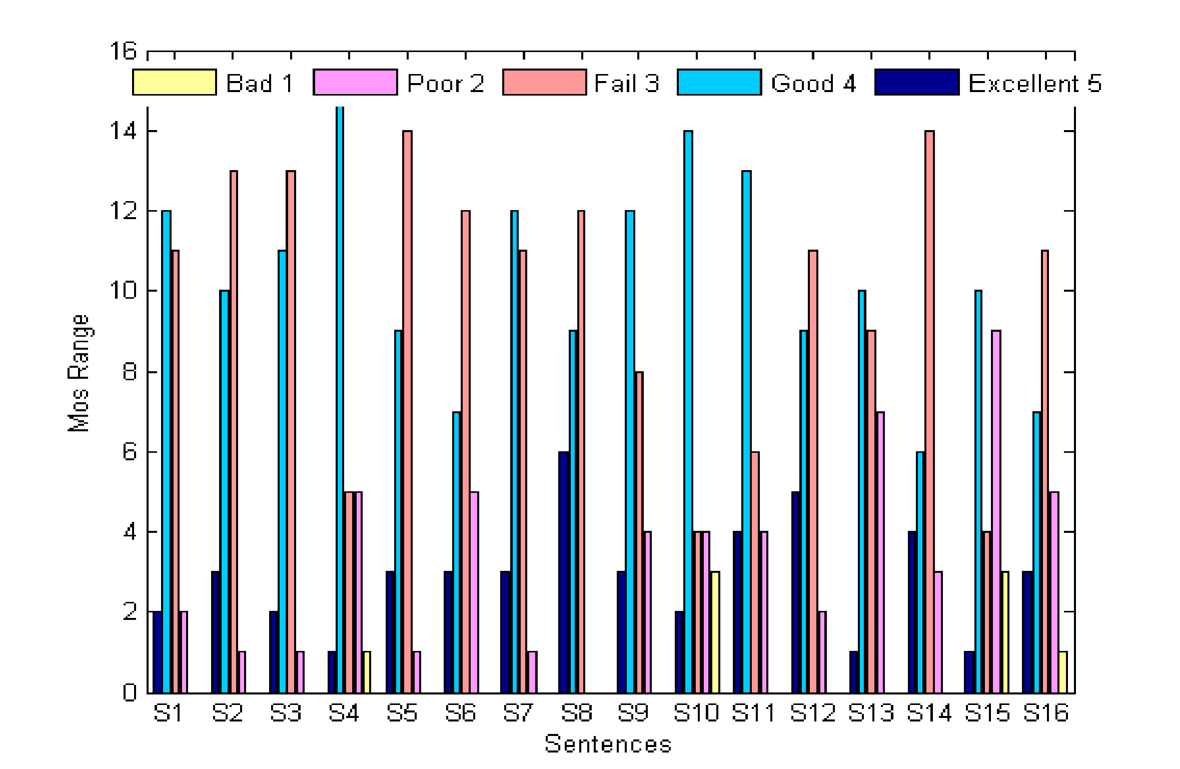


Figure 4.1 Distribution of MOS Values for Test Sentences Assigned by Testers

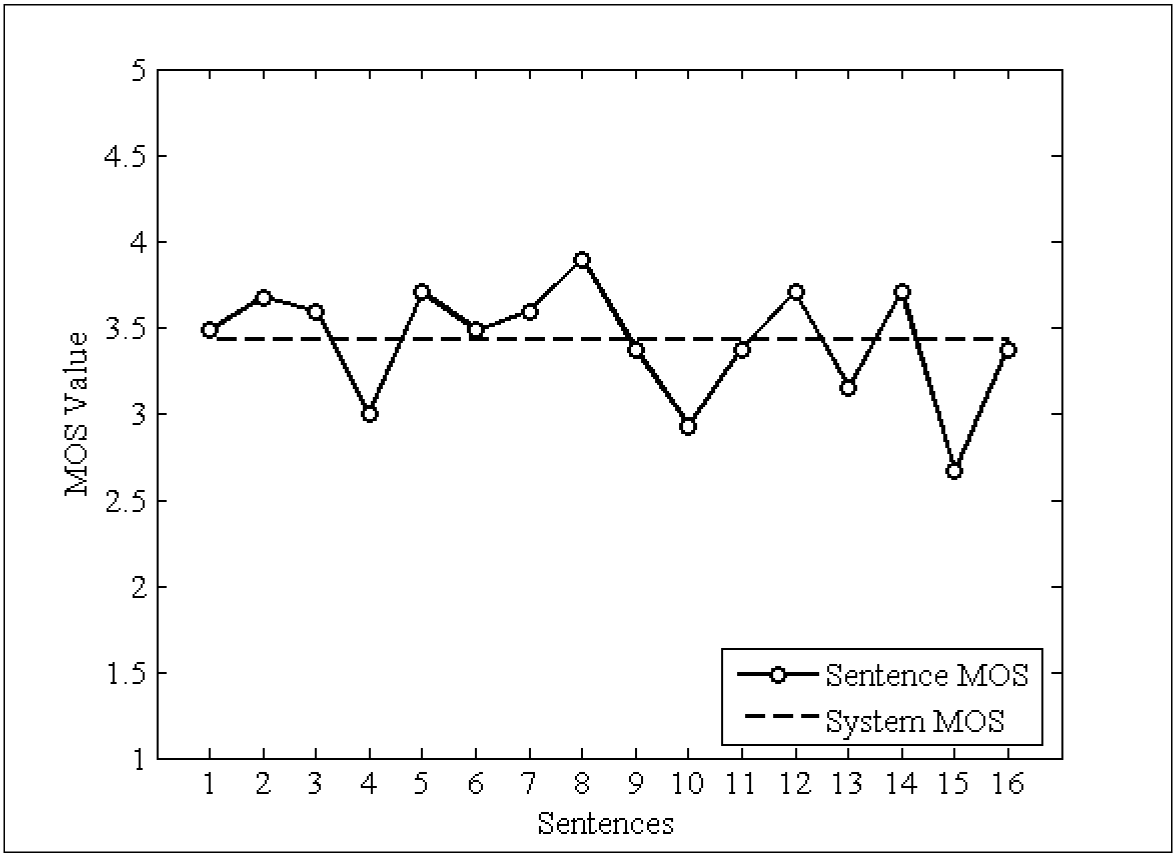


Figure 4.2 Average MOS Values for Each Sentence and System Average

4.2 COMPREHENSION TEST

In the comprehension tests, a subject hears a few sentences or paragraphs and answers the questions about the content of the text, so some of the items may be missed (Allen et al., 1987). It is not important to recognize one single phoneme, but the intended meaning. If the meaning of the sentence is understood, the 100% segmental intelligibility is not crucial for text comprehension and sometimes even long sections may be missed (Lemmety 1999) (Bernstein and Pisoni 1980).

In the comprehension tests three subtests are applied. In all three cases, the testers are allowed to listen to the sentences twice. In the majority of the tests, success is achieved for the first listening trial, and second one also improves the results.

First comprehension subtest has 8 sentences and 8 questions that are shown Table 4.4 about the content. Listeners answer the question about content of each sentence. First listening trial accuracy is calculated as the ratio of number of correct answers given by the testers to the whole set of correct answer as (T/N- 203/216) =0.94 and second trial accuracy is obtained as 1.

Second subtest is about answering common questions. It contains 8 sentences, S1-S8. Listeners answer the questions. Question sentences and number of correct answers at first and second listening are shown in Table 4.5. The results indicate that the understandability of the system is very high. Additionally, the accuracies of the first and second listening trials are 1.

Third subtest is applied as a filling in the blanks test. There are 8 noun phrases, one word of the phrase is provided to the listener and other is left as blank. The testers listened to the speech and filled in the blanks. Noun phrases and number of correct answers at first and second listening trial are shown in Table 4.6. The words that are underlined and given in capital letters are the blanks in the test. The accuracies of the first and second trial are 0.97 and 1, respectively by achieving a high understandability rate.

Table 4.4 Listening Comprehension (Orhan and Görmez, 2009a)

SentN.	Content question (Turkish)	Content question (English)	1st	Acc	2nd	Acc
S9	Konuşan kişi kaç yaşındadır?	How old is the speaker?	27	1	-	1
S10	Soran kişi hangi gün aramıştır?	When did the speaker call?	22	0,81	5	1
S11	Konuşan kişi neden gelemiyor?	Why can't the speaker come?	26	0,96	1	1
S12	Ne yemek yiyecekler?	What will they eat at lunch?	27	1	-	1
S13	Gelen kimdir?	Who came?	26	0,96	1	1
S14	Ali nereye gitmiş?	Where did Ali go?	27	1	-	1
S15	Neden eve gitmek istiyor?	Why does the speaker want to go home?	23	0,85	4	1
S16	Hangi dersi çalışmış?	Which course did the speaker study?	25	0,93	2	1

Table 4.5 Answering Common Questions (Orhan and Görmez, 2009a)

SentN.	Turkish	1st	Acc	2nd	Acc
S1	Bugün hangi gün?	27	1,00	-	1,00
S2	2+8=	27	1,00	-	1,00
S3	Senin adın nadir?	27	1,00	-	1,00
S4	Beyazın zıddı nedir?	26	0,96	1	1,00
S5	Türkiye'nin başkenti neresidir?	27	1,00	-	1,00
S6	Çalıkuşunun yazarı kimdir?	27	1,00	-	1,00
S7	Türkiye'nin, enbüyük gölü hangisidir?	27	1,00	-	1,00
S8	Oniki eksi beş kaç eder?	27	1,00	-	1,00

Table 4.6 Filling in the Blanks (Orhan and Görmez, 2009a)

Turkish	**English**	**1st**	**Acc**	**2nd**	**Acc**
İKİ bölüm	TWO section	26	0,96	1	1,00
Ali'nin KALEMİ	Ali's PENCIL	25	0,93	2	1,00
SEÇİM sonuçları	ELECTION results	26	0,96	1	1,00
kırmızı ÇİÇEK	red FLOWER	27	1,00	-	1,00
ÖĞLEDEN sonar	AFTERnoon	26	0,96	1	1,00
UZUN BOYLU çocuk	TALL child	26	0,96	1	1,00
küçük AHMET	young AHMET	26	0,96	1	1,00
çok SICAK	very HOT	27	1,00	-	1,00

4.3 DIAGNOSTIC RHYME TEST

Diagnostic rhyme test (DRT)[34] is an ANSI standard for measuring speech intelligibility (ANSI S3.2-1989). DRT, introduced by Fairbanks in 1958, uses a set of isolated words to test for consonant intelligibility in initial position (Lemmety, 1999), (Logan at el., 1989), (Goldstein 1995). DRT is used how the initial consonant is recognized properly. In the DRT test of the current system, the consonants that are similar to each other are selected and the listeners are asked to distinguish the correct consonant among the similar sounding alternatives. The letters that have the same way out such as ‘b’ and ‘p’ are plosive and bilabial consonant and can be easily misunderstood.

The similar sounding words that are used for DRT and number of correct answers for the first and the second listening trials are shown in Table 4.7. Bold words indicate the correct answers. Listeners choose one word from the table they hear. The accuracies are calculated above 0,90 and mostly close to 1 especially after the second trial.

The summary of accuracies of the CT and DRT tests are given in Table 4.8. It shows system intelligibility rate very high and satisfactory.

[34] ANSI S3.2, Method for measuring the intelligibility of speech over communication system. 1989

Table 4.7 Words for DRT (Orhan and Görmez, 2009a)

Recognize	**Listening**	**word 1**	**word 2**	**Acc**
m-n		**yaptım (I id)**	yaptın (you did)	
	1st	25	1	0,93
	2nd	25	2	0,93
z-s		zor (difficult)	**sor (ask)**	
	1st	2	25	0,93
	2nd	-	27	1,00
p-b		**prim (premium)**	birim (unit)	
	1st	26	-	0,96
	2nd	27	-	1,00
b-d		bilim (science)	**dilim (slice)**	
	1st	5	21	0,78
	2nd	1	25	0,93
l-r		**bilim (science)**	birim (unit)	
	1st	24	2	0,89
	2nd	27	-	1,00
v-f		**fan (fan)**	van (van)	
	1st	26	1	0,96
	2nd	27	-	1,00
l-r		deri (skin)	**deli (lunatic)**	
	1st	-	25	0,93
	2nd	-	27	1,00

Table 4.8 Comprehension Tests and DRT Accuracy (Orhan and Görmez, 2009a)

Tests	**Listening Trial Accuracies**	
Comprehension Test (CT)	**1st**	**2nd**
Answer the questions about the content	0,94	1
Answering common question	1	1
Fillings the blanks	0,97	1
Diagnostic Rhyme Test (DRT)	0,90	1

CHAPTER 5

CONCLUSION

In this study, the framework of a Turkish TTS that uses a concatenative synthesis approach is implemented and evaluated Although the system uses simple techniques, it provides promising results for Turkish, since the selected approach, the concatenative method, is very well suited for Turkish. This method is flexible enough to allow the synthesis of all types texts and the concatenation units are obtained from the atomic units.

The system can be improved by improving the quality of the speech files recorded. The sound files of news, films etc can be explored for extracting the recurrent sound units in Turkish instead of recording the diphones one by one. There are some ongoing projects about the analysis of speech signals for various applications. These projects can be helpful for obtaining wide ranges of phonemes in synthesis.

The punctuations are removed in the preprocessing step just to eliminate some inconsistencies and obtain the core system. In the future versions of the TTS, the text can be synthesized in accordance with the punctuations for considering the emotions and intonations as partially achieved in some of the researches. The synthesis of a sentence ending with a question mark can have an interrogative intonation and synthesis of a sentence ending with an exclamation mark can be an amazing intonation. In addition to these, other punctuations can be helpful for approximating the synthesized speech to its human speech form such as pausing at the end of the sentences ending with full stop and also pausing after the punctuation comma.

The evaluation process that yields high accuracies both for naturalness and intelligibility criterion is carried out by using the MOS, CT and DRT techniques as being the most frequently employed evaluation approaches in this field.

The capabilities of the system are as follows:

- All words can be vocalized , since the units are very small.
- Special characters can be vocalized
- Abbreviations can be recognized and vocalized.
- Dates can be recognized and vocalized.
- Currency values and numbers can be vocalized as default.
- Decimal numbers can be vocalized.
- Western-originated words such as train, plan, and professor can be vocalized correctly. For instance *plan* in English is vocalized as *pi* + *lan* in Turkish.
- The words borrowed from Arabic and Persian and which are pronounciated differently in Turkish can be vocalized successfully. To illustrate, the syllable of *kâ* in *kâğıt*,is vocalized with circumflex. Furthermore, *me* in *me:mur* is vocalized with long vowel *e*
- The system has fast or syllable by syllable (for new learners and children) reading option.

The sources of the previously implemented NLP applications are ought to be utilized by the coming researchers for building more comprehensive and better TTS systems. For instance, if there had not been Zemberek, a syllabication module would have had to implemented. Samely, if a speech database is developed for Turkish, researchers allocate their time used for developing a speech database to increase the quality of synthesized speech such as adding intonation and emotion. To exemplify, TIMIT[35] and Blizzard Challenge[36] databases are developed for English and researchers utilize these databases They can implement more comprehensive TTS applications with the help of these databases.

The TIMIT database is a continuous, speaker independent, phonetically-balanced and phonetically-labeled speech corpus developed by the Advanced Research Projects Agency (ARPA). TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus

[35] TIMIT, Acoustic-Phonetic Continuous Speech Corpus: www.ldc.upenn.edu/Catalog/LDC93S1.html

[36] Blizzard Challenge: http://festvox.org/blizzard/

includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16 kHz speech waveform file for each utterance (Garofolo, et al. 1993)

The Blizzard Challenge has been devised in order to better understand and compare research techniques in building corpus-based speech synthesizers on the same data. The basic challenge is to take the released speech database, build a synthetic voice from the data and synthesize a prescribed set of test sentences. The sentences from each synthesizer will then be evaluated through listening tests (Black and Tokuda, 2005).

In addition, if such a standard speech database exists for Turkish, a comparison between the systems using this database, susccess of the system is measured in this way as well. A comparitive measure can be accomplished for English TTS systems thanks to Blizzard Challenge.

Some standard sets should be assigned to evaluate the success of TTS systems, particularly in intelligibility measures such as DRT and MRT There are similar sets for English[37]. and researchers use them for intelligibility tests.

If two systems are compared exactly (perhaps to see if a particular change AB TESTS actually improved the system), AB tests can used. In AB tests,the same sentence synthesized by two different systems (an A and a B system), is played. The human listener chooses which of the two utterances they like better (Jurafsky and Martin, 2008).

[37] The 192 Stimulus Words of the DRT: http://www.meyersound.com/support/papers/speech/drtlist.htm

REFERENCES

Akın A.A., Akın M.D, *Zemberek, an open source NLP framework for Turkic Languages*, 2009, http://zemberek.googlecode.com/files/zemberek_makale.pdf.

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

Aşlıyan R., Günel K., "Türkçe Metinler İçin Hece Tabanlı Konuşma Sentezleme Sistemi", *Akademik Bilişim 2008,* Çanakkale Onsekiz Mart Üniversitesi, 2008.

Atal B.S., Rabiner L.R., "A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. Assp-24, No. 3, June 1976.

Benincasa D.S, Savic M.I., "Voicing State Determination of Co-Channel Speech", *IEEE ICASSP'98,* vol. 2, pp. 1021-1024, May 1998.

Bernstein J., Pisoni D., "Unlimited Text-to-Speech System: Description and Evaluation of a Microprocessor Based Device", *Proceedings of ICASSP 80 (3): 574-579*, 1980.

Beutnagel, M., A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, "The AT&T Next-Gen TTS system", *In The Proceedings of the Joint Meeting of ASA, EAA, and DAGA*, pp. 18-24, Berlin, Germany, 1999.

Black A., Taylor P., "CHATR: A Generic Speech Synthesis System". *COLING94,* Japan, 1994.

Black A., Taylor P., Caley R., *Festival Speech Synthesis System: System Documentation (1.4),* 1999, http://www.cstr.ed.ac.uk/projects/festival/manual/festival_toc.html.

Black A., Tokuda K., *The Blizzard Challenge 2005*, 2005, http://festvox.org/blizzard/.

Cerrato, L., *Introduction to speech synthesis*, 2005, http://stp.lingfil.uu.se/~matsd/uv/uv05/motis/lc_synt.pdf.

Cleary, J. G., Trigg, L. E., “K*: An Instance-Based Learner Using an Entropic Distance Measure”, *Proceedings Of The 12th International Conference On Machine Learning*, Tahoe City, California, USA, July 9-12, 1995, pp.108-114, 1995.

Cui Y., *Recognition of phonemes in a continuous speech stream by means of PARCOR parameters in LPC Vocoder*, Msc Thesis, University of Saskatchewan, 2007.

Dutoit T., Leich H., “MBR-PSOLA: Text-to-Speech Synthesis Based on an MBE Re-Synthesis of the Segments Database”, *Speech Communication*, vol. 13, pp. 435-440, 1993.

Dutoit T., Pagel V., Pierret N., Bataille F., Vrecken O., “The MBROLA Project: Towards a Set of High Quality Speech Synthesizers Free of Use for Non Commercial Purposes”, *Proceedings of ICSLP 96*, 1996.

Dutoit, T., *An Introduction to Text-To-Speech Synthesis*, Kluwer Academic Publishers, ISBN:1-4020-0369-2, 2001.

Eagleton, Maya B. *Reading the Web: Strategies for Internet Inquiry.* New York, NY, USA: Guilford Publications Incorporated, pp 59, 2006.

Garofolo J.S, et al., *TIMIT Acoustic-Phonetic Continuous Speech Corpus,* Linguistic Data Consortium, Philadelphia, 1993.

Goldstein M., “Classification of Methods Used for Assessment of Text-to-Speech Systems According to the Demands Placed on the Listener”, *Speech Communication* vol. 16, pp. 225-244, 1995.

González D., “Text-to-Speech Applications Used in EFL Contexts to Enhance Pronunciation”, *TESL-EJ*, September 2007.

Görmez Z., Orhan Z., "TTTS: Turkish Text-To-Speech System", *Proceedings of 12th WSEAS International Conference on Computers*, pp. 977-982, July 2008.

Güvenir H.A, “OKU 4.0: A Tool for Visually Handicapped People”, Microsoft Academic Days, 2005.

Honda M., “Human Speech Production Mechanisms”, *NTT Technical Review, Vol. 1 No. 2 May 2003*

Huang X., Acero A., Adcock J., Hon H., Goldsmith J., Liu J., Plumpe M., “Whistler: A Trainable Text-to-Speech System”, *Proceedings of ICSLP96,* 1996

Huang X., Acero A., Hon H., Ju Y., Liu J., Mederith S., Plumpe M., “Recent Improvements on Microsoft's Trainable Text-to-Speech System – Whistler”, *Proceedings of ICASSP97: 959-934*, 1997

Jurafsky, D., Martin, J.H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, 2008

Klatt, D. H., “The Klattalk text-to-speech conversion system”. *In IEEE ICASSP-82*, pp. 1589–1592, 1982.

Kohavi R., Provost F., “Special Issue on Applications of Machine Learning and the Knowledge Discovery Process”, *Machine Learning*, Vol.30, No.2-3, pp. 127-132, 1998.

Lavner Y., Porat G., *Voice Morphing*, SIPL – Technion IIT, 2002.

Lavner Y., Porat G., “VoiceMorphing using 3DWaveform Interpolation Surfaces and Lossless Tube Area Functions” EURASIP Journal on Applied Signal Processing 2005:8, pp. 1174–1184, 2005.

Lemmetty S., *Review of Speech Synthesis Technology*, MSc Thesis, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Finland, 1999.

Liddy, E.D., “Natural Language Processing”, *In Encyclopedia of Library and Information Science*, 2nd Ed. Marcel Decker, Inc., 2003.

Logan J., Greene B., Pisoni D., “Segmental Intelligibility of Synthetic Speech Produced by Rule”, *Journal of the Acoustical Society of America, JASA*, vol. 86 (2), pp. 566-581, 1989.

Maxey H.D., *Smithsonian Speech Synthesis History Project*, 2002, http://americanhistory.si.edu/archives/speechsynthesis/ss_spsyn.htm.

Moulines E., Charpentier F., "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones", *Speech Communication*, vol. 9, pp. 453-467, December 1990.

Nassos K., Vassilis P., *Speech Processing using MATLAB Part 1,* 2004 http://cvsp.cs.ntua.gr/courses/patrec/OnlineSpeechDemos/speechDemo_2004_Part1.html.

Orhan Z., Görmez Z., "The Framework Of The Turkish Syllable-Based Concatenative Text-To- Speech System With Exceptional Case Handling", *WSEAS Transactions on Computers*, ISSN: 1109-2750, Vol. 7, No. 10, pp. 1525-1534, Oct. 2008.

Orhan Z., Görmez Z., "Evaluation of the Concatenative Turkish Text-to-Speech System", (Accepted) *The 2nd International Conference on Image and Signal Processing (CISP'09),* 2009a.

Orhan Z., Görmez Z., "A Concatenative Turkish Text-to-Speech System and Evaluation Process", (Submitted) *6th International Conference on Electrical and Electronics Engineering (ELECO'09),* 2009b.

Patton J., *ELEC 484 Project: Pitch Synchronous Overlap-Add*, 2007. http://www.ece.uvic.ca/~jpatton/yeshua1984/Elec484/Elec484_files/ELEC 484 -PSOLA Final Project Report.pdf.

Podsiadlo, M., *Large Scale Speech Synthesis Evaluation,* MSc Thesis, University of Edinburg, 2007.

Qi Y., Hunt R.B., "Voiced-Unvoiced-Silence Classifications of Speech Using Hybrid Features and a Network Classifier"*, IEEE Transactions on Speech And Audio Processing*, Vol. 1, No. 2, April 1993.

Rabiner, L.R. and Schafer R.W., *Digital Processing of Speech Signals*, Prentice-Hall, Inc., Englewood Clis, New Jersey, 1978.

Reichel, U.D., Pfitzinger, H.R., "Text Preprocessing for Speech Synthesis", *In Proc. TC-Star Speech to Speech Translation Workshop*, pp 207-212, 2006.

Sanjaume, J.B., *Audio Time-Scale Modification in the Context of Professional Audio Post-production*, Phd. Thesis, Universitat Pompeu Fabra, Barcelona, 2002.

Shah, A.A., Ansari, A.W., and Das L., “Bi-Lingual Text to Speech Synthesis System for Urdu and Sindhi”, *National Conf. on Emerging Technologies NCET2004*, SZABIST, Karachi, Pakistan, Dec 2004.

Styger, T., Keller, E., “Formant synthesis In E. Keller (ed.)”, *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges,* Chichester: John Wiley, pp. 109-128, 1994.

Taylor P, *Text to Speech Synthesis*. Cambridge University Press, Cambridge, 2007, Draft, http://mi.eng.cam.ac.uk/pat40/ttsbook_draft_2.pdf.

Upperman, G., “Changing Pitch with PSOLA for Voice Conversion”, Connexions Web site. http://cnx.org/content/m12474/1.3/, Dec 17, 2004.

Utama, R.J, Syrdal, A.K., “Six Approaches to Limited Domain Concatenative Speech Synthesis”*, Interspeech*, Pittsburgh, Pennsylvania, 2006.

Waters, K., Levergood, T. M., *DECface: an automatic lip synchronization algorith for synthetic faces*, Technical Report CRL 93/4, DEC Cambridge Research Laboratory, Cambridge, MA, 1993.

Witten I.H., Frank E., *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Zhang, J., Language *Generation and Speech Synthesis in Dialogues for Language Learning,* MS thesis, Massachusetts Institute of Technology, 2004.

Zölzer U., *Digital Audio Effects*, John Wiley. & Sons Ltd, 2002.

Zue V. at el, “PEGASUS: a spoken dialogue interface for on-line air travel planning”, *Speech Communication*, v.15 n.3-4, p.331-340, Dec. 1994.