T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH

Volkan TUNALI

(Computer Engineering)

THESIS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

COMPUTER ENGINEERING PROGRAMME

SUPERVISOR
Prof. Dr. Murat DOĞRUEL

İSTANBUL 2005

T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH

Volkan TUNALI
(Computer Engineering)
(141100320020019)

THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING PROGRAMME

SUPERVISOR
Prof. Dr. Murat DOĞRUEL

İSTANBUL 2005

MARMARA UNIVERSITY

THE INSTITUTE FOR

GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

ACCEPTANCE AND APPROVAL DOCUMENT

A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH

Established committee listed below, on 18.07.2005 and 2005/15-8 by the *INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES*' Executive Committee, have accepted Mr.Volkan TUNALI 's Master of Science thesis, titled as "A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH" in Computer Engineering.

COMMITTEE

Advisor	:	Prof. Dr. Murat DOĞRUEL
Member	:	Prof. Dr. M. Akif EYLER
Member	:	Doç. Dr. Haluk TOPÇUOĞLU
Member	:	
Member	:	

Date of thesis' / dissertation's defense before the committee : 17.08.2005

APPROVAL

Mr. / Mrs. / Miss. ..... has satisfactorily completed the requirements for the degree of Doctor of Philosophy / Master of Science in ..... at Marmara University.
Mr. / Mrs. / Miss. ..... is eligible to have the degree awarded at our convocation on ..... Diploma and transcripts so noted will be available after that date.

Istanbul **DIRECTOR**

Prof. Dr. Adnan AYDIN

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Murat DOĞRUEL, for his encouragement and support on this project. He has helped me throughout the project with his invaluable original ideas.

I would like to thank Mr. Ozan MUT for his contributions to this project with his ideas and with his great effort to conduct experiments.

I would also like to thank my beloved wife Mrs. Sevdiye TUNALI for her endless relief and patience during my long work sessions on this project.

July, 2005 **Volkan TUNALI**

TABLE OF CONTENTS

ABSTRACT

A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH

The advances in digital signal processing technology has led the use of speech processing in many different application areas like speech compression, enhancement, synthesis, and recognition. In this thesis, the issue of speech recognition was studied and a speaker dependent, large vocabulary, isolated word speech recognition system was developed for Turkish Language.

A combination of two common approaches to speech recognition problem was used in the project: Acoustic-phonetic approach and stochastic approach. The phonemes modeled by two-state Hidden Markov Models (HMM) were used as the smallest unit for recognition. Mel-Frequency Cepstral Coefficients (MFCC) was preferred as the feature vector extraction method. A new algorithm was devised for phoneme detection and segmentation used in the training stage. Using phoneme-based recognition, the words that are not trained can be recognized by the system.

Keywords: Turkish speech recognition, phoneme based speech recognition, Hidden Markov Model (HMM), Mel-Frequency Cepstral Coefficients (MFCC), speech feature vector.

July, 2005 **Volkan TUNALI**

ÖZET

TÜRKÇE İÇİN KONUŞMACI BAĞIMLI, GENİŞ SÖZCÜK DAĞARCIKLI, AYRIK SÖZCÜKLÜ KONUŞMA TANIMA SİSTEMİ

Sayısal sinyal işleme teknolojisindeki gelişmeler, sinyal işlemenin ses sıkıştırma, geliştirme, sentezleme ve tanıma gibi çok değişik ve çeşitli alanlarda kullanımına yol açmıştır. Bu tez kapsamında, konuşma tanıma problemi ele alınmış ve Türkçe için konuşmacı bağımlı, geniş sözcük dağarcıklı, ayrık kelime konuşma tanıma sistemi geliştirilmiştir.

Projede, konuşma tanıma problemine iki genel yaklaşımın bir birleşimi kullanılmıştır: akustik-fonetik yaklaşım ve stokastik yaklaşım. Tanımadaki en küçük birim olarak iki durumlu Saklı Markov Modelleriyle (SMM) modellenmiş fonemler kullanılmıştır. Ses sinyalinden özellik vektörü çıkarım yöntemi olarak Mel-frekansı Kepstral Katsayılar (MFKK) tercih edilmiştir. Sistemin eğitimi aşamasında kullanılmak üzere fonem tespiti ve kesimlemesi için yeni bir algoritma geliştirilmiştir. Fonem tabanlı tanıma kullanılarak, sistemde eğitilmemiş sözcüklerin de tanınabilmesi sağlanmıştır.

Anahtar sözcükler: Türkçe konuşma tanıma, fonem tabanlı konuşma tanıma, Saklı Markov Modelleri (SMM), Mel-frekansı Kepstral Katsayılar (MFKK), ses özellik vektörü.

Temmuz, 2005 **Volkan TUNALI**

CLAIM FOR ORIGINALITY

A SPEAKER DEPENDENT, LARGE VOCABULARY, ISOLATED WORD SPEECH RECOGNITION SYSTEM FOR TURKISH

For languages like English, many speech recognition systems have been developed and have found several applications in real life. However, there have been too few attempts for a speech recognition system for Turkish language. Moreover, there is no commercially accepted large vocabulary speech recognition application in Turkish today. This has several reasons, and the most important one is that unlike Indo-European languages, Turkish has an agglutinative and suffixing morphology, which results in a very large vocabulary. This thesis is a significant attempt to develop such a system that many Turkish computer users can benefit from.

In this thesis, due to the morphological structure of Turkish language, a phoneme-based speech recognition system was developed rather than a triphone-based system which is a very common approach for large vocabulary speech recognition systems today. The main reason for this selection is that a triphone-based system requires quite larger number of triphone template models whereas a phoneme-based system requires very little number of template models. Moreover, a triphone-based system is usually used in order to eliminate the negative co-articulation effects during phoneme transitions. However, by means of good phoneme segmentation, a phoneme-based system can perform well without being much affected by co-articulation. A genuine algorithm was developed for phoneme detection and segmentation. This algorithm has the key importance because the performance of the phoneme segmentation is directly effective on the performance of the recognition. The phonemes were modeled using 2-state HMM models generated from 22-dimensional feature vectors obtained by applying MFCC on raw speech signal.

The speaker dependent, isolated word speech recognition system developed in this thesis is an important milestone through the way to a speaker independent, continuous speech recognition system for Turkish language.

July, 2005 **Prof. Murat DOĞRUEL** **Volkan TUNALI**

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION AND OBJECTIVES

Speech is the most natural way to communicate for humans. While this has been true since the dawn of civilization, the invention and widespread use of the telephone, audio-phonic storage media, radio, and television has given even further importance to speech communication and speech processing [2].

The advances in digital signal processing technology has led the use of speech processing in many different application areas like speech compression, enhancement, synthesis, and recognition [4]. In this thesis, the issue of speech recognition is studied and a speech recognition system is developed for Turkish Language.

Speech recognition can simply be defined as the representation of a speech signal via a limited number of symbols. The aim here is to find the written equivalent of the signal [5].

Why does the speech recognition problem attract researchers and funding? If an efficient speech recognizer is produced, a very natural human-machine interface would be achieved. By natural one means something that is intuitive and easy to use by a person, a method that does not require special tools or machines but only the natural capabilities that every human possesses. Such a system could be used by any person able to speak and will allow an even broader use of machines, specifically computers. This potentiality promises huge economical rewards to those who learn to master the techniques needed to solve the problem, and explains the surge of interest in the field during the last 15 years [7].

If an efficient speech recognition machine is enhanced by natural language systems and speech producing techniques, it would be possible to produce

computational applications that do not require a keyboard and a screen. This would allow incredible miniaturization of known systems facilitating the creation of small intelligent devices that can interact with a user through the use of speech. An example of this type of machines is the Carnegie Mellon University JANUS system [8] that does real time speech recognition and language translation between English, Japanese and German. A perfected version of this system could be commercially deployed to allow future customers of different countries to interact without worrying about their language differences. The economical consequences of such a device would be gigantic [7].

In this thesis,

- a speaker dependent, large vocabulary, isolated word speech recognition system for noise-free environments is developed for Turkish language,
- a new segmentation algorithm which allows for simple and good phoneme detection is presented, and
- a GUI application is implemented, which makes use of the recognition system developed.

Chapter II introduces the general view of a speech recognition system, including classification of, and approaches to speech recognition systems.

In Chapter III, production and representation of speech signal are given along with some other essential acoustic features of speech signal.

Chapter IV is devoted to speech processing techniques used in speech recognition. Specifically, along with an introduction to Digital Signal Processing (DSP), techniques for feature extraction step, and the Mel Frequency Cepstral Coefficients (MFCC) are defined.

Chapter V discusses the morphological structure of Turkish language which makes the recognition system for Turkish a challenging task, as well as the details of the phoneme-based recognition chosen for this system.

In Chapter VI, techniques and algorithms used in the training and recognition stages are explained in some detail.

Chapter VII presents a sample training and recognition session using the GUI application developed for the thesis along with several screenshots.

Chapter VIII displays and discusses the test results of the system.

Chapter IX gives the conclusions and the directions for future research.

CHAPTER II

SPEECH RECOGNITION

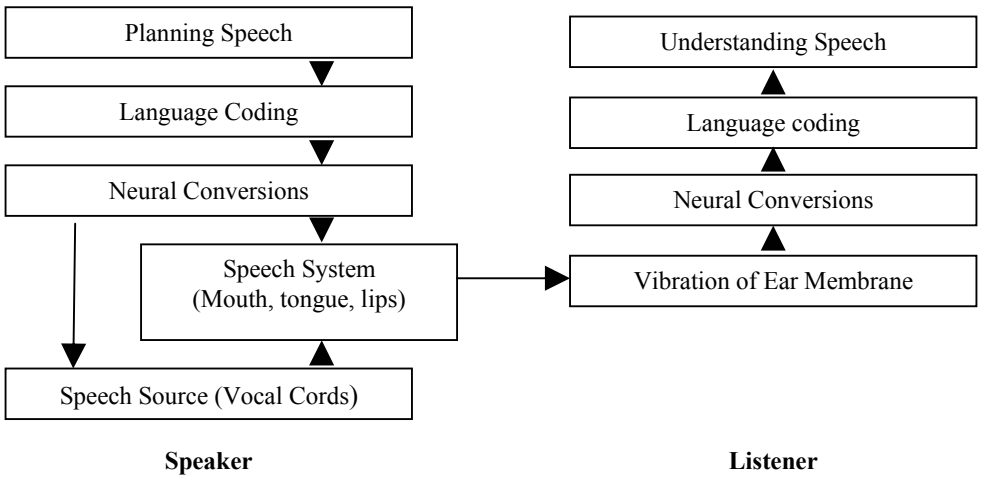
Speech recognition presents great advantages to human-computer interaction. It is easy to obtain speech data, and it does not require special skills like using keyboard, entering data via clicking the buttons on the GUI programs, and so on. Transferring text data into electronical media using speech is about 8-10 times faster than hand-writing, and about 4-5 times faster than using keyboard by the most skilled typist. Moreover, the user can continue entering text while moving or doing any work that requires her to use her hands. Since a microphone or a telephone can be used, it is more economic to enter data, and it is possible to enter data from a remote point via telephone.

Besides its advantages mentioned above, there are several challenges in speech recognition. Those may change according to the aim of use, and to the methods used. For example, speech data is very vulnerable to noise; therefore, the speech signal needs to be filtered using some noise filtering methods.

The speech communication among human-beings can be modeled as in Figure II.1 [9].

Speech communication process can be summarized as follows:

- Conversion of speaker's ideas into words
- Generation of voice of the words using the vocal cords and speech system
- Transmission of voice to the ear of the listener as vibrations
- Transmission of voice to brain via auditory nerves of the listener and conversion of those vibrations to language code equivalent by the brain
- Extraction of meaning from those codes, words, gathered.



FigureII.1. Speech Communication among Human-beings [9].

Humans can more effectively understand artificially-generated speech than machines can understand human speech because of the experience on and understanding of speech by humans. This experience makes humans more tolerant to speech containing error. Humans can correct errors in the speech using their knowledge of grammar and skill of understanding the speech spoken by different speakers.

The main goal of a speech recognition system is to substitute for a human listener, although it is very difficult for an artificial system to achieve the flexibility offered by human ear and human brain. Thus, speech recognition systems need to have some constraints. For instance, number of words is a constraint for a word-based recognitions system. In order to increase the performance of the recognition, the process is dealt with in parts, and researches are concentrated on those parts. This approach of splitting the process into parts provide better performance achievement for each of the parts, thus resulting in increased overall performance.

Speech recognition is the process of extraction of linguistic information from speech signals. The linguistic information which is the most important element of speech is called phonetic information.

A classical speech recognition system can be modeled as in Figure II.2. In this model, no change is made in the recognition process at the block borders, but frequent returns are needed between the blocks. These returns are for making the system more flexible and more consistent [5].

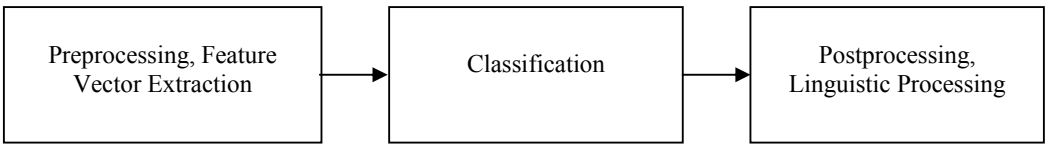


Figure II.2. Speech Recognition Process [5].

In the speech recognition model in Figure II.2, role of the preprocessing and classification modules is the conversion of speech signal into language symbols. Then, the last step is the generation of meaningful linguistic elements; that is, words and sentences.

The work principle of speech recognition systems is roughly based on the comparison of input data to prerecorded patterns. These patterns can be arranged in the form of phoneme or word. By this comparison, the pattern to which the input data is most similar is accepted as the symbolic representation of the data. It is very difficult to compare raw speech signals directly. Because the intensity of speech signals can vary significantly, a preprocessing on the signals is necessary. This preprocessing is called *Feature Extraction*. First, short time feature vectors are obtained from the input speech data, and then these vectors are compared to the patterns classified prior to comparison. The feature vectors extracted from speech signal are required to best represent the speech data, to be in size that can be processed efficiently, and to have distinct characteristics.

Dynamic structure of speech is the most important problem encountered in speech recognition systems. There are great differences in the speech signals of the same content when uttered by different persons. In addition, stress, intensity, and pitch level of the speech greatly vary at different regions of the signal. It is very possible that the speech of the same content is uttered very differently by the same person. Transitions between phonemes may carry more information than the stationary regions of the signal due to these variations. Thus, obtaining a very clear distinction of speech is the main goal of the feature vector extraction.

Essentially, speech is very similar to written expression. That is, speech is composed of subsequent voice fragments. Similarly, in written expressions, these voice fragments are replaced with the symbols of the language, letters.

The smallest unit used in the symbolic representation of speech is called *phoneme*. Phonemes do not have meanings alone. However, they are useful for distinguishing the words. This property of phonemes can be sampled as below.

KAR KİR

KALE LALE

In the samples, the words KAR and KİR are distinguished with the symbols A and İ. These symbols are called phonemes because they cause the meaning to change. Phoneme is a unit used for speech but it is not the same as letter. Turkish is a phoneme-based (*phonemic*) language; therefore, there is an exact phoneme equivalent of each letter in Turkish. However, this situation can be different in other languages. For example, although there are 26 letters in English, there are about 40 phonemes.

The speech unit used in voicing a phoneme is called a *phone*. A phone is used in the realization of a phoneme, and it does not have distinction. Speech recognition can also be defined as the conversion of phones to phonemes. As the result of such a conversion, the phonetic representation of the speech is obtained. Transition from this phonetic representation to words and sentences is performed by implementing several grammar rules. The conversion of phones to phonemes and basing the recognition on this conversion is called phoneme-based speech recognition. Rather than a single phoneme, units chosen for recognition can be *diphone* (2 phonemes), *triphone* (3 phonemes), *syllable*, and whole *word*. The larger the unit selected, the greater the processing amount, but the better the recognition accuracy.

Another important point here is that it is necessary to put limits on the system according to the size of the recognition unit. For example, it is inevitable to limit the word number in a word-based recognition system. On the other hand, word number is unlimited in a phoneme-based recognition system.

In this thesis, phonemes are the building blocks of the speech recognition system developed. In a phoneme-based system, there is no limit to word number. Furthermore, the system can be thought as independent of language. For another language, it is enough to change the pattern phonemes. However, performance of this system is dependent on the performance of the conversion of phonemes to language symbols.

An important matter for a phoneme-based speech recognition system is the transition effects occurring between sequential phonemes called *co-articulation*. During these transitions, due to the structure of human vocal chords and throat,

articulation of one phoneme has not ended yet when the articulation of next phoneme is started. Therefore, a definite boundary cannot be determined between phonemes, which poses a disadvantage in the segmentation of speech.

Another problem encountered in speech recognition is the variations in the utterances of the words. While the utterances of words can be different among different speakers, they can be different according to the content of speech, and according to the noise level of the environment.

II.1. CLASSIFICATION OF SPEECH RECOGNITION SYSTEMS

It is possible to classify speech recognition systems at different levels. According to the continuity of the speech;

- In *Isolated Word Recognition*, words that are uttered with short pauses are recognized.
- In *Continuous Speech Recognition*, words that are uttered continuously without having pauses between them are recognized.

Continuous Speech Recognition system can be divided into two groups as *Connected Word Recognition*, and *Conversational Speech Recognition*. The former aims at performing the recognition word-by-word, however, the latter aims at understanding the meaning of the sentence. Therefore, Conversational Speech Recognition systems are also called *Speech Understanding* systems, and they require the use of complex grammar rules in the system. This is dealt with in another computer science field called *Natural Language Processing/Understanding – NLP*.

Besides being isolated or continuous, speech recognitions systems are divided into two categories according to dependence on speaker;

- *Speaker Dependent*
- *Speaker Independent*

For the first one, reference patterns are constructed for a single speaker. In order for system to recognize the speech of different speakers, the reference patterns must be updated for new speakers. In the second, however, the system is able to recognize the speech of any speaker. Unfortunately, it is much more difficult to develop such a system than a speaker-dependent one.

Another classification can be made according to the size of the recognition unit chosen, as mentioned before. Speech recognition systems can be separated into two groups;

- In a *Word Based Speech Recognition*, the smallest recognition unit is a word.
- *Phoneme Based Speech Recognition* systems are the ones that use phonemes as the recognition units.

Recognition accuracy of the first one is very high because the system is free from negative side effects of co-articulation. However, for continuous speech recognition, transition effects between words again cause problems. Moreover, for a word-based recognition system, processing time and memory requirements are very high because there are many words in a language which are the bases of the reference patterns. In a phoneme-based system, while recognition accuracy decreases, it is possible to apply error-correction using the ability to produce fast results with very few phoneme numbers. There can be several speech recognition systems that make use of sub-word units like diphone-based, triphone-based, and syllable-based.

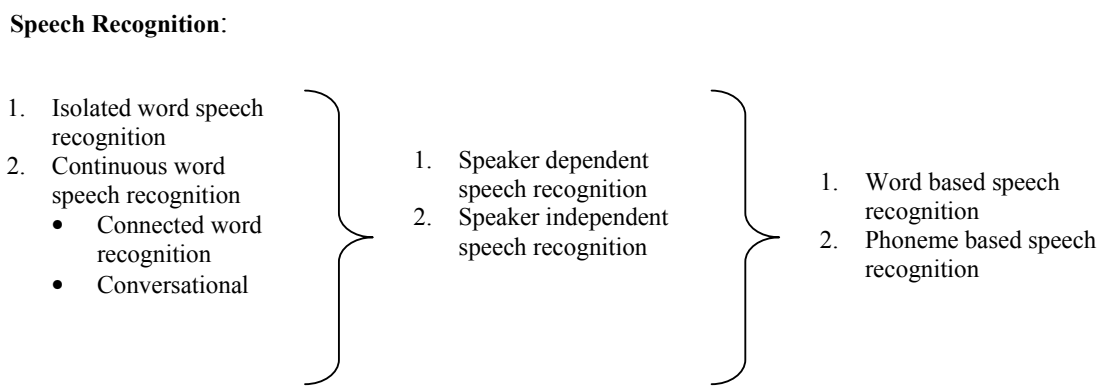


Figure II.3. Classification of Speech Recognition Systems [5].

II.2. APPROACHES TO SPEECH RECOGNITION

There are three competing recognition technologies found in commercial speech recognition systems. These are: [3]

- *Template matching*
- *Acoustic-phonetic recognition*
- *Stochastic processing*

These approaches differ in speed, accuracy, and storage requirements.

II.2.1. Template Matching

Template matching is a form of pattern recognition. It represents speech data as sets of feature/parameter vectors called *templates*. Each word or phrase in an application is stored as a separate template. Spoken input by end users is organized into templates prior to performing the recognition process. The input is then compared with stored templates, as Figure II.4 indicates; the stored template most closely matching the incoming speech pattern is identified as the input word or phrase. The selected template is called the *best match* for the input. Template matching is performed at the word level and contains no reference to the phonemes within the word. The matching process entails a frame-by-frame comparison of spectral patterns and generates an overall similarity assessment for each template [3].

The comparison is not expected to produce an identical match. Individual utterances of the same word, even by the same person, often differ in length. This variation can be due to a number of factors, including difference in the rate at which the person is speaking, emphasis or emotion. Whatever the cause, there must be a way to minimize temporal differences between patterns so that fast and slow utterances of the same word will not be identified as different words. The process of minimizing temporal/word length differences is called *temporal alignment*. The approach most commonly used to perform temporal alignment in template matching is a pattern-matching technique called *dynamic time warping* (*DTW*). DTW establishes the optimum alignment of one set vectors (template) with another [3].

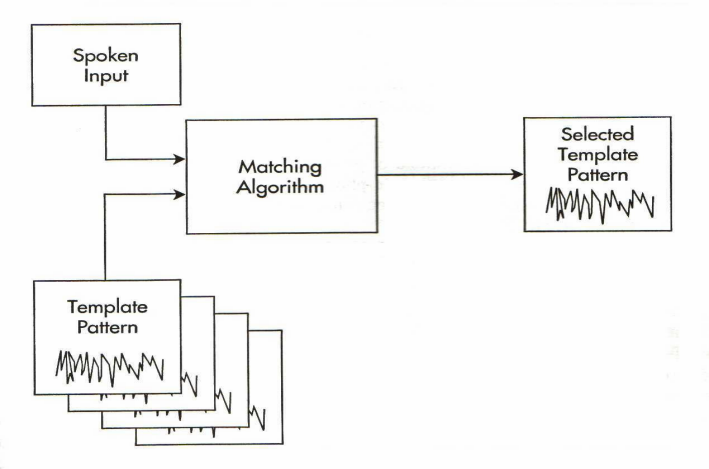


Figure II.4. Recognition Using Template Matching [3].

Most template matching systems have a predetermined threshold of acceptability. Its function is to prevent noise and words not in the application vocabulary from being incorrectly identified as acceptable speech input. If no template match exceeds the threshold of acceptability no recognition is recorded. Applications and systems differ on how such non-recognition events are handled. Many systems ask the user to repeat the word or utterance [3].

Template matching performs very well with small vocabularies of phonetically distinct items but has difficulty making the fine distinctions required for larger vocabulary recognition and recognition of vocabularies containing similar-sounding words (called *confusable* words). Since it operates at the word level there must be at least one stored template for each word in the application vocabulary. If, for example, there are five thousand words in an application, there would need to be at least five thousand templates [3].

II.2.2. Acoustic-Phonetic Recognition

Unlike template matching, acoustic-phonetic recognition functions at the phoneme level. Theoretically, it is an attractive approach to speech recognition because it limits the number of representations that must be stored to the number of phonemes needed for a language. For English, that number is around forty no matter how large the application vocabulary is [3].

Acoustic-phonetic recognition generally involves three steps:

- *Feature-extraction*
- *Segmentation and labeling*
- *Word-level recognition*

During feature extraction the system examines the input for spectral patterns, such as formant frequencies, needed to distinguish phonemes from each other. The collection of extracted features is interpreted using acoustic-phonetic rules. These rules identify phonemes (*labeling*) and determine where one phoneme ends and the next begins (*segmentation*) [3].

The high degree of acoustic similarity among phonemes combined with phoneme variability resulting from co-articulation effects and other sources create uncertainty with regard to potential phoneme labels. As a result, the output of the segmentation and labeling stage is a set of phoneme hypotheses. These hypotheses can be organized into a phoneme lattice, decision tree, or similar structure [3].

Once the segmentation and labeling process has been completed, the system searches through the application vocabulary for words matching the phoneme hypotheses. The word best matching a sequence of hypotheses is identified as the input item [3].

II.2.3. Stochastic Processing

The term *stochastic* refers to the process of making a sequence of *non-deterministic* selections from among sets of alternatives. They are non-deterministic because the choices during the recognition process are governed by the characteristics of the input and not specified in advance. The use of stochastic models and processing permeates speech recognition. Stochastic processing dominates current word-construction/recognition and grammar [3].

Like template matching, stochastic processing requires the creation and storage of models of each of the items that will be recognized. At that point the two approaches diverge. Stochastic processing involves no direct matching between stored models and input. Instead, it is based upon complex statistical and probabilistic analyses which are best understood by examining the network-like structure in which those statistics are stored: the *hidden Markov model (HMM)* [3].

In this thesis, modeling of the templates is performed using HMM. Details of HMM are given in Appendix A. Application of HMM in speech recognition is explained in detail in Chapter V.

II.3. PREVIOUS WORK ON SPEECH RECOGNITION FOR TURKISH

There are a few noteworthy studies on the problem of speech recognition for specifically Turkish language.

In [4], a triphone-based, large vocabulary speech recognition system was introduced. In this study, 3-state HMM's were used to model triphones. Furthermore, a new dictionary model based on trie structure was also introduced for Turkish with a new search strategy.

In [5], a rule based speech recognition system for Turkish was developed. The smallest recognition unit of this work is phoneme. A special syllable database was used to control and correct the transformation of speech to text. In this study, several

methods for feature extraction were also compared. Moreover, several approaches to phoneme segmentation can be found in the work.

In [19], a phoneme based speech recognition system for Turkish was developed using Artificial Neural Networks (ANN). This system was also speaker dependent and isolated word. The ANN is used as the pattern classifier and recognizer. The input of the ANN is the FFT of speech signals; that is, simple FFT was used as the feature extraction method in the study.

CHAPTER III

BASIC ACOUSTICS AND SPEECH SIGNAL

As relevant background to the field of speech recognition, this chapter intends to discuss how the speech signal is produced and perceived by human beings. This is an essential subject that has to be considered before one can pursue and decide which approach to use for speech recognition.

III.1. THE SPEECH SIGNAL

Human communication is to be seen as a comprehensive diagram of the process from speech production to speech perception between the talker and listener, see Figure III.1 [10].

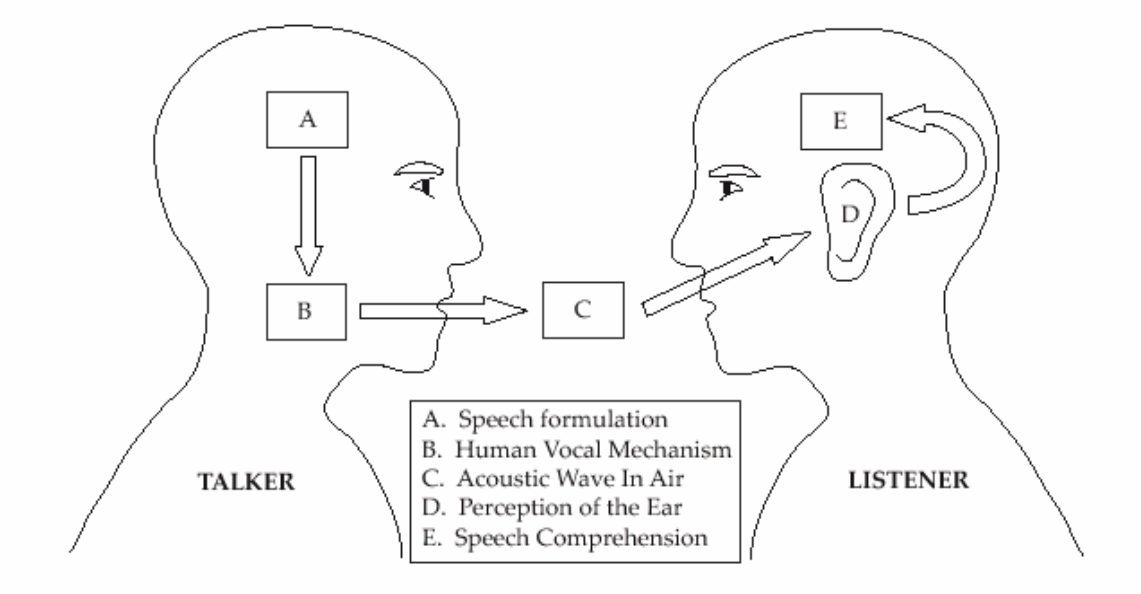


Figure III.1. Schematic Diagram of the Speech Production/Perception Process [10].

Five different elements, A.Speech formulation, B.Human vocal mechanism, C.Acoustic air, D.Perception of the ear, E.Speech comprehension, will be examined more carefully in the following sections [10].

The first element (A.Speech formulation) is associated with the formulation of the speech signal in the talker's mind. This formulation is used by the human vocal mechanism (B.Human vocal mechanism) to produce the actual speech waveform. The waveform is transferred via the air (C.Acoustic air) to the listener. During this transfer the acoustic wave can be affected by external sources, for example noise, resulting in a more complex waveform. When the wave reaches the listener's hearing system (the ears) the listener percepts the waveform (D.Perception of the ear) and the listener's mind (E.Speech comprehension) starts processing this waveform to comprehend its content so the listener understands what the talker is trying to tell him or her [10].

One issue with speech recognition is to "simulate" how the listener process the speech produced by the talker. There are several actions taking place in the listeners head and hearing system during the process of speech signals. The perception process can be seen as the inverse of the speech production process [10].

III.2. SPEECH PRODUCTION

To be able to understand how the production of speech is performed one need to know how the human's vocal mechanism is constructed, see Figure III.2 [1].

The most important parts of the human vocal mechanism are the vocal tract together with nasal cavity, which begins at the velum. The velum is a trapdoor-like mechanism that is used to formulate nasal sounds when needed. When the velum is lowered, the nasal cavity is coupled together with the vocal tract to formulate the desired speech signal. The cross-sectional area of the vocal tract is limited by the tongue, lips, jaw and velum and varies from 0-20 cm^2 [1].

When humans produce speech, air is expelled from the lungs through the trachea. The air flowing from the lungs causes the vocal cords to vibrate and by forming the vocal tract, lips, tongue, jaw and maybe using the nasal cavity, different sounds can be produced.

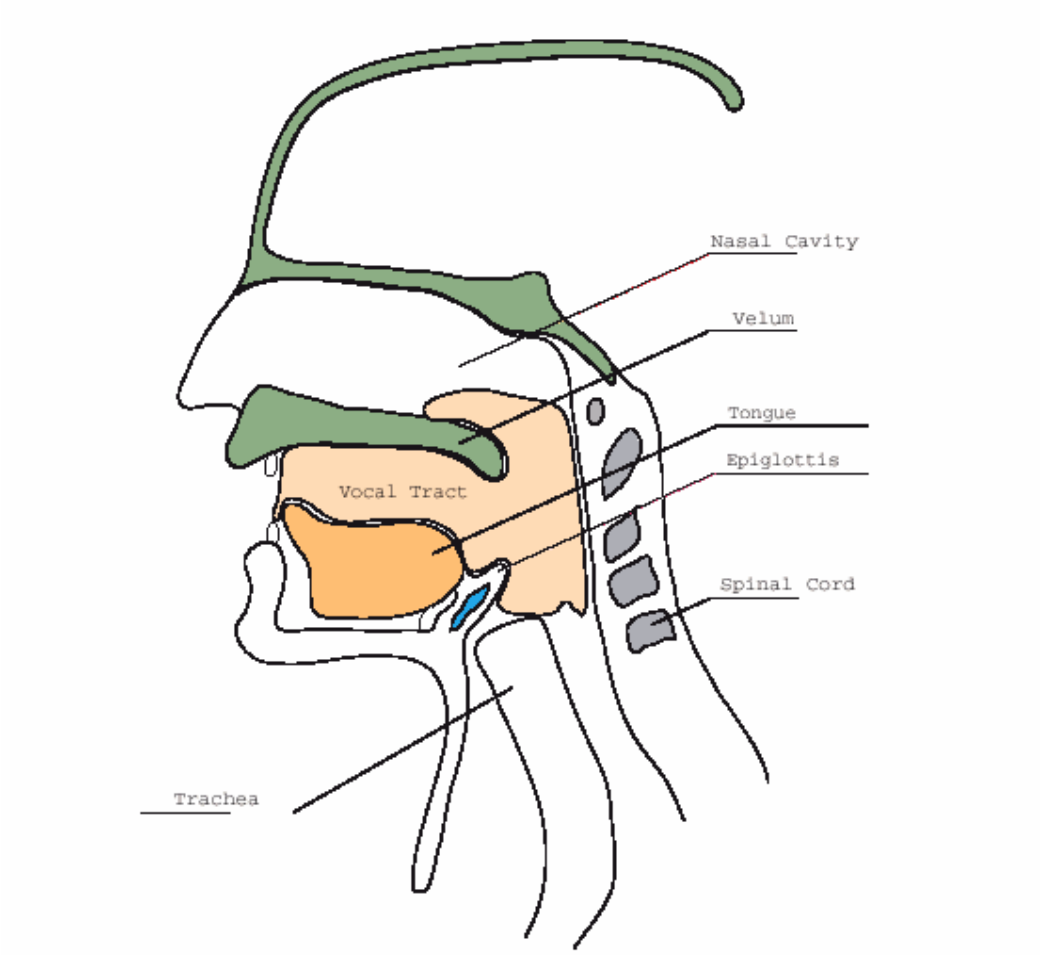


Figure III.2. Human Vocal Mechanism [1].

III.3. SPEECH REPRESENTATION

The speech signal and all its characteristics can be represented in two different domains, *the time* and *the frequency domain*.

A speech signal is a slowly time varying signal in the sense that, when examined over a short period of time (between 5 and 100 ms), its characteristics are short-time stationary. This is not the case if we look at a speech signal under a longer time perspective (approximately time T >0.5 s). In this case the signals characteristics are non-stationary, meaning that it changes to reflect the different sounds spoken by the talker.

To be able to use a speech signal and interpret its characteristics in a proper manner some kind of representation of the speech signal are preferred. The speech representation can exist in either the time or frequency domain, and in three different ways [1]. These are a *three-state representation*, a *spectral representation* and the last

representation is a *parameterization of the spectral activity* .These representations will be discussed in the following sections.

III.3.1. Three-state Representation

The three-state representation is one way to classify events in speech. The events of interest for the three-state representation are:

- Silence (S) - No speech is produced.
- Unvoiced (U) - Vocal cords are not vibrating, resulting in an aperiodic or random speech waveform.
- Voiced (V) - Vocal cords are tensed and vibrating periodically, resulting in a speech waveform that is quasi-periodic.

Quasi-periodic means that the speech waveform can be seen as periodic over a short-time period (5-100 ms) during which it is stationary.

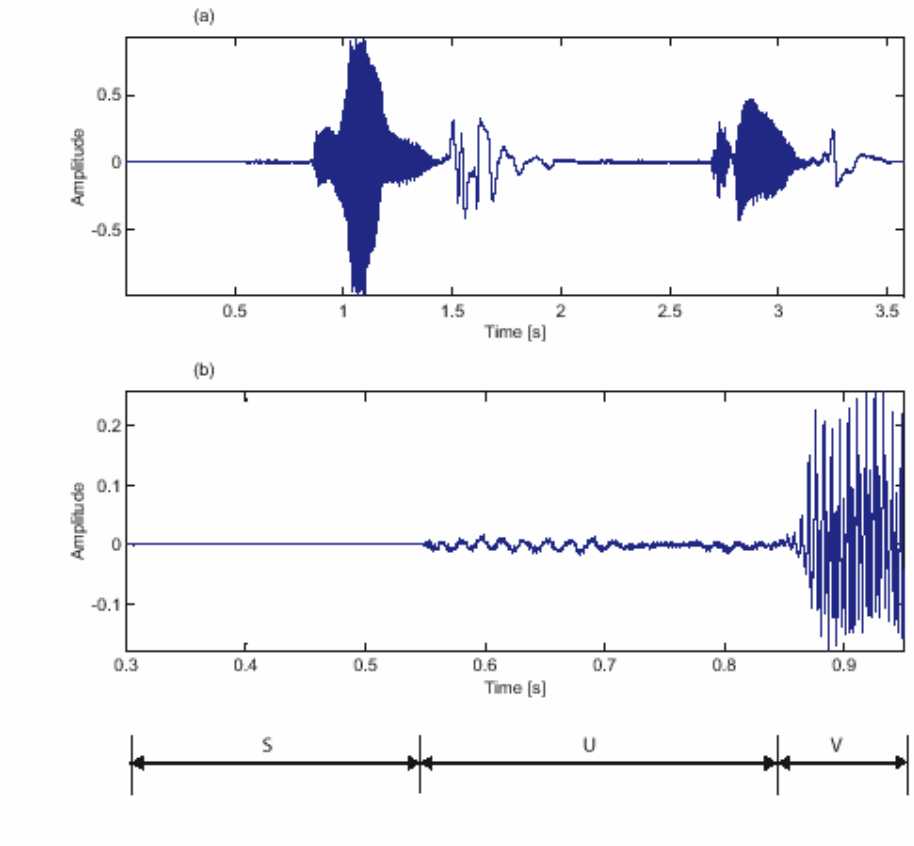


Figure III.3. Three-state Representation [10].

The upper plot (a) contains the whole speech sequence and in the middle plot (b) a part of the upper plot (a) is reproduced by zooming an area of the whole speech

sequence. At the bottom of Figure III.3 the segmentation into a three-state representation, in relation to the different parts of the middle plot, is given.

The segmentation of the speech waveform into well-defined states is not straightforward. But this difficulty is not as a big problem as one can think. However, in speech recognition applications the boundaries between different states are not exactly defined and therefore non-crucial.

As complementary information to this type of representation it might be relevant to mention that these three states can be combined. These combinations result in three other types of excitation signals: *mixed*, *plosive* and *whisper* [10].

III.3.2. Spectral Representation

Spectral representation of speech intensity over time is very popular, and the most popular one is the *sound spectrogram*, see Figure III.4.

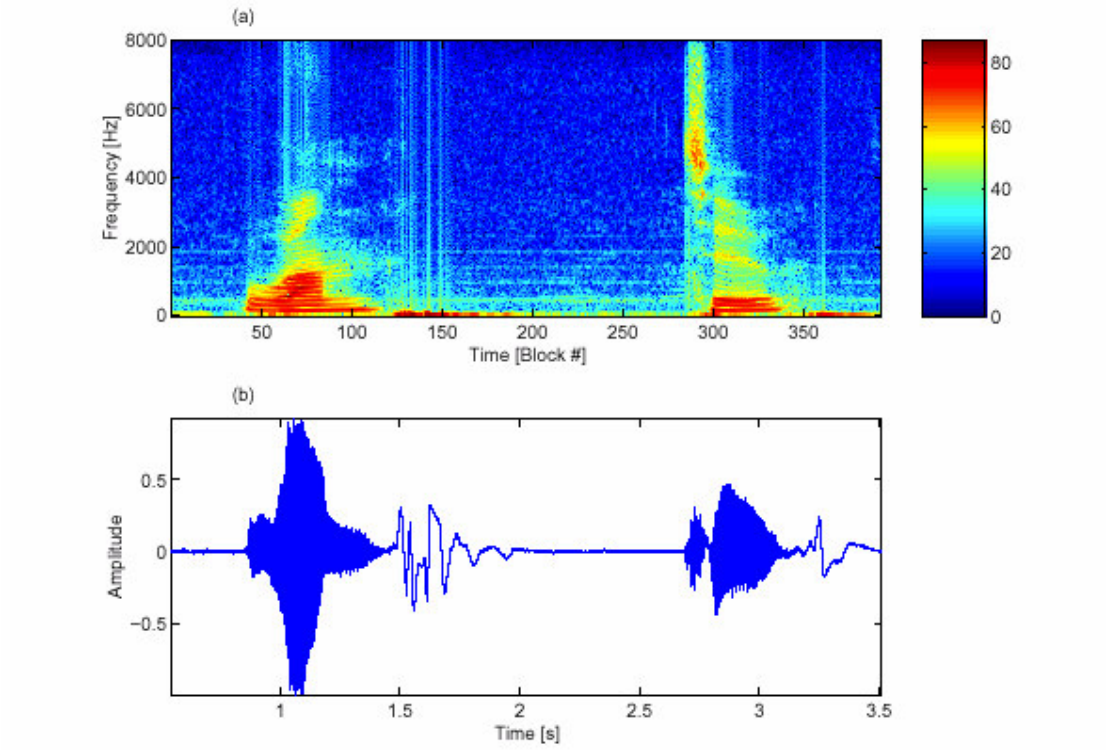


Figure III.4. Spectrogram Using Welch's Method (a) and Speech Amplitude (b) [10].

Here the darkest (dark blue) parts represent the parts of the speech waveform where no speech is produced and the lighter (red) parts represent intensity if speech is produced.

Figure III.4a shows a spectrogram in the frequency domain and in Figure III.4b the speech waveform is given in the time domain. For the spectrogram Welch's method is used, which uses averaging modified periodograms. Parameters used in this method are block size K=320, window type Hamming with 62.5% overlap resulting in blocks of 20 ms with a distance of 6.25 ms between blocks [10].

III.3.3. Parameterization of the Spectral Activity

When speech is produced in the sense of a time-varying signal, its characteristics can be represented via a parameterization of the spectral activity. This representation is based on the model of speech production.

The human vocal tract can (roughly) be described as a tube excited by air either at the end or at a point along the tube. From acoustic theory it is known that the transfer function of the energy from the excitation source to the output can be described in terms of natural frequencies or resonances of the tube, more known as *formants*. Formants represent the frequencies that pass the most acoustic energy from the source to the output. This representation is highly efficient, but is more of theoretical than practical interest. This is because it is difficult to estimate the formant frequencies in low-level speech reliably and defining the formants for unvoiced (U) and silent (S) regions [10].

III.4. PHONEMICS AND PHONETICS

As discussed earlier in this chapter, the speech production begins in the human's mind, when he or she forms a thought that is to be produced and transferred to the listener. After having formed the desired thought, he or she constructs a phrase/sentence by choosing a collection of finite mutually exclusive sounds. The basic theoretical unit for describing how to bring linguistic meaning to the formed speech, in the mind, is called *phonemes*.

Phonemes can be seen as a way of how to represent the different parts in a speech waveform, produced via the human vocal mechanism and divided into continuant (stationary) or non-continuant parts, see Figure III.5.

A phoneme is continuant if the speech sound is produced when the vocal tract is in a steady-state. In opposite of this state, the phoneme is non-continuant when the vocal tract changes its characteristics during the production of speech. For example if the area in the vocal tract changes by opening and closing the mouth or moving your

tongue in different states, the phoneme describing the speech produced is non-continuant.

Phonemes can be grouped based on the properties of either the time waveform or frequency characteristics and classified in different sounds produced by the human vocal tract [10].

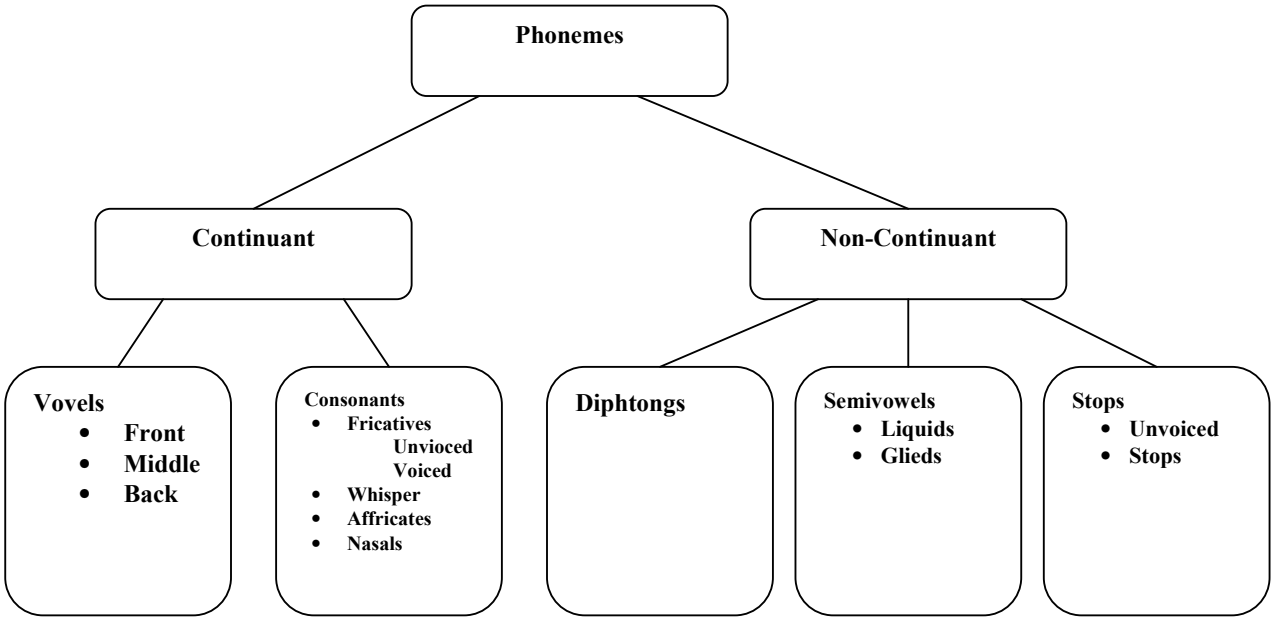


Figure III.5. Phoneme Classification [10].

CHAPTER IV

SPEECH PROCESSING

Once the speech signal is captured and retained in the form of digital data, it needs to be processed prior to being used for the speech recognition process. This chapter is devoted to speech processing techniques used in speech recognition systems. After a short introduction to Digital Signal Processing (DSP), techniques for feature extraction step, and the Mel Frequency Cepstral Coefficients (MFCC) are defined.

IV.1. DIGITAL SIGNAL PROCESSING (DSP)

Digital Signal Processing is one of the most powerful technologies that will shape science and engineering in the twenty-first century. Revolutionary changes have already been made in a broad range of fields: communications, medical imaging, radar & sonar, high fidelity music reproduction, and oil prospecting, to name just a few. Each of these areas has developed a deep DSP technology, with its own algorithms, mathematics, and specialized techniques [15].

Digital Signal Processing is distinguished from other areas in computer science by the unique type of data it uses: signals. In most cases, these signals originate as sensory data from the real world: seismic vibrations, visual images, sound waves, etc. DSP is the mathematics, the algorithms, and the techniques used to manipulate these signals after they have been converted into a digital form. This includes a wide variety of goals, such as: enhancement of visual images, recognition and generation of speech, compression of data for storage and transmission, etc.

IV.1.1. Audio Processing

The two principal human senses are vision and hearing. Correspondingly, much of DSP is related to image and audio processing. People listen to both music and speech. DSP has made revolutionary changes in both these areas.

IV.1.2. Speech Production

The human vocal tract is an acoustic cavity with resonant frequencies determined by the size and shape of the chambers. Sound originates in the vocal tract in one of two basic ways, called voiced and fricative sounds. With voiced sounds, vocal cord vibration produces near periodic pulses of air into the vocal cavities. In comparison, fricative sounds originate from the noisy air turbulence at narrow constrictions, such as the teeth and lips.

IV.1.3. Speech Recognition

The automated recognition of human speech is immensely more difficult than speech generation. Speech recognition is a classic example of things that the human brain does well, but digital computers do poorly. Digital computers can store and recall vast amounts of data, perform mathematical calculations at blazing speeds, and do repetitive tasks without becoming bored or inefficient. Unfortunately, present day computers perform very poorly when faced with raw sensory data. Teaching a computer to send you a monthly electric bill is easy. Teaching the same computer to understand your voice is a major undertaking.

IV.2. FEATURE EXTRACTION

Obtaining the acoustic characteristics of the speech signal is referred to as Feature Extraction. Feature Extraction is used in both training and recognition phases. It comprise of the following steps:

1. Frame Blocking
2. Windowing
3. FFT (Fast Fourier Transform)
4. Mel-Frequency Wrapping
5. Cepstrum (Mel Frequency Cepstral Coefficients)

} Feature Extraction

This stage is often referred as *speech processing front end.* The main goal of Feature Extraction is to simplify recognition by summarizing the vast amount of speech data without losing the acoustic properties that defines the speech [12]. The schematic diagram of the steps is depicted in Figure IV.1.

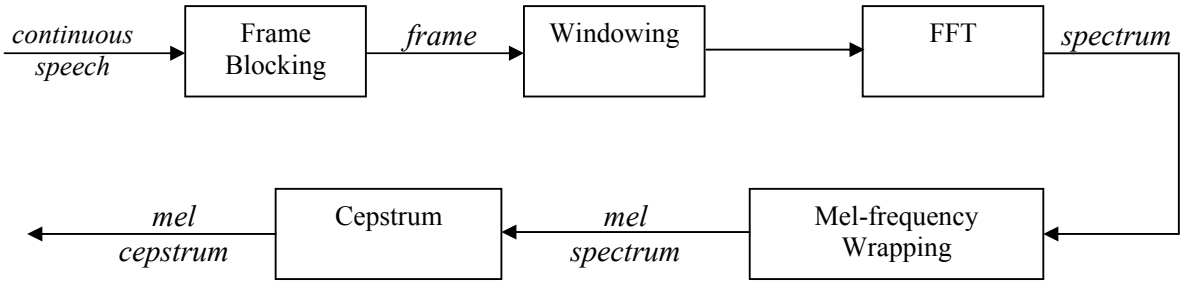


Figure IV.1. Feature Extraction Steps.

IV.2.1. Frame Blocking

Investigations show that speech signal characteristics stays stationary in a sufficiently short period of time interval (It is called *quasi-stationary).* For this reason, speech signals are processed in short time intervals. It is divided into frames with sizes generally between 30 and 100 milliseconds. Each frame overlaps its previous frame by a predefined size. The goal of the overlapping scheme is to smooth the transition from frame to frame [12].

IV.2.2. Windowing

The second step is to window all frames. This is done in order to eliminate discontinuities at the edges of the frames. If the windowing function is defined as w(n), $0 < n < N-1$ where N is the number of samples in each frame, the resulting signal will be; $y(n) = x(n)w(n)$. Generally *hamming* windows are used [12].

IV.2.3. Fast Fourier Transform (FFT)

The next step is to take Fast Fourier Transform of each frame. This transformation is a fast way of Discrete Fourier Transform and it changes the domain from time to frequency [12].

IV.2.4. Mel-frequency Wrapping

The human ear perceives the frequencies non-linearly. Researches show that the scaling is linear up to 1 kHz and logarithmic above that. The Mel-Scale (Melody Scale) filter bank which characterizes the human ear perceiveness of frequency is as shown in Figure IV.2. It is used as a band pass filtering for this stage of identification. The signals for each frame is passed through Mel-Scale band pass filter to mimic the human ear [17][12][18].

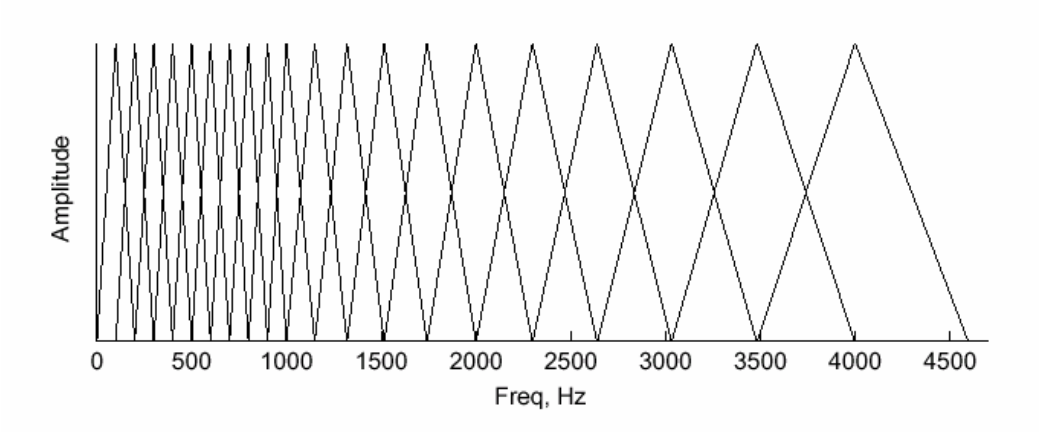


Figure IV.2. Mel-Scaled Filter Bank [18].

IV.2.5. Cepstral Coefficients

As of the final step, each frame is inverse Fourier transformed to take them back to the time domain. Instead of using inverse FFT, Discrete Cosine Transform is used as it is more appropriate [17][18]. The discrete form for a signal *x(n)* is defined as

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\pi(2n-1)(k-1)}{2N} \qquad w(k) = \begin{cases} \sqrt{1/N}, & k = 1 \\ \sqrt{2/N}, & 2 \le k \le N \end{cases}$$

x: original signal,

y: Resulting Discrete Cosine Transformed signal,

N: number of samples.

As a result of this process, Mel-Frequency Cepstral Coefficients (22 coefficients) are obtained. These coefficients are called feature vectors. They are also called observation vectors in the speech recognition terminology.

CHAPTER V

LANGUAGE MODEL FOR TURKISH

V.1. MORPHOLOGY OF TURKISH LANGUAGE

This chapter examines the Turkish language from the point of view of speech recognition problem.

Turkish has an agglutinative morphology with productive inflectional and derivational suffixations. Since it is possible to produce new words with suffixes, the number of words is very high. According to [11], the number of distinct words in a corpus of 10 million words is greater than 400 thousand. Such sparseness increases the number of parameters to be estimated for a word based language model.

In agglutinative languages, it is possible to add morphemes after another one. Each morpheme conveys some morphological information such as tense, case, agreement etc. This property of agglutinative languages results in a large number of words which have the same stem but different endings.

Since there are so many words in the dictionary, if we use a large vocabulary speech recognition system for Turkish language, there will be a large number of out-of-vocabulary (OOV) words which are not modeled.

An example for word production in Turkish;

OSMANLILAŞTIRAMAYABİLECEKLERİMİZDENMİŞSİNİZCESİNE

This word can be broken down into morphemes as follows:

OSMAN + LI + LAŞ + TIR + AMA + YABİL + ECEK + LER + İMİZ + DEN + MİŞ + SİNİZ + CESİNE

The meaning of this word is “as if you were of those whom we might consider not converting into an Ottoman".

In Turkish Language, each letter corresponds to a particular phoneme:

- Vowels..... : a e ı i o ö u ü
- Consonants: b c ç d f g ğ h j k l m n p r s ş t v y z

Language modeling for agglutinative languages needs to be different from modeling languages like English. Such languages also have inflections but not to the same degree as an agglutinative language [13].

V.2. PHONEME-BASED RECOGNITION

Many HMM-based small vocabulary or even medium vocabulary speech recognition systems assign fixed model topology to each of the words. That is, each HMM model in the system has the same number of states. In such systems, the smallest unit for recognition is the words themselves. It is impossible to use any part of the trained model of a word for the training or the recognition of different words. Such systems recognize only the words that are trained. Fixed topology word models are not reasonable for large vocabulary speech recognition systems.

Many modern large vocabulary speech recognition systems use phonemes as the smallest unit for speech recognition. In such strategies, the model of a word is obtained by cascading the models of the phonemes which make up the word. In this thesis, a similar strategy is followed.

Every phoneme is modeled with a 2-state HMM $\lambda = (A, B, \pi)$. This model has two emitting states and simple left-to-right topology as illustrated in Figure V.1. As can be seen from the figure, the HMM model has the property that, as time increases, the state index either increases by one or stays the same. This fundamental property of the topology is expressed mathematically as follows:

$$a_{ij} = 0, \text{ where } \begin{cases} j < i, \quad i,j \le N \\ \\ j > i + 1, \quad i,j \le N \end{cases} \qquad \text{(V.1)}$$

The model states proceed from left to right. This topology is convenient to model the speech signal whose properties change over time in a successive manner.

The model of a word is obtained by cascading the models of the phonemes which make up the word. Figure V.2 illustrates the process of constructing the HMM model for the word “sev”.

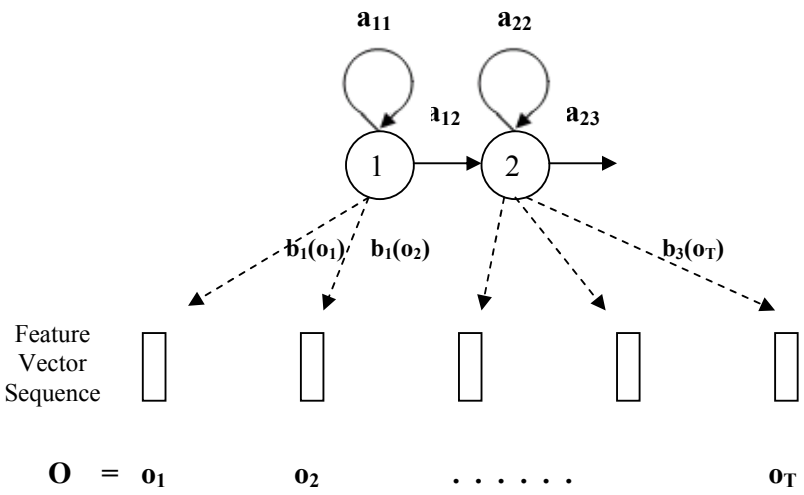


Figure V.1. Example HMM Model for a Phoneme.

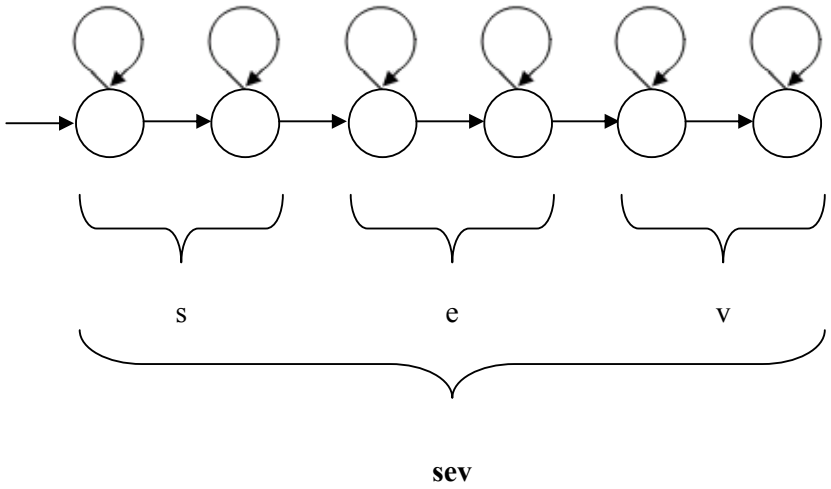


Figure V.2. Construction of HMM Model for the Word "sev" by Cascading Models of Separate Phonemes "s", "e", "v".

CHAPTER VI

IMPLEMENTATION OF THE SYSTEM

VI.1. IMPLEMENTATION

Two important software programs were used during the development of the recognition system for Turkish language. For the data collection stage which mainly consists of voice recording and voice signal editing, Sound Forge (version 4.5) audio editing software of Sonic Foundry was used. This program has extensive audio editing and spectrum analysis capabilities which made the experimental training stage easier. The second program was MATLAB (version 6.1) of MathWorks, where the system was actually developed and tested. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Along with MATLAB, two MATLAB toolboxes, provided under GNU General Public License, were used.

1. VOICEBOX is a toolbox that includes many useful functions for speech processing. Specifically for MFCC calculations, MELCEPST function of this library was used [16].
2. H2M is MATLAB code library which contains several functions for implementing several types of HMM [14].

Implementation of the system includes two distinct stages, namely, a training stage and a recognition (testing) stage.

VI.2. TRAINING STAGE

Since the system is speaker dependent, a training session must be completed for every user of the system. For each user, an enrollment process has to be performed. During this process, roughly, voice recordings of the words user says are transformed into HMM models of the phonemes of these words. General architecture of the training stage is given in Figure VI.1. This section explains the whole process in detail.

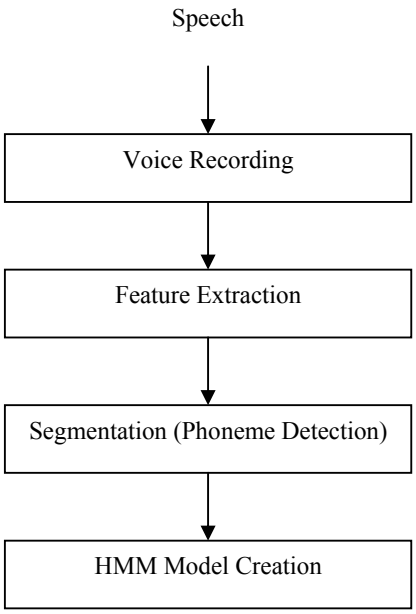


Figure VI.1. General Architecture of Training Stage.

VI.2.1. Voice Recording

Every user has to complete a voice recording session in which a number of words are recorded. These words are selected so that every phoneme has sufficient number of HMM models to be used in recognition. The list of those words is given in Appendix B.

Every word is recorded in 22050 Hz., 16 bit, mono Wave file format with the same name as the word in lowercase letters. However, special Turkish letters are represented their uppercase Latin letter counterpart as in Table VI.1.

Turkish Letter	Representation
Ç	C
Ğ	G
I	I
İ	i
Ö	O
Ş	S
Ü	U

Table VI.1. Representation of Special Turkish Letters.

VI.2.2. Feature Extraction

A key assumption made in the design of most speech recognition systems is that the segment of a speech signal can be considered as stationary over an interval of few milliseconds. Therefore the speech signal can be divided into blocks which are usually called frames. The spacing between the beginnings of two consecutive frames is in the order of 10 msecs, and the size of a frame is about 25 msecs. That is, the frames are overlapping to provide longer analysis windows. Within each of these frames, some feature parameters characterizing the speech signal are extracted. These feature parameters are then used in the training and also in the recognition stage [4].

Before creating the HMM models of the phonemes of the words recorded, the recording is applied a feature extraction process. In this thesis, MFCC is chosen as the feature extraction method.

MFCC is applied with frame length of 128 which is equal to 5.8 msecs, and with frame overlap of length 32 which is equal to 4.25 msecs (1 second of recording is equal to 22050 data points). These values are found to result in most accurate recognition after several experiments. The output of MFCC is a 22 dimensional matrix where the number of rows is the number of frames of the speech signal. The dimension is fixed 22 which represent the length of the vocal tract of humans. The function MFCC is capable of providing first and second order derivatives of the feature vector, however, it was observed that using those derivatives has no positive effect on the recognition accuracy while putting a computation overhead on the recognition.

VI.2.3. Segmentation (Phoneme Detection)

In order to model phonemes of a spoken word, the word must be split into its phonemes. However, there is not a straightforward algorithm for phoneme detection

and segmentation, and a common practice is to manually split the training words into phoneme segments. However, it is not a useful method if there are many training words. Thus, a spectral method was devised which applied HMM Viterbi search on the trained word itself to determine the best path through the phonemes of the word. Actually this method failed for many Turkish words, resulting in bad modeling and finally bad detection scores. Then, a different method was devised, in which every phoneme was given a weight according to the type of the phoneme (e.g. vowels are told longer in duration than consonants), the location of the phoneme in the word etc. It was observed that this method performs very well. Details of the phoneme weighting algorithm are as follows:

```
if vowel then
      weight = 9
if 's' then
      if first_phoneme then
            weight = 6.5
      else
            weight = 8
if 'C'  (actually Ç) then
      weight = 9
if 'c' then
      weight = 7
if 'S'  (actually Ş) then
      weight = 9
if 'p' then
      weight = 9
if 'y' then
      weight = 4.5
if 't' or 'k' then
      if last_phoneme then
            weight = 9
      else
            weight = 4
if 'm' or 'n'
      if last_phoneme then
            weight = 8
      else
            weight = 4
else
      if first_phoneme then
            weight = 3.5
      else
            weight = 4
end
```

Figure VI.2. Segmentation Algorithm.

As seen in Figure VI.2, every vowel is given the highest weight because every vowel is uttered the longest period of time within the word, and every vowel is uttered almost the same period of time in average. Thus, if the letter of the word is

vowel, then it is given 9 points. Similarly, consonants are uttered almost one third of vowels in average with a few exceptions. Those exceptions include the letters Ç, C, Ş, S, T, K, P, Y, M, and N, a few of which are of different length if they are at the beginning or at the end of word due to the stress on the part of the word. Figure VI.3 illustrates sample segmentation for word "merkez" using the proposed algorithm.

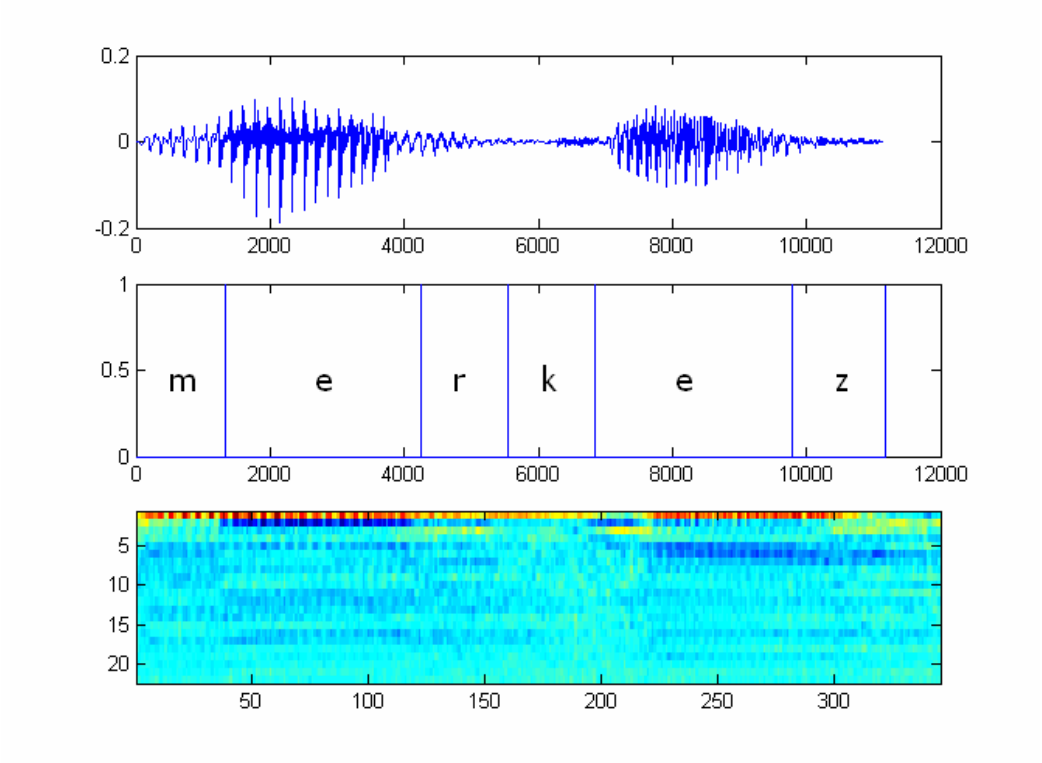


Figure VI.3. Segmentation of Word "merkez".

In the figure above, topmost part is the speech signal of the word "merkez" which lasts about 0.5 seconds (it is equal to about 12000 samples as seen in the figure). The bottom part is the cepstrum of the same signal, that is, the output of MFCC. The middle part is the result of the segmentation algorithm proposed.

VI.2.4. HMM Model Creation

After the segmentation, parts of the MFCC matrix that corresponds to each phoneme is used in the construction of HMM models of that phoneme. For example, from the MFCC of the word "merkez", we obtain one model for "m", "r", "k", "z", and two models for "e". While training several words, it is possible to model the same phoneme many times. In this case, each model file of a phoneme is named like

"_e_1", "_e_2", "_e_15", etc. Having more than one model for a phoneme increases the accuracy of recognition while decreasing the response time.

A HMM model file consist of two parameters used in HMM functions, namely, *mu* and *sigma*. *mu* is 2x22 matrix; 2 rows for 2 states of HMM, and 22 columns for 22 dimensions of observations vector. Similarly, *sigma* is 1x22 matrix.

VI.3. RECOGNITION STAGE

In this stage, an unknown utterance is to be identified as a Turkish word using the phoneme models obtained in the enrollment phase. General view of this stage is illustrated in Figure VI.4.

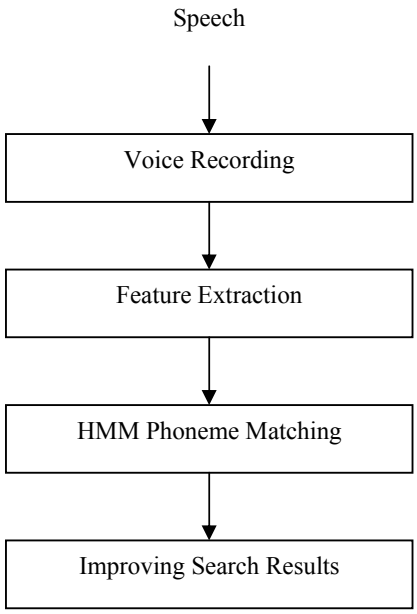


Figure VI.4. General Architecture of Recognition Stage.

The first two steps performed in recognition are the same as of training stage. Then, other steps include a brute force pattern matching operation, and an intelligent improvement on the raw search results of brute force pattern matching.

VI.3.1. HMM Phoneme Matching

Because training is performed at the phoneme level, the recognition has to be performed at the same level. However, we cannot use phoneme segmentation and therefore we cannot make a phoneme-by-phoneme matching because during

recognition the uttered word is unknown whereas it is a known item during training. Therefore, we need to split the unknown utterance into parts of such a length that each part should be as small as possible while retaining the speech data. While performing the training, the length of the smallest of all phonemes is saved in a parameter file. Now, while performing the recognition, we split the unknown utterance into parts smaller than this saved value because we should not miss any phoneme. An optimum minimum split length is found by experiment to be about half the saved value. In chapter VIII, test results for five different split ratios are given.

After the split, for each part of the utterance, HMM Viterbi search is made with the feature vector extracted from that part. The search is made against every phoneme model and the model with the highest score is saved and this process continues until every part of the unknown utterance is tested. After this search ends, we come up with a result like "mmmmeeeeeeeeCrrrrkkkkeeeeeeeezzzzz" for the word "merkez" which was tested as the unknown utterance. Here, it is seen that the unknown utterance was split into 35 parts, and those parts are tested and recognized like this. The raw result contains redundancy and also some detection errors. Then, this raw result needs to be refined with an extra step. Table VI.2. shows several sample raw results of the system.

Original Word	**Raw Detection Result**
CIkan	nhlCCCIukIIIIIIIkkkkaaaaaannnnnnnnnn
CalISkan	CCCCraaaaaIaaalllllIIIIIISSSSSSSSakkkaaaaaaaannnnnnnn
Onemli	aOOOOOnnnnneeeeeeemmmmmlllliiiiiiiik
ayrIca	aaaaaaayyyyrrrrIIIIIIIImccccScraaaaaaaaah
iCinde	eiiiyiikCCCCCCCiiiiiiinnnnnddddeeeeeee
genellikle	gggeeeeeeeenneeeeeeelllllliiiiiiikkkkkllleeeeeeee
olumlu	Booooooolllluuuuuuuuummmmlllluuuuuuuud
tarihinde	tttaaaaaaaarrrriiieyiihhhhiinnniinnnnddeeeeeeeeek
zaman	Zzzzaaaaaaaaaammmmnuaaaaaaaannnnnnnnnp

Table VI.2. Several Raw Detection Results of the System.

VI.3.2. Improving Search Results

The last step of the recognition phase is to make improvements on the raw result so that more useful result is obtained. Raw results contain several errors and redundancies so two simple refinement algorithms was devised and used:

1. Removing single letters: Some parts of the unknown utterance can be detected to be more similar to other phonemes than the correct phoneme. Thus, the raw result can contain wrong phonemes. For example, the raw result "aaaaaaayyyyrrrrIIIIIIIImccccScraaaaaaaaah" for the word "ayrıca" contains wrong phonemes like "m", "S", "r", and "h". We can easily and reliably make the decision that those phonemes are falsely detected and can be removed because they appear within the result only once in a sequence. That is, we expect a phoneme to belong to the detected word, it should repeat one after another at least twice. Thus, internal refinement is then: "aaaaaaayyyyrrrrIIIIIIIIcccccaaaaaaaaa".
2. Removing repeated letters: For a correct detection, we expect a phoneme to repeat one after another at least twice. This means, we regard repeating distinct phonemes as single detected phoneme. The second refinement removes the redundancy of phonemes and results in the cleanest result like "ayrıca" for the original word "ayrıca".

VI.3.3. Handling Repeating Letters

There is one more case that needs to be handled specially: repeating letters in the words like "s**aa**t", "gene**ll**ikle". The solution is easy and straightforward. For example, we expect the middle letter "l" of the word "genellikle" to repeat very few times when compared to the letter "e". However, in the internal result of the recognition "gggeeeeeeeeenneeeeeeee**lllllll**iiiiiiikkkkkllleeeeeeee", there are 7 "l"s. This number is about 2-fold of the average number of other consonants in the word. In addition, this number is about the same as the average number of vowels in the word. Therefore, we have a strong and reliable hint that the letter "l" should repeat twice, and the final result should be "gene**ll**ikle" instead of "gene**l**ikle". In Turkish, there is no such possibility that a letter repeats more than 2, so we can simply make the decision that a letter should appear once or twice.

CHAPTER VII

SAMPLE TRAINING AND RECOGNITION SESSION WITH SCREENSHOTS

VII.1. MAIN MENU

In this chapter, a sample training and recognition session is given from the speech recognition application with screenshots. The application is developed in both English and Turkish. In Figure VII.1 main menu of the application is seen. Then, other parts of the application are selected through this menu.

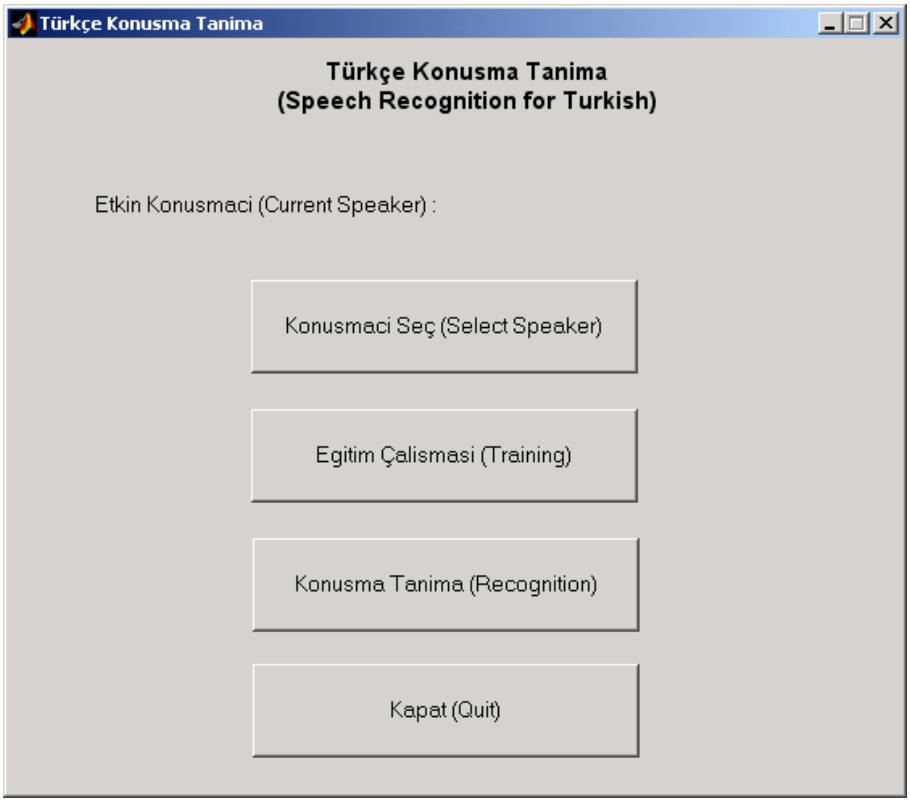


Figure VII.1. Main Menu of the Speech Recognition Application.

When the application begins, there is no user (speaker) currently selected. In order for training and recognition, a user must be selected or created if it does not exist via Select Speaker button. If there is a folder with the same name as the speaker name entered exits, then that user is currently ready to use the system. She can continue with more training or she can continue with recognition. If the folder does not exist, then a new folder with that name is created, and the user should start with a training session.

VII.2. TRAINING

The graphical user interface of the training part of the application is shown in Figure VII.2.

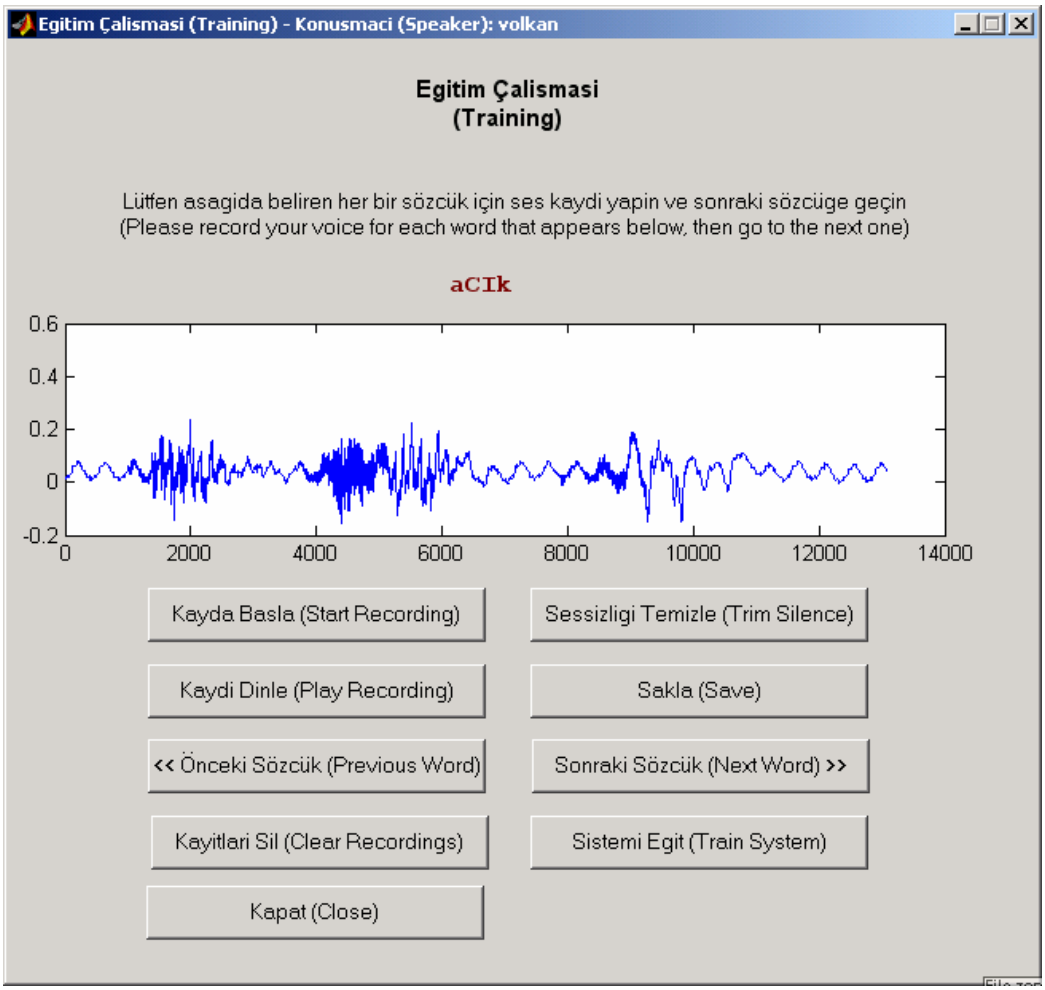


Figure VII.2. GUI of Training Screen.

In the training screen, the user is instructed to record her voice by speaking the words appearing at the top of the screen in red. If the recording is acceptable, she can

skip the next word after saving the current one. If the recording contains blanks at both ends of the speech, which are silence actually, she can trim those parts by using "Trim Silence" button. After all the words are recorded, the training must be completed with the creation of phoneme HMM models. The user can do this by pressing "Train System" button. This process may take some time. After this, the system is ready for recognition.

VII.3. RECOGNITION

The graphical user interface of the recognition part of the application is shown in Figure VII.3.

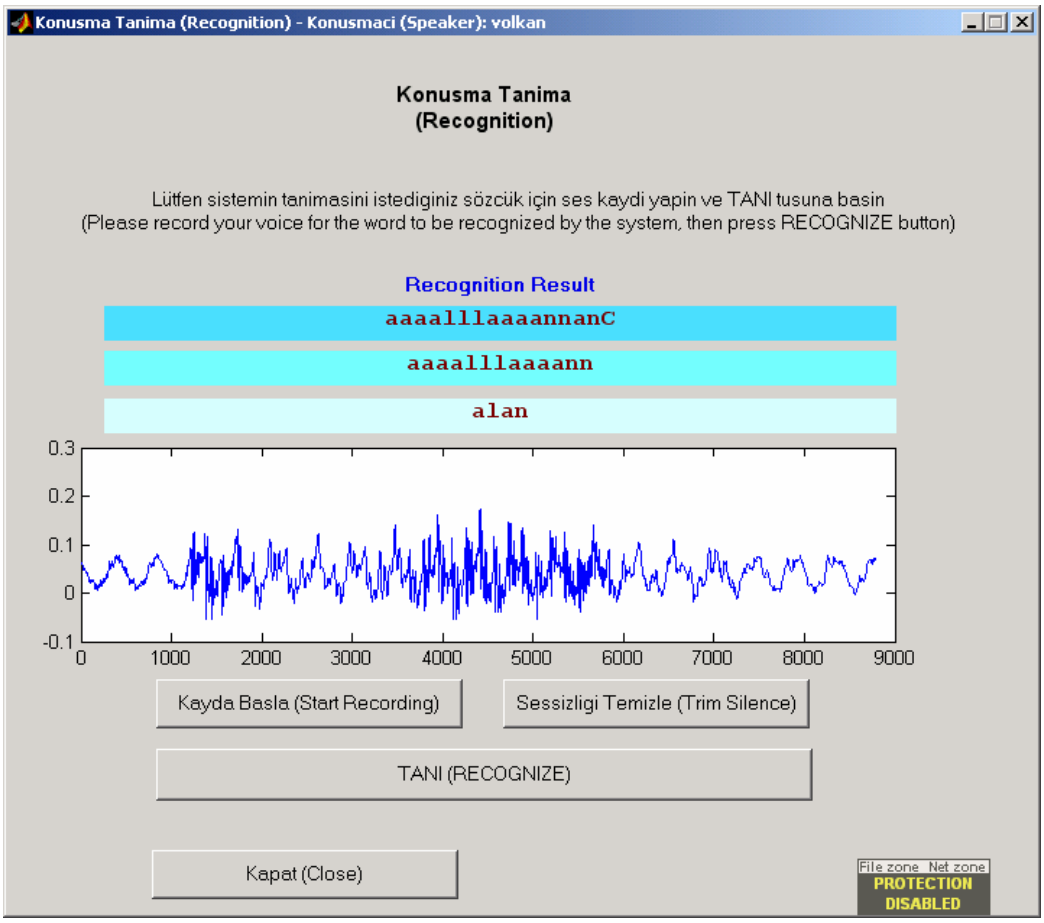


Figure VII.3. GUI of Recognition Screen.

In the recognition screen, the user records her speech to be recognized, and uses the "Trim Silence" button if necessary. Then, she uses "RECOGNIZE" button and waits for the application's response. The raw result appears on the top blue line. The first refined result appears below it. Then the final result appears on the bottom light blue line.

CHAPTER VIII

RESULTS

The system was tested on a PC with P4 2800 MHz CPU, 512 MB RAM, and Windows XP OS. For the testing of the system, 100 words were trained, after which total of 591 HMM models were created. The training session of 100 words was completed in just 56.34 seconds. Training words are listed in Appendix B.

For recognition, a set of 50 words of different length were used. These words are listed in Appendix C.

Five tests were performed for different values of the split ratio of the unknown utterance: 1.2, 1.5, 1.8, 2.1, and 2.5. Detailed test results including response times are given in Appendix D. Test results for split ratio 1.8 is intentionally given in this section in Table VIII.1. With this ratio, the most acceptable recognition results are achieved. Furthermore, a comparison of the five split ratios is given in Table VIII.2.

It is observed that the length of the part of unknown utterance being tested against templates must be smaller than the minimum phoneme length calculated during the training session. With large parts, it is possible to miss some phonemes. Therefore, it is necessary to use smaller parts. On the other hand, with too small parts, it is inevitable to recognize phonemes incorrectly due to lack of speech data. Increasing the split ratio value increases the details of recognition; that is, more phonemes are detected for the same word. Increasing the details, however, does not necessarily causes an increase in the recognition accuracy. The test results show that the optimum length of the portions tested should be the minimum phoneme length calculated during the training session divided by split ratio of 1.8.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaaOCrCSSccsssssssaannInnIIndddaaaaannnnnnnn	aScsanIdan	44.4
2	akISkan	kaaaanapkkkIekkSccCSSSkkkkaaaaannannnnnll	akcSkanl	42.6
3	alISkanlIk	kaaaaoIllIIIccISSSkkkSaaaamuunullOIIuInbkkkkk	alIcSkaulIk	46.6
4	amerika	aaaaanunmmeieiIeeiiiiyykkkaaaaaak	ameiyka	34.0
5	aranjman	haaaaaaaaaannnmanibjsypummaaannnnnnnml	anman	40.0
6	aydInlIk	aaaaaiiiiiikIIndnauunnnuaUUinbkkkkkkk	aiIunUk	38.4
7	barIS	bbaaaaarreeIIeaSjjSjSSSS	bareIjS	24.9
8	baSbakan	bbbbaaaaaSejjjSStgaaaOaakkkOaannkm	bajSakan	35.4
9	bilgi	bbkieeiiiiyypgiiieiiiiiny	beiyi	25.8
10	bizimle	bbIeiisisszzieieeeeImmilleeeeeeke	biszemle	33.9
11	CarSamba	iSScaaaaaaCjSSSSaannanuaaaaabbbaaaka	SaSanaba	37.3
12	cesaret	CCciddeesssssssaadadaarraIaeeeeeettnssttt	Cdesaretst	41.4
13	Cetin	CCCSeiiiiettienIndnnnnmng	Citn	26.0
14	dakika	paaaeeIkCkceeeiigiyCkkOaaaaaa	aeika	30.3
15	deGerli	ddiiiieeeeeeeeeerrnIllygggiiieeii	dierlgiei	34.3
16	devlet	ddeeleeeidIIiiIleeeeeetntsittt	deIiet	31.0
17	eGitim	nyieeeeeieiiiymynykttiliieannnmmmkb	eitinm	35.8
18	evet	eeeeenbUeeeeeeiekbteeett	et	24.6
19	gelenek	ggceeeeeieeeeeenneeeeeiiiuSckkkir	geneik	34.0
20	genC	yyeeiitnnaanngcCCS	yeinanC	18.7
21	gerCek	bbbkceaedeeaenllcIgymkCCCCCceeeeikkikCSjkkS	belCek	44.4
22	haksIzlIk	hhaaakkkkkkestssssaaazzzsaaaIpIIInydmkkkkk	haksazaIk	42.5
23	imkansIz	kiieigiimmnnkCeeaaannIasssssaaaalaazzszssss	imneansazs	43.6
24	istakoz	giiisssssssttaaaaaakkuvaIaluasssss	istaks	35.2
25	istanbulda	niecsssssstttadaaadaannIIbuolluumuiibaaaaaakbb	stanIluiab	47.9
26	iSyeri	iiiiSSSCCCCSCkikggiiiiieeirreIyeekiit	iSCgierei	39.2
27	izmir	giiiiiiezzzsidunnniiiiidiccrrrrkS	iznicr	35.2
28	jale	CkjjjCreaaaaaanInlyeeeeemt	jae	26.7
29	jOle	ndCcjjjjCCcIOOrrIleeeeeeerei	jCOre	28.7
30	kompUter	kkkfoooluuaamdmpppmrUnmmmeeeeaaaCSiii	kouapmeai	38.3
31	konuSma	kkuooauunnanoSCCCCdpaIbaaaaaaakd	kounCa	32.9
32	meydana	nimmaeeeeeeiipladaannaadIIaIdkk	meianaIk	31.6
33	murat	nnuubuunorraaaaaakmpttttttt	nurat	26.9
34	nedeniyle	nnndeeeeeiIeeeendniiiiiiiiiyyneelleeteb	neiyele	39.9
35	olasIlIk	ooooollllddaaIaszsssostlIlIIIlaIIUUnnkkkkkkkk	olasIUnk	46.3
36	saat	ssssdaaaabkaaaaaIakCttttt	sat	25.5
37	saGlIk	ssssaaaaaaaaaaulIIIIIInIkkkk	saIk	29.1
38	sanayi	sssstaaannnaaaaaaOeayimgiieiei	sanai	31.0
39	Sarj	SSSSkaaaaaaaaaCrjjjjjS	Saj	22.6
40	saz	sssstafaaaaaaaazzzzs	saz	20.8
41	seCim	sssssteeiiingcCCSiiiIimammmnblnmb	seiCim	33.7
42	sessizlik	sssssseeeeeeissssssiieeeszzzsysbieiieyuigiSCkkkhI	sesiezik	50.3
43	sistemi	sssssseiizsssssssttyteeeeeeidiiiiiim	sistei	37.2
44	Soyle	CSSCCSCaOOUUOyeillceeeeeeel	SCOUle	27.6
45	takIm	ttaaIIIikkkkaaIIunnnnnumyby	taIkaIn	27.9
46	tanIma	ttaaaaauundaIIInInnnmmaaaaaaankk	tauInmak	31.8
47	Uzerinde	iOOaiiszzsssyennedddyicieinnnnnnnnunnndeeeeeeeeee	Oizsndne	51.0
48	volkan	bbmnpbooooolluknnkkCaaaaaaanndnp	bolnkan	33.0
49	yakIn	iyyaaUeIUIkkkIICdnmnnnmn	yakIn	24.7
50	yardImcI	lyygayaaaaaaaadntddaandnnaCccCcIIIImn	yadancI	43.3

Table VIII.1. Recognition Results of the System for Split Ratio = 1.8.

Split Ratio		1.2		1.5		1.8		2.1		2.4	
No	Original	Final Result	Time	Final Result	Time	Final Result	Time	Final Result	Time	Final Result	Time
1	aCIsIndan	aCsnan	38.3	aCSsndan	40.7	aScsanIdan	44.4	aCsanIndan	48.7	aScsandan	51.3
2	akISkan	akSkan	36.6	akanb	39.5	akcSkanl	42.6	akIkSCSkanml	46.9	akSCSkan	49.1
3	alISkanlIk	aISak	39.9	aIcSkauIk	43.2	alIcSkaulIk	46.6	alIcSkaunmIk	50.8	alIcSkanulIk	53.4
4	amerika	aeika	29.0	aiya	31.3	ameiyka	34.0	aeiyka	37.0	anieiyka	39.2
5	aranjman	anuan	34.0	anamanm	36.7	anman	40.0	anuanm	43.5	aniyuman	46.0
6	aydInlIk	aink	33.1	aiInk	35.4	aiIunUk	38.4	aiIanank	42.5	aiInanInk	44.5
7	barIS	aIjS	21.5	arIS	22.8	bareIjS	24.9	arIS	27.3	barSjS	28.9
8	baSbakan	bajSaka	29.7	bajSakan	32.5	bajSakan	35.4	baejSakak	38.9	baejSaOak	40.4
9	Bilgi	i	21.8	iem	23.6	beiyi	25.8	bieiygieim	28.0	bieiygim	29.7
10	bizimle	isiele	29.0	biezie	31.3	biszemle	33.9	biszіemle	37.0	bisziemile	39.1
11	CarSamba	aSaba	32.0	aSaba	34.0	SaSanaba	37.3	SajSa	40.6	SyaSaba	42.8
12	cesaret	Csaet	35.5	Cdset	38.1	Cdesaretst	41.4	Cidesaretst	45.2	Ciesaretst	47.8
13	Cetin	Cen	22.3	Cing	23.9	Citn	26.0	CitiIdng	28.8	Cein	29.9
14	Dakika	eka	25.7	aegka	28.1	aeika	30.3	Seigka	33.3	aeSegka	35.0
15	deGerli	iergi	29.4	iegi	31.8	dierlgiei	34.3	ielgi	37.8	diergi	39.4
16	Devlet	et	26.5	eIet	28.3	deIiet	31.0	eIiIetst	34.1	deIlest	35.7
17	eGitim	ieiye	30.3	einm	32.7	eitinm	35.8	ieigtinmb	38.9	eigtinm	40.9
18	Evet	e	20.7	et	22.5	et	24.6	eIetet	27.0	eIet	28.0
19	gelenek	geik	28.4	geik	30.8	geneik	34.0	geieik	37.0	geineik	38.5
20	genC	enC	16.2	eanC	17.2	yeinanC	18.7	yeitnaC	20.7	yetnanC	21.7
21	gerCek	belCeiCk	37.3	eCeiCk	40.3	belCek	44.4	nlCeiCk	48.3	nelgCek	50.7
22	haksIzlIk	aksazaIk	36.2	aksazIk	39.0	haksazaIk	42.5	aksazImk	46.9	haksazaIk	49.1
23	imkansIz	eikasazs	37.2	imasazs	39.5	imneansazs	43.6	ieimkeasazs	47.9	ieimkansazs	50.3
24	istakoz	isas	30.4	isaus	32.0	istaks	35.2	istakalas	38.8	istakulas	40.7
25	istanbulda	istaua	40.0	istaIuak	42.7	stanIluiab	47.9	istaIlulan	51.6	istdabluiya	54.2
26	iSyeri	iSCkiei	33.1	iSCgiek	35.6	iSCgierei	39.2	iSCSkgieieki	42.5	iSCSkgiIre	44.7
27	Izmir	inicrS	29.8	iziniecr	32.1	iznicr	35.2	iszncrS	38.6	iszsnicr	40.2
28	Jale	ae	22.8	jae	24.3	jae	26.7	Cane	29.0	jale	30.5
29	jOle	CIe	24.4	CIre	25.8	jCOre	28.7	njCare	31.3	djCOre	33.0
30	kompUter	koumea	32.3	koumeai	34.7	kouapmeai	38.3	kouampmeaSi	41.8	kouamkmeaSi	43.8
31	konuSma	uCa	28.0	uCa	30.0	kounCa	32.9	uouaCdak	36.1	kounCdma	37.7
32	meydana	eiaI	26.7	eaIk	28.7	meianaIk	31.6	eidandI	34.4	meanaI	36.0
33	Murat	uat	22.8	uat	24.4	nurat	26.9	uakt	29.1	muoat	30.6
34	nedeniyle	neie	34.1	neniye	36.1	neiyele	39.9	mneniye	43.5	mneniyele	46.1
35	olasIlIk	olasIk	39.5	olIsInk	42.3	olasIUnk	46.3	olsIUnk	50.7	olazsIlIUnk	53.4
36	Saat	sat	21.5	sat	23.4	sat	25.5	sakt	27.9	sakakt	28.8
37	saGlIk	salnk	25.0	saIk	27.0	saIk	29.1	salIaInk	31.9	salInk	33.8
38	Sanayi	saiei	26.6	sanayei	28.5	sanai	31.0	sanayi	34.3	sanaeyi	35.7
39	Sarj	Saj	19.5	SaCj	20.6	Saj	22.6	SaCj	24.6	Sarj	26.1
40	Saz	saz	17.9	saz	19.2	saz	20.8	saz	22.9	sazs	24.2
41	seCim	sCim	28.4	seCim	30.6	seiCim	33.7	seCiyn	36.8	secCim	38.5
42	sessizlik	seseik	42.6	sesik	46.4	sesiezik	50.3	seisesiyk	55.3	sesezik	57.7
43	sistemi	stei	31.8	sisei	34.1	sistei	37.2	szstei	40.8	sizsteni	42.9
44	Soyle	OUele	23.5	Caele	25.1	SCOUle	27.6	CSOUele	29.8	CSCOUyle	31.6
45	takIm	Ikan	24.2	aikIn	26.0	taIkaIn	27.9	taIkaInm	30.8	takan	32.3
46	tanIma	aIa	27.0	taInak	29.3	tauInmak	31.8	tauInak	35.0	taInma	36.7
47	Uzerinde	sinde	44.0	izsdine	46.8	Oizsndne	51.0	Ozeine	56.1	Oaizsnine	58.6
48	Volkan	okan	27.9	bolnkan	30.3	bolnkan	33.0	bnounkan	36.2	nuolnkan	37.5
49	yakIn	knm	21.5	ikIn	22.7	yakIn	24.7	kIn	27.2	yakIn	28.5
50	yardImcI	aI	37.1	ancIn	39.7	yadancI	43.3	yancIb	47.1	yeadancI	49.7

Table VIII.2. Comparison of Recognitions Results of Different Split Ratios.

The response time of recognition depends on the length of the utterance. For example, recognition of a 5-letter word takes about 25 seconds (for split ratio of 1.8). About 18 second of this time is spent on HMM Viterbi search because the system performs 591 HMM Viterbi search of each little part of the unknown utterance. This part of the system needs some speed up in order to be practical. Several strategies can be developed in order to reduce the number of Viterbi searches by guessing and eliminating the phonemes which likely to result in lower score, etc.

One reason for the long response times found in the tests performed is MATLAB which the recognition system was developed with. MATLAB provides a very high level programming environment. Programs written in MATLAB are not compiled, instead, they are interpreted. Compiled C or C++ code typically runs faster than its MATLAB equivalents because compiled code usually runs faster than interpreted code. C or C++ can avoid unnecessary memory allocation overhead that the MATLAB interpreter performs [20]. Thus, it is likely that the recognition system can be made faster by converting MATLAB code to C or C++ code.

CHAPTER IX

CONCLUSION

In this thesis, a speaker dependent, large vocabulary, isolated word speech recognition system is developed for Turkish language.

A combination of acoustic-phonetic approach and stochastic approach to speech recognition is proposed and developed. From the acoustic phonetic approach, a phoneme based modeling is adopted. Furthermore, the template classifier and pattern recognizer are built using Hidden Markov Models, which is of stochastic approach.

Phonemes are used as the smallest unit for recognition. Detection and automatic segmentation of phonemes of the training words are performed with an algorithm devised which is based on the generalization that vowels are uttered about 3 times longer than consonants in Turkish, with a few exceptions handled specially.

Detected phonemes are modeled by 2-state HMM models which use 22-dimensional MFCC feature matrices as the observation vectors.

During the recognition phase, the unknown utterance is split into parts not larger than the smallest phoneme trained. Then those parts are searched against the HMM phoneme models trained. The concatenation of the scores of those parts makes up the whole result. In this thesis, the proposed recognition strategy was described in detail and the experiment results were presented.

The strategies and algorithms adopted in this thesis can be extended to be used with Turkish language models which may use grammar rules to improve both the system response time and recognition rate.

REFERENCES

[1] Rabiner, L.; Juang B.: "Fundamentals of Speech Recognition", Prentice Hall, Englewood Cliffs, New Jersey, (**1993**).

[2] Keller, E.: "Fundamentals of Speech Synthesis and Speech Recognition", John Wiley & Sons, New York, USA, (**1994**).

[3] Markowitz, J.A.: "Using Speech Recognition", Prentice Hall, (**1996**).

[4] Yılmaz, C.: "A Large Vocabulary Speech Recognition System for Turkish", *MS Thesis*, Bilkent University, Institute of Engineering and Science, Ankara, Turkey, (**1999**).

[5] Mengüşoğlu, E.: "Rule Based Design and Implementation of a Speech Recognition System for Turkish Language", *MS Thesis*, Hacettepe University, Inst. for Graduate Studies in Pure and Applied Sciences, Ankara, Turkey, (**1999**).

[6] Mengüşoğlu, E.; Deroo, O.: "Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language", Faculté Polytechnique de Mons, TCT Labs, Mons, Belgium, (**2002**).

[7] Zegers, P.: "Speech Recognition Using Neural Networks", *MS Thesis*, University of Arizona, Department of Electrical Engineering in the Graduate College, Arizona, USA, (**1998**).

[8] Woszczyna, M.: "JANUS 93: Towards Spontaneous Speech Translation", *IEEE Proceedings Conference on Neural Networks*, (**1994**).

[9] Somervuo, P.: "Speech Recognition using context vectors and multiple feature streams", *MS Thesis*, (**1996**).

[10] Nilsson, M.; Ejnarsson, M.: "Speech Recognition Using HMM: Performance Evaluation in Noisy Environments", *MS Thesis*, Blekinge Institute of Technology, Department of Telecommunications and Signal Processing, (**2002**).

[11] Hakkani-Tur, D.; Oflazer, K.; Tur, G.:. "Statistical Morphological Disambiguation for Agglutinative Languages", *Technical Report*, Bilkent University, (**2000**).

[12] Ursin, M.: “Triphone Clustering in Continuous Speech Recognition”, *MS Thesis*, Helsinki University of Technology, Department of Computer Science, (**2002**).

[13] Mengüşoğlu, E.; Deroo, O.: “Confidence Measures in HMM/MLP Hybrid Speech Recognition for Turkish Language”, Proceedings of the ProRISC/IEEE workshop, (**2000**).

[14] Cappé, O.: “H2M: A set of MATLAB/OCTAVE functions for the EM estimation of mixtures and hidden Markov models“, ENST dpt. TSI / LTCI (CNRS-URA 820), Paris, France, (**2001**).

[15] www.dspguide.com/zipped.htm: “The Scientist and Engineer's Guide to Digital Signal Processing” (Access date: March **2005**).

[16] Brookes, M.: “VOICEBOX: a MATLAB toolbox for speech processing”, *www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html*, (**2003**).

[17] Davis, S.; Mermelstein, P.: “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 4 (**1980**).

[18] Skowronski, M.D.: “Biologically Inspired Noise-Robust Speech Recognition for Both Man and Machine”, *PhD Thesis*, The Graduate School of the University of Florida, (**2004**).

[19] Aydın, Ö.: “Development of a Speech Recognition System Using Artificial Neural Networks”, *MSc Thesis*, Trakya University, Graduate School of Natural and Applied Sciences, Departmant of Computer Engineering, (**2005**).

[20] MATLAB Product Help: “MATLAB Compiler: Introducing the MATLAB Compiler: Why Compile M-Files?”, (**2001**).

APPENDIX A

HIDDEN MARKOV MODEL (HMM)

Hidden Markov Models (HMM) are the most widely used technique in modern speech recognition systems. This is due to the fact that a great deal of effort has been devoted in research during 1980's and 1990's, making it very challenging for alternative methods to get even close to their performance with moderate investments.

Markov models were introduced by Andrei A. Markov and were initially used for a linguistic purpose, namely modeling letter sequences in Russian literature. Later on, they became a general statistical tool.

Markov models are finite state automata with probabilities attached to the transitions. The following state is only dependent on the previous state.

Traditional Markov models can be considered as 'visible', as one always knows the state of the machine. For example, in the case of modeling letter strings, each state would always represent a single letter.

However, in hidden Markov models the exact state sequence that the model passes through is not known, but rather a probabilistic function of it [12].

An HMM model is a finite state machine that changes state at every time unit as shown in Figure A.1. At each discrete time instant t, transition occurs from state i to j, and the observation vector $\mathbf{o}_t$ is emitted with the probability density $b_j(\mathbf{o}_t)$. Moreover, the transition from state i to j is also random and it occurs with the probability a_{ij}.

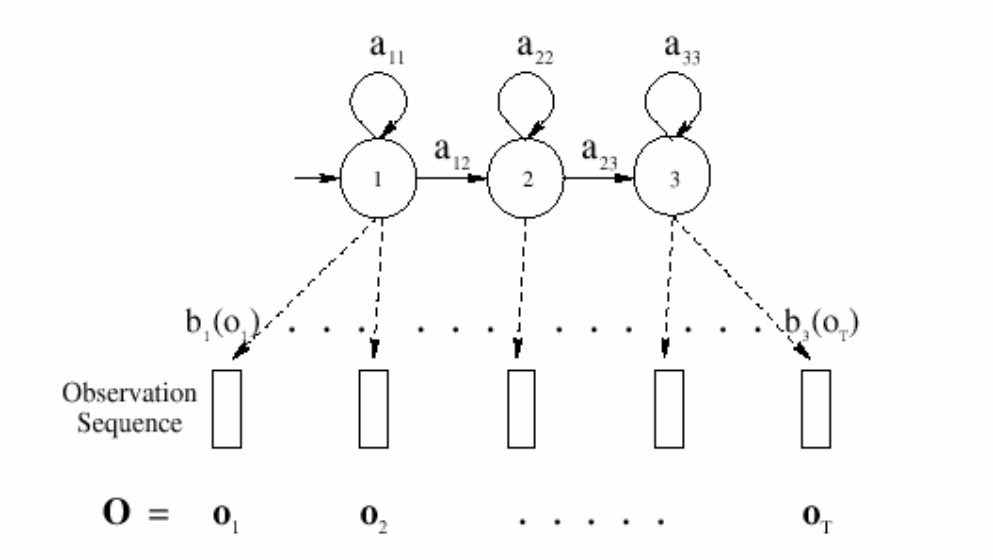


Figure A.1. A Three-state Left-to-Right HMM Model with the Observation Vectors Each Being Generated by One State (State 1 Represents the Start State).

The underlying assumption of an HMM model in speech recognition problem is that a speech signal can be well characterized as a parametric random process, and the parameters of the stochastic process can be estimated in a precise and well-defined manner. An HMM model is considered as a generator of observation sequences (i.e. feature vectors). In practice, only the observation sequence is known and the underlying state sequence is hidden. That is why this structure is called a Hidden Markov Model. This chapter gives a short introduction to the theory of HMM models for speech recognition process [4].

A.1.1. Elements of HMM

A complete specification of an HMM model requires specification of (1) two model parameters, N and M, (2) observation symbols, and (3) three sets of probability measures A, B, π. The definitions of these parameters are as follows:

1. The parameter, N, is the number of states in the HMM. The individual states are labeled as $\{1, 2, \ldots, N\}$, and the state at time t is denoted as q_t.
2. The parameter, M, is the number of distinct observation symbols per state. The observation symbols represent the physical output of the system being modeled. The individual observation symbols are denoted as $V = \{v_1, v_2, \ldots, v_M\}$. Only in HMM models for discrete observation symbols the parameter M is defined. HMM models for continuous observation sequences, as we have in

this thesis, clearly do not have the parameter M, but have an observation set whose elements are continuous variables.

3. The matrix, $A = \{a_{ij}\}$, is the state transition probability distribution where a_{ij} is the transition probability from state i to j, i.e.,

$$a_{ij} = P(q_{t+1} = j \mid q_t = i), \quad 1 \le i, j \le N \tag{A.1}$$

If a state j cannot be reached by a state i in a single transition, we have $a_{ij} = 0$ for all i, j.

4. Let $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$ be the set of observation symbols. The matrix $B = \{b_j(\mathbf{o}_t)\}$, is the set of observation symbol probability distribution in which

$$b_j(\mathbf{o}_t) = P(\mathbf{o}_t \mid q_t = j), \quad 1 \le t \le T \tag{A.2}$$

defines the symbol distribution in state j, $j = 1, 2, \ldots, N$. In speech recognition problem, observation symbols are feature vectors.

5. The vector, $\pi = \{\pi_i\}$, is the initial state distribution, in which

$$\pi_i = P(q_1 = i), \quad 1 \le i \le N. \tag{A.3}$$

For convenience, we use the compact notation

$$\lambda = (A, B, \pi) \tag{A.4}$$

to indicate the complete parameter set of an HMM model. This parameter set defines a probability measure for a given observation sequence $\mathbf{O}$, i.e., $P(\mathbf{O} \mid \lambda)$. We use HMM model to indicate the parameter set λ and the associated probability measure interchangeably without ambiguity [4].

The mathematical details of HMM will not be covered in here. In short, the three main problems that have to be solved for utilizing the models in speech recognition are:

1. The evaluation problem: Given a model and a sequence of observations, one needs to compute the probability that the sequence was produced by the model.

This can be also seen as the problem to score the match between the observations and the model. For this problem an exact solution exists and can be efficiently calculated by using the *forward-backward* algorithm.

2. The estimation problem: Given an observation sequence or a set of sequences, this problem involves finding the parameter values that specify a model most likely to produce the given sequence. This problem is involved in speech recognition in the training phase, and is solved iteratively using the *Baum-Welch* algorithm.
3. The decoding problem: The third problem involves finding the most likely state sequence for a given observation sequence. There are different search algorithms for this, for example, the *beam search* algorithm, V*iterbi search* algorithm, etc. [12].

In this thesis, phonemes are used as the smallest unit for speech recognition. Each phoneme is represented by a 2-state left-to-right HMM model as illustrated in Figure V.1. We make use of HMM models both in training and recognition stages of the system.

For the implementation of HMM in the system developed, functions provided with *H2M matlab toolbox* are used. H2M is defined by its authors as a set of MATLAB/OCTAVE functions that implement the EM algorithm in the case of mixture models or hidden Markov models with multivariate Gaussian state conditional distribution [14].

APPENDIX B

TRAINING WORD LIST

açık	bize	etmek	istedi	savunma
ajan	bizim	farklı	kadar	sendika
alan	böyle	gece	kadın	şey
amacıyla	bugün	gelecek	kalan	şimdi
anadolu	bulundu	gelen	konu	sonra
anayasa	bundan	genel	kullanım	sonunda
arada	bunlar	genellikle	lira	söyledi
ayrı	çalışan	gereken	merkez	tarihinde
ayrıca	çalışkan	gibi	mesut	telefon
bağlı	çalışma	hakkında	mustafa	türkiye
bana	çıkan	hakları	nasıl	yabancı
başarılı	çocuk	halinde	neden	yapılması
başkanı	çünkü	haline	olabilir	yapmak
belediye	daha	hareket	olan	yaptı
belki	derece	herhangi	olay	yatırım
benim	destek	herkes	olumlu	yeni
bin	durumda	herşey	önce	yerine
biraz	eden	içinde	önemli	yılında
birinci	erken	insan	özellikle	yönetim
birlikte	eski	istanbul	sadece	zaman

APPENDIX C

TESTING WORD LIST

açısından	çarşamba	gerçek	konuşma	seçim
akışkan	cesaret	haksızlık	meydana	sessizlik
alışkanlık	çetin	imkansız	murat	sistemi
amerika	dakika	istakoz	nedeniyle	şöyle
aranjman	değerli	istanbulda	olasılık	takım
aydınlık	devlet	işyeri	saat	tanıma
barış	eğitim	izmir	sağlık	üzerinde
başbakan	evet	jale	sanayi	volkan
bilgi	gelenek	jöle	şarj	yakın
bizimle	genç	kompüter	saz	yardımcı

APPENDIX D

TEST RESULTS

In the following pages, 5 test results are given as one table per page.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaCCCSCsssssanInnnadaaaannnnny	aCsnan	38.3
2	alISkan	paaaakkkekSSCSCkkkaaaannnmnb	akSkan	36.6
3	alISkanlIk	haaaIlIIIcSSSkCaaamunlaIUInkkkk	aISak	39.9
4	amerika	baaanmeieieeiiynkkaaaan	aeika	29.0
5	aranjman	haaaaaaannaibkyuumaannninbl	anuan	34.0
6	aydInlIk	aaaeiiiitInnannnaUnnkkkkkk	aink	33.1
7	barIS	aaaaIIlIeIIjjSSSt	aIjS	21.5
8	baSbakan	nbbaaaejjSStaaOakkkaakn	bajSaka	29.7
9	bilgi	iikeiiiypgiieiiiy	i	21.8
10	bizimle	biiisssziieeemilleeeeek	isiele	29.0
11	CarSamba	iSyaaaajSSSSaanuaaabbaaak	aSaba	32.0
12	cesaret	CCidessssssaadaraaaeeeettsttt	Csaet	35.5
13	Cetin	CCSeeietidIdnnnyg	Cen	22.3
14	dakika	dIaeICSieeIgikkOaaaa	eka	25.7
15	deGerli	diiieeeeeeerrlIggyiiiii	iergi	29.4
16	devlet	deeeeerIiIeleeetesttt	et	26.5
17	eGitim	iieeeieiiiyyktieeamnmdmn	ieiye	30.3
18	evet	eeeeIeeeeeebteet	e	20.7
19	gelenek	ggeeeieeeeneeeeiikSkkk	geik	28.4
20	genC	yeeitnanncCCS	enC	16.2
21	gerCek	bbciaeeaellrIykCCCSeeeeiiCCkk	belCeiCk	37.3
22	haksIzlIk	aaakkkkstsssaazzaaIaIImnekkk	aksazaIk	36.2
23	imkansIz	yieeiimnkkeaanaassssaaaazzzss	eikasazs	37.2
24	istakoz	iiisssssstaaaakuauauassss	isas	30.4
25	istanbulda	yiisssstttaaaaaaIbuuluulipaaankb	istaua	40.0
26	iSyeri	iiiSSCCSCkkgiiiieiiIeekiib	iSCkiei	33.1
27	izmir	giiiiezisinniiiieccrrSS	inicrS	29.8
28	jale	CkjCraaaaanudeeeet	ae	22.8
29	jOle	ndCCjCCiOrIIeeeeaei	CIe	24.4
30	kompUter	kkfoouuammpkmdnkmeeaaaSki	koumea	32.3
31	konuSma	kuoauuanCCCCdpImaaaakb	uCa	28.0
32	meydana	niaeeeeiiIaaanaaaIIdk	eiaI	26.7
33	murat	muuouaraaaaaktttttt	uat	22.8
34	nedeniyle	nndeeeieeeniiiiiiiyiieleeih	neie	34.1
35	olasIlIk	ooollllaaazsssslIIIlrIUnkkkkkkk	olasIk	39.5
36	saat	sssaaabaaaaakpttt	sat	21.5
37	saGlIk	sssaaaaaaallarnnkkkk	salnk	25.0
38	sanayi	sssaaanIaaaaOaiieeiii	saiei	26.6
39	Sarj	SSScaaaaaCrjjjj	Saj	19.5
40	saz	sssafaaaaazzzs	saz	17.9
41	seCim	sssseiIetCCSiiiimmmnln	sCim	28.4
42	sessizlik	sssseeeeissssiseezisbsyiiyugiCkkh	seseik	42.6
43	sistemi	sssseszsssttydeeeiIeiibn	stei	31.8
44	Soyle	CSCSCOOUUeelleeeee	OUele	23.5
45	takIm	tCIIikkkaaInannmynk	Ikan	24.2
46	tanIma	taaaaaaaIIndnmaaaaaak	aIa	27.0
47	Uzerinde	mOaiszssdendieiciinnnnnnnnddeeeeee	sinde	44.0
48	volkan	bniboooalunkkCaaaanndn	okan	27.9
49	yakIn	yiyIbIkkIaInnnmmy	knm	21.5
50	yardImcI	lyeaaaaaaanpdandniCScIIInbknb	aI	37.1

Table D.1. Test Results for Split Ratio = 1.2.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaaCCCSScsssssssannInIanddaaaannninnn	aCSsndan	40.7
2	akISkan	kaaaaaakkkekIScCSCSkkCaaaannnnnmnbb	akanb	39.5
3	alISkanlIk	kaaaoIlIIIccISSkkCaaaauunulaIIuIbkkkkd	aIcSkauIk	43.2
4	amerika	aaaanuaiiieiIiiilyyykaaaaaaa	aiya	31.3
5	aranjman	haaaaaaaaannnaaikypnmmaannnninmm	anamanm	36.7
6	aydInlIk	aaaaiiiimiIInnaununuaUninkkkkkk	aiInk	35.4
7	barIS	aaSaarrIIIeaIjSjSSSt	arIS	22.8
8	baSbakan	gbbaaaaSejjjSSgaaOaakkkaanknn	bajSakan	32.5
9	bilgi	ibiieiiiiypgiieeeimmy	iem	23.6
10	bizimle	bbIiieeszziiieeemIeleeeeeeek	biezie	31.3
11	CarSamba	iSeaaaaaajSSSSaaamnaaaabbbaaap	aSaba	34.0
12	cesaret	CCcddesssssadadararIaeeeettntsttt	Cdset	38.1
13	Cetin	CCCeiiisetidIndnnmbgg	Cing	23.9
14	dakika	paaeeIkCSieeIggikkkaaaaaa	aegka	28.1
15	deGerli	diiiieeeeeeeernIlIyggyiieiit	iegi	31.8
16	devlet	deeeeeedIIiIieeeeetntsttt	eIet	28.3
17	eGitim	bkeieeeeiiiimupktiliiamnnmmmn	einm	32.7
18	evet	eeeeUbUeeeeeeibteett	et	22.5
19	gelenek	gggeeeeieeeemieeeeiiikSikki	geik	30.8
20	genC	yeeeitnaanncCCS	eanC	17.2
21	gerCek	bnkhiadeeaelnrIgigCCCCCeeeeeiikCCkkS	eCeiCk	40.3
22	haksIzlIk	aaakkkkkmstssssaazzsdaInIIImIdkkkkk	aksazIk	39.0
23	imkansIz	yieieiimmnkCeaaanaIsssssaaaaazzzzss	imasazs	39.5
24	istakoz	iiisssssstaaaaakuaaIuuasssss	isaus	32.0
25	istanbulda	niiisssssttadaadaaIIbuuuuuulimaaaaakkt	istaIuak	42.7
26	iSyeri	iiicSSCCCSCkiggiiiiieieIyeekkitk	iSCgiek	35.6
27	izmir	giiiiiszziiIunniiieeiccrrkCSC	iziniecr	32.1
28	jale	CkjjSraaaaaaeIleeeeeet	jae	24.3
29	jOle	ndCjSCCCIIOrrIleeeeeeri	CIre	25.8
30	kompUter	kkfooouuaimmmfpmdngmIreeaaeSiii	koumeai	34.7
31	konuSma	kuuoauunancSCCCdpIbaaaaaakd	uCa	30.0
32	meydana	niaeeeeeeimdldaanaaaIIaIkk	eaIk	28.7
33	murat	nauuuunraraaaakmpttttt	uat	24.4
34	nedeniyle	nnneeeeebdeennmiiiiiiiyyneeeeeee	neniye	36.1
35	olasIlIk	oooollldIIaszssssaIIlIlaIIUnnkkkkkkkf	olIsInk	42.3
36	saat	ssstaaapkaaaaaakptttt	sat	23.4
37	saGlIk	ssssaaaaaaaauIIarInIkkkk	saIk	27.0
38	sanayi	ssssaaannaaaaaOeiyygeeiii	sanayei	28.5
39	Sarj	SSSSaafaaaaCCjjjjj	SaCj	20.6
40	saz	ssssaaaataaazzzzs	saz	19.2
41	seCim	sssseeeeIicCCCSiiiimmymnmmn	seCim	30.6
42	sessizlik	sssssseeeeeissssssssieszizsysyiieyugiSCkkeI	sesik	46.4
43	sistemi	sssssesiissssstbyteeeiediiiiii	sisei	34.1
44	Soyle	CSCCSaaOaUieelleeeeeee	Caele	25.1
45	takIm	taaIaiikkkaIIununnumynk	aikIn	26.0
46	tanIma	ttaaaauanaIIInnnmaaaaaaakk	taInak	29.3
47	Uzerinde	mUOiiszzssyenedddiciiinnnnnennnbdeeeeeeee	izsdine	46.8
48	volkan	bbnpuoooollknnkkCaaaaanndnn	bolnkan	30.3
49	yakIn	iiayeIUkkkIaIImnnmnb	ikIn	22.7
50	yardImcI	lyiaeaaaaaaandpdannnnicckcrIImnnknb	ancIn	39.7

Table D.2. Test Results for Split Ratio = 1.5.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaaOCrCSSccsssssssaannInnIIndddaaaaannnnnnnn	aScsanIdan	44.4
2	akISkan	kaaaanapkkkIekkSccCSSSkkkkaaaaannannnnnll	akcSkanl	42.6
3	alISkanlIk	kaaaaoIllIIIccISSSkkkSaaaamuunullOIIuInbkkkkk	alIcSkaulIk	46.6
4	amerika	aaaaanunmmeieiIeeiiiiyykkkaaaaaak	ameiyka	34.0
5	aranjman	haaaaaaaaaannnmanibjsypummaaannnnnnnml	anman	40.0
6	aydInlIk	aaaaaiiiiiikIIndnauunnnuaUUinbkkkkkkk	aiIunUk	38.4
7	barIS	bbaaaaarreeIIeaSjjSjSSSS	bareIjS	24.9
8	baSbakan	bbbbaaaaaSejjjSStgaaaOaakkkOaannkm	bajSakan	35.4
9	bilgi	bbkieeiiiiyypgiiieiiiiiny	beiyi	25.8
10	bizimle	bbIeiisisszzieieeeeImmilleeeeeeke	biszemle	33.9
11	CarSamba	iSScaaaaaaCjSSSSaannanuaaaaabbbaaaka	SaSanaba	37.3
12	cesaret	CCciddeesssssssaadadaarraIaeeeeeettnssttt	Cdesaretst	41.4
13	Cetin	CCCSeiiiiettienIndnnnnmng	Citn	26.0
14	dakika	paaaeeIkCkceeeiigiyCkkOaaaaaa	aeika	30.3
15	deGerli	ddiiiieeeeeeeeeerrnIllygggiiieeii	dierlgiei	34.3
16	devlet	ddeeleeeidIIiiIleeeeeetntsittt	deIiet	31.0
17	eGitim	nyieeeeeieiiiymynykttiliieannnmmmkb	eitinm	35.8
18	evet	eeeeenbUeeeeeeiekbteeett	et	24.6
19	gelenek	ggceeeeeieeeeeenneeeeeiiiuSckkkir	geneik	34.0
20	genC	yyeeiitnnaanngcCCS	yeinanC	18.7
21	gerCek	bbbkceaedeeaenllcIgymkCCCCCceeeeikkikCSjkkS	belCek	44.4
22	haksIzlIk	hhaaakkkkkkestssssaaazzzsaaaIpIIInydmkkkkk	haksazaIk	42.5
23	imkansIz	kiieigiimmnnkCeeaaannIasssssaaaalaazzszsss	imneansazs	43.6
24	istakoz	giiisssssssttaaaaaakkuvaIaluasssss	istaks	35.2
25	istanbulda	niecsssssstttadaaadaannIIbuolluumuiibaaaaaakbb	stanIluiab	47.9
26	iSyeri	iiiiSSSCCCCSCkikggiiiiieeirreIyeekiit	iSCgierei	39.2
27	izmir	giiiiiiezzzsidunnniiiiiidiccrrrrkS	iznicr	35.2
28	jale	CkjjjCreaaaaaanInlyeeeeemt	jae	26.7
29	jOle	ndCcjjjjCCcIOOrrIleeeeeeerei	jCOre	28.7
30	kompUter	kkkfoooluuaamdmpppmrUnmmmeeeeaaaCSiii	kouapmeai	38.3
31	konuSma	kkuooauunnanoSCCCCdpaIbaaaaaaakd	kounCa	32.9
32	meydana	nimmaeeeeeeiipladaannaadIIaIdkk	meianaIk	31.6
33	murat	nnuubuunorraaaaaakmpttttttt	nurat	26.9
34	nedeniyle	nnndeeeeeiIeeeendniiiiiiiiiyyneellеeteb	neiyele	39.9
35	olasIlIk	ooooolllldaaIaszsssostlIlIIIlaIIUUnnkkkkkkkkk	olasIUnk	46.3
36	saat	ssssdaaaabkaaaaaIakCttttt	sat	25.5
37	saGlIk	ssssaaaaaaaaaaulIIIIIInIkkkk	saIk	29.1
38	sanayi	sssstaaannnaaaaaaOeayimgiieiei	sanai	31.0
39	Sarj	SSSSkaaaaaaaaaCrjjjjjS	Saj	22.6
40	saz	sssstafaaaaaaaazzzzs	saz	20.8
41	seCim	sssssteeiiingcCCSiiiIimammmnblnmb	seiCim	33.7
42	sessizlik	sssssseeeeeeissssssiieeeszzzsysbieiieyuigiSCkkkhI	sesiezik	50.3
43	sistemi	sssssseiizsssssssttyteeeeeeidiiiiiim	sistei	37.2
44	Soyle	CSSCCSCaOOUUOyeillceeeeeeel	SCOUle	27.6
45	takIm	ttaaIIIikkkkaaIIunnnnnumyby	taIkaIn	27.9
46	tanIma	ttaaaaauundaIIInInnnmmaaaaaankk	tauInmak	31.8
47	Uzerinde	iOOaiiszzsssyennedddyicieinnnnnnnnunnndeeeeeeeeee	Oizsndne	51.0
48	volkan	bbmnpbooooolluknnkkCaaaaaaanndnp	bolnkan	33.0
49	yakIn	iyyaaUeIUIkkkIICdnmnnnmn	yakIn	24.7
50	yardImcI	lyygayaaaaaaaadntddaandnnaCccCcIIIImn	yadancI	43.3

Table D.3. Test Results for Split Ratio = 1.8.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaaaOCCCCSCCcsssssssaannnnInnIInnddaaaaaaaanannninunnm	aCsanIndan	48.7
2	akISkan	kfaaaaapaakkkkIIkkSSSSCCSSCSkkkkaaaaaaannnnnnnmmbllb	akIkSCSkanml	46.9
3	alISkanlIk	khaaaaaolllIIrIIccISSSSkkkCaaaaaanuunnmmlaOIUuIIbkkkkkkk	alIcSkaunmIk	50.8
4	Amerika	aaaaaanaunmeieieieIeeiiiliyylkkkaaaaaaaak	aeiyka	37.0
5	Aranjman	haaaaaaaaaaaaaannnmanabhjIpyuumaaaaaannnnnmnmmln	anuanm	43.5
6	aydInlIk	aaaaaamaiiiikiikIIndnaaunnnnuaarInnnnykkkkkkkkk	aiIanank	42.5
7	barIS	baaSaaaarrIIeiIencISjSSjSSSSSt	arIS	27.3
8	baSbakan	nibbbaaaaaaSeeSjjSSStapaaaOaaekpkkOaarnkkdn	baejSakak	38.9
9	Bilgi	bbiiieeiiiiiiyypggiiiieeiiimmny	bieiygieim	28.0
10	Bizimle	bbbIgiississzzziieieeeeemmIelleeeeeeeeeek	bisziemle	37.0
11	CarSamba	iSSeyaaaeaaeCjjSSSSSaaanaamnuanaaadbabaaaaapk	SajSa	40.6
12	Cesaret	CCCiiddeeessssssssaaadaadaarrrlIleeeeeeeiittnbsstttt	Cidesaretst	45.2
13	Cetin	CCCCSeiIiiisettiidIIddnnnnnbyggb	CitiIdng	28.8
14	Dakika	paIaUeIbpkCSScieeiiiggmiCkkkaaaaaaaaa	Seigka	33.3
15	deGerli	diiiieieeeeeeeeeeeeernrllnIydgggyiieieiIit	ielgi	37.8
16	Devlet	edeeeleeeeerdIIiiIIleleeeekttntsssttt	eIiIetst	34.1
17	eGitim	nyiieeeeeeieiiiimiynuggtthiieidimdnnnmmbbbn	ieigtinmb	38.9
18	evet	eeeeeeeIIUeeeeeereeetbtteetttm	eIetet	27.0
19	gelenek	ggcieeeeeeiiieeeeemIneeeeeeiieiugSckkkkkr	geieik	37.0
20	genC	yyeeeiittnnaandnucCCCCS	yeitnaC	20.7
21	gerCek	innbchiaeydleyalnlllcrIgynkCCCCCCCceeeyeekiiikCCCjkkS	nlCeiCk	48.3
22	haksIzlIk	haaaakkkkkkkmsstsssssaaaazzzzsadaIIpaIIInyImmkkkkkkk	aksazImk	46.9
23	imkansIz	kiiieieeiimmynukkCeeaaaanavlzsssssssaaaapaazzzzzzzsssz	ieimkeasazs	47.9
24	istakoz	giiiisssssssssttttdaaaaaakkkaaoIalluaasssss	istakalas	38.8
25	istanbulda	nyiicsssssssstttttadaaaadaIunIIbnuolluuuullilguaaaaaankbnn	istaIlulan	51.6
26	iSyeri	iiiiiSSSCCCCCSSCkkikggiyiiiiieeiieiIeeekkiiitk	iSCSkgieieki	42.5
27	izmir	giiyiiiissszzzisakunnnnuimieidigccSrrrSkSSC	iszncrS	38.6
28	Jale	CCkCSjSryeaaaaaaannanldeeeeaetet	Cane	29.0
29	jOle	nnbCjjSCjCCCIiaaOrrrIleeeeeeeeeeeek	njCare	31.3
30	computer	bkkkfooooluuaanmmbmppkmUrmnmkmmereeeaaaaCSSiii	kouampmeaSi	41.8
31	konuSma	kauuooouuuunaanocScCCCddpaIbaaaaaaaaakkd	uouaCdak	36.1
32	meydana	gbimOeeeeeieeeiimdIddaaannnaddIIInIIkf	eidandI	34.4
33	murat	nmuuubuulnoaaaaaaaakakkmpttstttt	uakt	29.1
34	nedeniyle	mmnndeeeeeeeiIedeeennneiiieiiiiiiiyynieeeeeeerbt	mneniye	43.5
35	olasIlIk	oooooollllIlldaIalaszssssossalIlulIIlarIIUUnntkkkkkkkkkk	olsIUnk	50.7
36	saat	sssssstaaaaaakaaaaaaaIakkdtstttt	sakt	27.9
37	saGlIk	sssssaaahaaaaaaaaallIIaaIIInnIkkkkk	salIaInk	31.9
38	sanayi	ssssssaaaannnuaaaaaaaaOaiyyiieieieieim	sanayi	34.3
39	Sarj	SSSSSCraaaaaaaaraCCrjjjjjkS	SaCj	24.6
40	saz	sssssstaaaaaaaaaaaazzzzszs	saz	22.9
41	seCim	tsssssetieeiIengcCCCCSiiiUiimnmyymynmlnnb	seCiyn	36.8
42	sessizlik	sssssssseeeeeeeeiissssssssiseeesszszssbybIyiiieyyuygkiSCkkketI	seisesiyk	55.3
43	sistemi	sssssssetsizzsssssssttgttteeeeeleiniiiiniimn	szstei	40.8
44	Soyle	CCCCCCSSraOOaUUUyeeyllkceeeeeeeee	CSOUele	29.8
45	takIm	ttaaaIICnmkkkkCaaIIIunannnnummykbk	taIkaInm	30.8
46	tanIma	tttaaaaaaauuandaIIIInnndnnnmaaaaaaaaaakkh	tauInak	35.0
47	Uzerinde	iOOOniaOzzzzzzseieneediidiiiiieiennnnnnnnnnudnnbdeeeeeeeeeecbk	Ozeine	56.1
48	volkan	nbbnnpbuoooooaluuunnnkkkCaaaaaaaannndnnn	bnounkan	36.2
49	yakIn	iyiayOIamUkkkkIIIaIdnnnnnndnib	kIn	27.2
50	yardImcI	lnyyiaeaaaaaaaaabadndCkdaamndnnnCicckcIrIIIIIbbnknbb	yancIb	47.1

Table D.4. Test Results for Split Ratio = 2.1.

No	Original	Raw result	Final Result	Time (sec)
1	aCIsIndan	aaaaOCICSSCcccsssssssssaannnnInnnnInnddddaaaaaaaaannnnnnnnny	aScsandan	51.3
2	akISkan	kraaaaknpakkkkkIeIkkSSSSCCSSSSkkkkkaaaaaaaaanrunnnnnmnlb	akSCSkan	49.1
3	alISkanlIk	bhaaaaaollllIIrIccccSSSSkSkkCaaaaaamnnuunulluaOIIuIbnbkkkkkkk	alIcSkanulIk	53.4
4	amerika	faaaaaaannameiiieyimieeiiiiiyygbkkkaaaaaaaaan	anieiyka	39.2
5	aranjman	haahaaaaaaaaaaaanunnnnaiibackyypuummaaaaannnnnnmnmnbnl	aniyuman	46.0
6	aydInlIk	aaaaaaaeaiiiimiykIIInnnaauanunnuIIrInninnkkkkkkkkk	aiInanInk	44.5
7	barIS	bbaaaaaaaIrreIeiOinaSSjjSSjSSSCtSt	barSjS	28.9
8	baSbakan	bgbbabaaaaaeSeejjjjSStyapaaaOOaakknkkOalanakbm	baejSaOak	40.4
9	bilgi	bbiiikieeiiiiiyyyggiiiiiieiiyiimmyg	bieiygim	29.7
10	bizimle	bbIgiiisssssszzziieiieeeeemmiillleeeeeeeekek	bisziemile	39.1
11	CarSamba	iSSSyyaaaaaaeCSjSSSjSCaaaaaamnuaaaaaabbababaaaapk	SyaSaba	42.8
12	cesaret	CCCciideeezsssssssssaadaadaaararrIaeeeeeeeeittnbsssttttt	Ciesaretst	47.8
13	Cetin	CCCCCeeiIliiseteiiedInndnnnnnmbybg	Cein	29.9
14	dakika	paIaaUeedpgCSSceeeeiIgggmiCkkkkaaaaaaaaa	aeSegka	35.0
15	deGerli	ddiiiiieieeeeeeeeeeeeeerrrIlnrylgggiyiiieiiiii	diergi	39.4
16	devlet	ddeeeeeeeeelrIIIIiIgielleeeetetnitsssttttt	deIlest	35.7
17	eGitim	nyeieeeeeeeieiiiiiiimydyggkttiiliieimdnnuyldmmbn	eigtinm	40.9
18	evet	eeeeeeenIIUeeeeeeieeeekbpteeettt	eIet	28.0
19	gelenek	gggcieeeeeeiiieeeeeemnneeeeeieeiiiukSckkkrik	geineik	38.5
20	genC	yyeeeeiattnnaaannyhcSCCCS	yetnanC	21.7
21	gerCek	innbkhiyaeedeeeaeenllIcrIggkgkCSCCCCCceeeeeieeeitnSCSjkkkS	nelgCek	50.7
22	haksIzlIk	hhaaaaakkknkkkStststssssozaaaaazazzaadauInaIIIImiImkkkkkkk	haksazaIk	49.1
23	imkansIz	kniieeegiiimmynkkiCaeaaaaannnaIsssssssaaaaIlaazzzzsszszss	ieimkansazs	50.3
24	istakoz	giiiiissssssssssttaaaaaaaakkklvauualluaasssssss	istakulas	40.7
25	istanbulda	nyiicissssssssttttaddaaaadaarIlIbbnuolluuomuuiilyynaaaaaanknkn	istdabluiya	54.2
26	iSyeri	iiiiiiSSSSCCCCCSSkkkigggiiiiiieiieierrIIyeeikeilitb	iSCSkgiIre	44.7
27	izmir	giiiiiiiiisszzzzssadknInnnyiiiidediccrrrrrSikS	iszsnicr	40.2
28	jale	hCkjjCjCreaaaaaaaaaanmunlleeeeeaeemt	jale	30.5
29	jOle	kdddCjjjCSCCCIiIOOrrarIlreeeeeeeeaeeeek	djCOre	33.0
30	kompUter	bkkkpooooooouuaaauidmmpkkmUrInmgcmmereeraaarkcSSiii	kouamkmeaSi	43.8
31	konuSma	kakkoooauuuuoannocSCcCCCddpaImmaaaaaaanaakb	kounCdma	37.7
32	meydana	gbimmeeeeeeeeeeimimpdIadaaannaaaarIIInIIk	meanaI	36.0
33	murat	dmmuuubuulnooaaaaaaaaakabkmpktttttttt	muoat	30.6
34	nedeniyle	mmnndeeeeeeeeiIieeeennnneiiieiiiiiiiiyyylaeelleeeerkh	mneniyele	46.1
35	olasIlIk	ooooooll111111dIaaIdszzsssssssslIIlIlIIllrIIUUUnntkkkkkkkkkkk	olazsIlIUnk	53.4
36	saat	sssssaaaaaaaakkaaaaaaaIakkkttttttt	sakakt	28.8
37	saGlIk	sssssaaahaaaaaaaaaallIIaIaInbnnkkkkkkk	salInk	33.8
38	sanayi	sssssaaaaannnIaaaaaaaaaOeeyyimgiiiieieii	sanaeyi	35.7
39	Sarj	SkSSSSIaaaaaaaaaraaCrrjjjjjSjS	Sarj	26.1
40	saz	sssssaaaaaaaaaaaaaaazzzzzzss	sazs	24.2
41	seCim	tsssssseteeeiIentccCCCSiiieUiiimnmmymnblmnm	secCim	38.5
42	sessizlik	sssssssseteeeeeeeeississsssssseieeszzszistsbsyaiiiiiyuyugiSCkkkket	sesezik	57.7
43	sistemi	sssssssseti izzzsssssssstttgypteeeeeeleinniiieiiimn	sizsteni	42.9
44	Soyle	CCSSSCCSCraOOaUUUiyyeyllkceeeeeeeeee	CSCOUyle	31.6
45	takIm	tteaaaICICnykkkkkIaaaInunannnniubnyknk	takan	32.3
46	tanIma	tttaaaaaaaaaaaadaaIIInnumnnnnmaaaaaaaaaankb	taInma	36.7
47	Uzerinde	iOOOaaiiszzzzzssdyennndiiieiiIciieiennnnnnnnnunudnnldeeeeeeeeeeeete	Oaizsnine	58.6
48	volkan	nnnmnpduuoooooolallulnnnkkkkaaaaaaaaannnndnn	nuolnkan	37.5
49	yakIn	iyyaeaIeaaUkkkkkIIaCIdnnnnnnmnnny	yakIn	28.5
50	yardImcI	kmyygaeeaaaaraaaaandmnkdddaannnbninCiccSjIIrIIInInn	yeadancI	49.7

Table D.5. Test Results for Split Ratio = 2.4.

BIOGRAPHY

Volkan TUNALI was born in Bursa on August 18, 1978. After graduating from Computer Department of Bursa Demirtaşpaşa Technical High School in 1995, he began Marmara University, Faculty of Engineering, Computer Engineering Department in 1996. He graduated from the faculty in 2001, and began his graduate studies in the field of Computer Engineering at Marmara University, Institute for Graduate Studies in Pure and Applied Sciences.