A CONTINUOUS SPEECH RECOGNITION SYSTEM FOR TURKISH LANGUAGE BASED ON TRIPHONE MODEL

by Fatma PATLAR

Istanbul Kultur University, 2009

THESIS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

COMPUTER ENGINEERING PROGRAMME

SUPERVISOR

Assist.Prof. Dr. Ertuğrul SAATÇİ

ABSTRACT

A CONTINUOUS SPEECH RECOGNITION SYSTEM FOR TURKISH LANGUAGE BASED ON TRIPHONE MODEL

PATLAR, Fatma

M.S. in Computer Engineering

Supervisor: Assist.Prof. Dr. Ertuğrul SAATÇİ

August 2009, 73 pages

The field of speech recognition has been growing in popularity for various applications. Such recognition embedded applications include automated dictation systems and command interfaces. Embedding recognition to a product allows a unique level of hands-free usage and user interaction. Our main goal was to develop a system that can perform a relatively accurate transcription of speech and in particular, a Continuous Speech Recognition based on Triphone model for Turkish Language. Turkish is generally different from Indo-European languages (English, Spanish, French, German etc.) by its agglutinative and suffixing morphology. Therefore vocabulary growth rate is very high and as a consequence, constructing a continuous speech recognition system for Turkish based on whole words is not feasible. By considering this fact in this thesis, acoustic models which are based on triphones, are modelled as five state Hidden Markov Models. Mel-Frequency Cepstral Coefficients (MFCC) approach was preferred as the feature vector extraction method and training is done using embedding training that uses Baum-Welch re-estimation. Recognition is implemented on a search network which can be ultimately seen as HMM states connected by transitions and Viterbi Token Passing algorithm runs on this network to find the mostly likely state sequence according to the utterance. Also to make a more accurate recognition bigram language model is constructed. MATLAB is used in processing speech and The Hidden Markov Model Tool Kit (HTK) is used to train models and perform recognition.

The performance of this thesis has been evaluated using two different databases, one of them is more commonly formed TURTEL speech database that is used for speaker independent system tests and the other one is particularly formed weather forecast reports database that is used for speaker dependent system tests.

In recognition experiments, word accuracy of speaker independent system has been measured as 59-63 percent. After finding optimum value for decision tree pruning factor by try outs, system tests have been repeated again by using the language model and the optimum pruning factor. These adjustments improved the performance by 30-33 percent and word accuracy has reached to 92-93 percent for all tests.

While the word accuracy of the speaker dependent system tests on the single person database is between 89-93 percent, usage of the language model and the optimum decision tree pruning factor has resulted with an increase in the performance and the word accuracy has reached to 95-97 percent.

Keywords: Continuous Speech Recognition, Triphone, Hidden Markov Model, Language Modelling, Bigram language model

ÖZET

ÜÇLÜ SES MODELLİ TÜRKÇE SÜREKLİ KONUŞMA TANIMA SİSTEMİ

PATLAR, Fatma

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd.Doç.Dr. Ertuğrul SAATÇİ

Ağustos 2009, 73 sayfa

Konuşma tanıma tabanlı uygulamaların popülaritesi her geçen gün daha da artmaktadır. Bu uygulamalara dikte sistemlerini ve komut arayüzlü sistemleri örnek olarak verebiliriz. Bir ürüne konuşma tanımayı entegre etmek kullanıcıya benzersiz bir kullanım kolaylığı ve etkileşim imkanı sunar. Bizimde buradaki asıl amacımız Türkçe için nispeten hassas çeviri imkanı sunacak geniş kelime dağarcıklı bir sistem tasarlamaktı. Türkçe, sondan eklemeli morfolojisiyle genel olarak Hint-Avrupa dillerinden (İngilizce, İspanyolca, Fransızca, Almanca vs.) farklıdır. Bu yapısı sözcük dağarcığında büyük bir artışa neden olmakta ve sonuç olarak Türkçe için kelime tabanlı sürekli konuşma tanıma sistemlerinin yapılabilirliği pek mümkün olmamaktadır. Bu gerçeğide göz önüne alarak, bu tezde, akustik modeller, beş durumlu Saklı Markov Modelleri olarak modellenmiş üçlü-sesler temel alınarak oluşturulmuşlardır. Özellik vektörü çıkarımı için Mel Kepstral Katsayılar yaklaşımı tercih edilmiş, eğitim ise Baum-Welch yeniden tahmin algoritmasını kullanan "gömülü eğitim" yöntemi kullanılarak yapılmıştır. Tanıma işlemi bir arama ağı üzerinde işleyen Viterbi Token Passing algoritması kullanılarak gerçeklenmiştir. Bu arama ağı aslında model durumlarının geçişlerler birbirine bağlanmış hali olarak görülebilir. Aynı zamanda daha doğru bir tanıma yapabilmek için ikili dil modellemesi de uygulanmıştır. SMM'i, "gömülü eğitim" kullanılarak eğitilmiş; tanıma kısmında ise "Andaç geçirmeli Viterbi algoritması" kullanılmıştır.

Konuşmanın analizi ve işlenmesinde MATLAB; modellerin eğitimde ve tanımanın gerçekleştirilmesinde ise Hidden Markov Toolkit (HTK)'den faydalanılmıştır.

Eğitim ve testlerde iki ayrı ses veritabanı kullanılmıştır. Genel amaçlı hazırlanmış olan TURTEL veritabanı kullanıcı bağımsız testlerde, daha özel amaçlı oluşturulan hava durumu tahmin raporları veritabanı ise kullanıcı bağımlı sistem testlerinde kullanılmıştır.

Konuşmacı bagımsız sistem tanıma testlerinde kelime doğruluk yüzdesi 59-63 olarak hesaplandı. Sistem performansını arttırmak için en uygun karar ağacı budama eşiği seçildi ve bunun sisteme dil modeli ile uygulanmasının ardından yüzde 30-33 arası artış sağlanarak doğruluk yüzdesinde 92-93 arasi değerler elde edildi. Kullanıcı bağımlı olan tek kişilik veritabanında yapılan testlerde doğruluk oranı yüzde 89-93 civarında iken, en uygun karar ağacının ve dil modelinin kullanılmasının doğruluk oranını yüzde 95-97'lere yükselttiği gözlemlendi.

Anahtar Kelimeler : Sürekli Konuşma Tanıma, Dil Modeli, Üçlü Ses, Saklı Markov Modeli, İkili Dil Modeli

To My Family

ACKNOWLEDGEMENTS

I would like to express my appreciation, for their valuable supervision and his strong encouragement throughout the development of this thesis my advisor, Assist. Prof. Dr. Ertuğrul SAATÇİ.

I am most grateful to my family for everything and all my friends in the office and others for their patience, help and being very supporting and kind to me.

Finally, I would like to thank The Scientific and Technical Research Council of Turkey (TÜBİTAK, UEKAE) for support for my thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYSMBOLS – ABBREVATIONS

ASR	Automatic Speech Recognition
CSR	Continuous Speech Recognition
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
MFCC	Mel Frequency Cepstral Coefficient
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
$A = \{a_{ij}\}$	The state transition probability
$B = \{b_j(k)\}$	The observation symbol probability
$\pi = \{\pi_i\}$	The initial state
$\lambda = (A, B, \pi)$	HMM Model
$S = \{S_1, S_2, \dots, S_N\}$	HMM States
$V = \{v_1, v_2, \dots, v_M\}$	Observation symbols
$O = O_1, \dots, O_T$	Observation sequence
$\beta_t(i)$	Backward variable
$\xi_t(i,j)$	Forward and Backward variables
$\alpha_t(i)$	This partial probability
$\delta_1(i)$	The best score (highest probability)

EQUATIONS

INTRODUCTION

Speech Recognition is an active area of research for over sixty years and its popularity grows increasingly over the past years. Speech communication within human beings has played an important role because of its inherent superiority over other modes of human communication. As time passed, the need for better control of complex machines appeared and speech response systems have begun to play a major role in human-machine communication.

As technology created so many other ways of communication to human, nothing can replace or equal speech. In the past, human has been required to interact with machines in the language of those machines as they were not able to train them in the way of understanding human speech. With speech recognition and speech response system, human can communicate with machines using natural language human terminology.

The use of voice processing systems for voices input and output provides a degree of freedom for mobility, alternative modalities for command and data communication and the possibility of substantial reduction in training time to learn to interface with complex system. All those characteristics of speech yield lots of positive advantages over other methods of human-machine interaction, when incorporated into an effective voice control or voice data entry system [1].

In early 1970s commercial application of speech recognition started. Up to that time, as lots of advances have been made in the development of more powerful algorithms for speech analysis and classification. The extraction of word features to permit improved performance in isolated and connected word recognition and in the reduction of the cost of the related hardware has been the major goals of this field.

Most successful speech recognition systems today use the statistical approach. The Hidden Markov Models are used to model speech variations in a statistical and stochastic manner [2]. The idea is collecting statistics about the variations in the signal over a database of samples, and uses these to make a representation of the stochastic process. The speech signal carries acoustic information about the symbolic translation of its meaning. The acoustic information is embodied in HMMs and sound events are modelled by them. Besides the acoustic information, the

sequence of the symbols also constitutes a stochastic process, which can also be modelled. Such statistical approaches to model the symbolic events are called language models.

There are few studies in Turkish speech recognition with respect to the most of the other languages. Especially the large vocabulary recognition field for Turkish is quite empty. The main reason for this is the difficulties caused by the Turkish language. Turkish is an agglutinative language which makes it highly productive in terms of word generation. This is an important problem for traditional speech recognizer which has a fixed vocabulary.

In real life this speech recognition technology might be used to facilitate for people with functional disability or can also be applied to many other areas. This leads to the discussion about intelligent computers and homes where could operate the light switch turn off/on, using computer without any contact to keyboard or operate some domestic appliances.

CHAPTER 1: SPEECH PROCESSING

1 BACKGROUND INFORMATION

1.1 Historical Knowledge on Speech Processing

It has been over 60 years since first works on the speech recognition and so much has been accomplished not only in recognition accuracy but also in performance factor.

For example in isolated word recognition, vocabulary sizes have increased from ten to several hundred words. Highly confusable words have been distinguished, improved adaptation to the individual speaker dependent and speaker independent systems have been developed, the telephone and noisy, distorting channels have been effectively used and effect of other environmental conditions like vibration, g-forces, and emotional stress have also been explored [3].

Historically, start and development of speech recognition researches can be summarized as follows;

The first serious attempt at automatic speech recognition was described by Dreyfus-Graf (in France) in 1950. In the ‘stenosograph’ different sequences of sounds gave different tracks around the screen [3].

In 1952, Davis, Biddulph, and Balashek of Bell Telephone Laboratories developed the first complete speech recognizer.

Several years later, Dudley and Balashek (1958) developed a recognizer called “Audrey” that used ten frequency bands and derived certain spectral features whose durations were compared with stored feature pattern for words in the vocabulary.

A major events in the history of speech recognition occurred in 1972, when the first commercial products appeared that word recognizer (100 Words / Phonological constraints), from Scope Electronics Inc, and from Threshold Technology Inc.

Early history of speech recognition:

1947: Sound Spectrograph

1952: Digits, using word templates, 1 speaker

1958: Digits, using phonetics sequences

1960: Digits, digital computer time normalization

1962: IBM Shoebox Recognizer

1964: Word recognizers for Japanese

1965: Vowels and consonants detected in continuous speech

1967: Voice actuated astronaut manoeuvring unit

1968: 54 Word recognizers, digit string recognizer

1969: Vicens reddy recognizer of continuous speech

1972: 1st Commercial word recognizer

100 Words / Phonological constraints

1974: Dynamic programming (200 Words)

Telephone; Oxygen Mask

1975: Alphabet and Digits, 91 Words

Multiple talker, No training

1976: ARPA Systems; HARPY, HEARSAY, HWIM

182 Talkers, 97% Accuracy, Telephone

1977: CRT Compatible voice terminal

1978: IBM Continuous Speech Recognizer

Spoken language systems technology has made rapid advances in the past decade of eighties, support by progress in the underlying speech and language technologies as well as rapid advances in computing technology.

As a result, there are now several research prototype spoken language systems that support limited interactions in domains such as travel planning, urban exploration, and office management. These systems operate in near real life, accepting spontaneous, continuous speech from speakers with no prior enrolment; they have vocabularies of 1000- 2000 words, and an overall correct understanding rate of almost 90%.

If we look at the speech recognition from Turkish Language perspective, there are only a few studies on Turkish speech recognition. Most of the studies were being done on isolated word recognition, where only a word is recognized at each search step [4] [5] [6] [7] [8].

Recently, large vocabulary continuous speech recognition systems are started to be studied. However, the performance of these systems are poor and not in the level of systems in other languages like English.

Similar or same methods used in English speech recognition generally also can be used in Turkish recognition. The acoustic modelling is almost universal for all languages, since the methods that are used yield similar results [9] [10] [11]. However, the classical language modelling techniques cannot be applied successfully to Turkish.

Turkish is an agglutinative language. Thus, great number of different words can be built from a base by using derivations and inflections [12]. The productive derivational and inflectional structure brings the problem of high growth rate of vocabulary. Furthermore, Turkish is a free word order language. So the language models that are constructed using the co occurrences of words do not perform well.

1.2 Speech Production System

Human communication is to be seen as a comprehensive diagram of the process from speech production to speech perception between the talker and listener

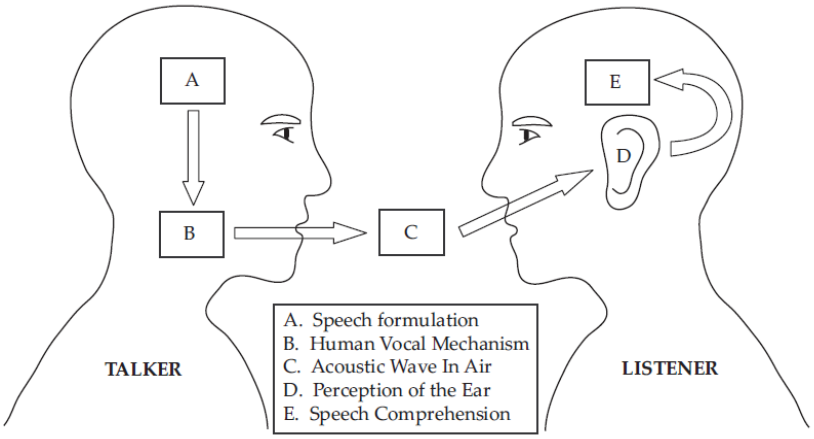

Figure 1 : Schematic diagram of the speech production [13]

The first element Speech formulation (A) is associated with the formulation of the speech signal in the talker's mind. This formulation is used by the Human Vocal Mechanism (B) to produce the actual speech waveform. The waveform is transferred via the air (C) 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 listener percept the waveform (D) and the listener's mind (E) starts processing this waveform to comprehend its content so the listener understands what the talker is trying to tell him or her [13].

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 hearing system during the process of speech signals. The perception process can be seen as the inverse of the speech production process. Worth mentioning is that the production and perception is highly a nonlinear process [13].

To be able to understand how the production of speech is performed one need to know how the human's vocal mechanism is constructed.

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 [14].

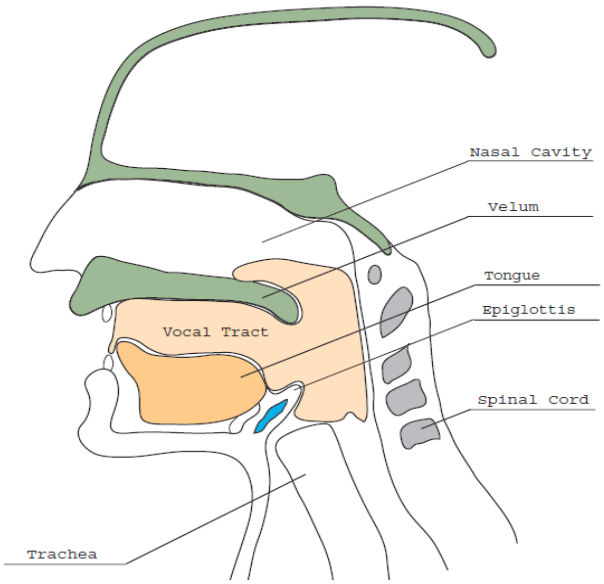

Figure 2 :Human Vocal Mechanism [13]

When human 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.

1.3 Language and Phonetic Features

The speech production begins in the human's mind, when we form a thought that is to be produced and transferred to the listener. After having formed the desired thought, we construct a phase/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.

Phonetics is the study of the sounds of human speech. It is concerned with the actual properties of speech sounds (phones), and their production, audition and perception, while phonology, which emerged from it, studies sound systems and abstract sound units (such as phonemes and distinctive features). Phonetics deals with the sounds themselves rather than the contexts in which they are used in languages.

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 [15], 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.

Language modelling for agglutinative languages needs to be different from modelling language like English. Such languages also have inflections but not to the same degree as an agglutinative language [16].

The Turkish alphabet contains 29 letters (45 phonetic symbols) and is a phoneme based language which means phonemes are represented by letters in the written language. There are 8 vowels and 21 consonants;

Vowels: a e ı i o ö u ü

Consonants : b c ç d f g ğ h j k l m n o ö p r s ş t v y z

There is nearly one to one mapping between written text and its pronunciation but some vowels and consonants have variants depending on the place they are produced in the vocal tract.

1.4 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 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.

The speech representation can exist in either the time or frequency domain, and in three different ways [14]. These are three-state representation, spectral representation and parameterization of the spectral activity.

1.4.1 Three-State Representation

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

Silence (S) – No speech is produced

Unvoiced (U) – Vocal cords are not vibrating, resulting in a periodic 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 short time period (5-100 ms) during which it is stationary. Stationary signals can be define constant in their statistical parameters over time.

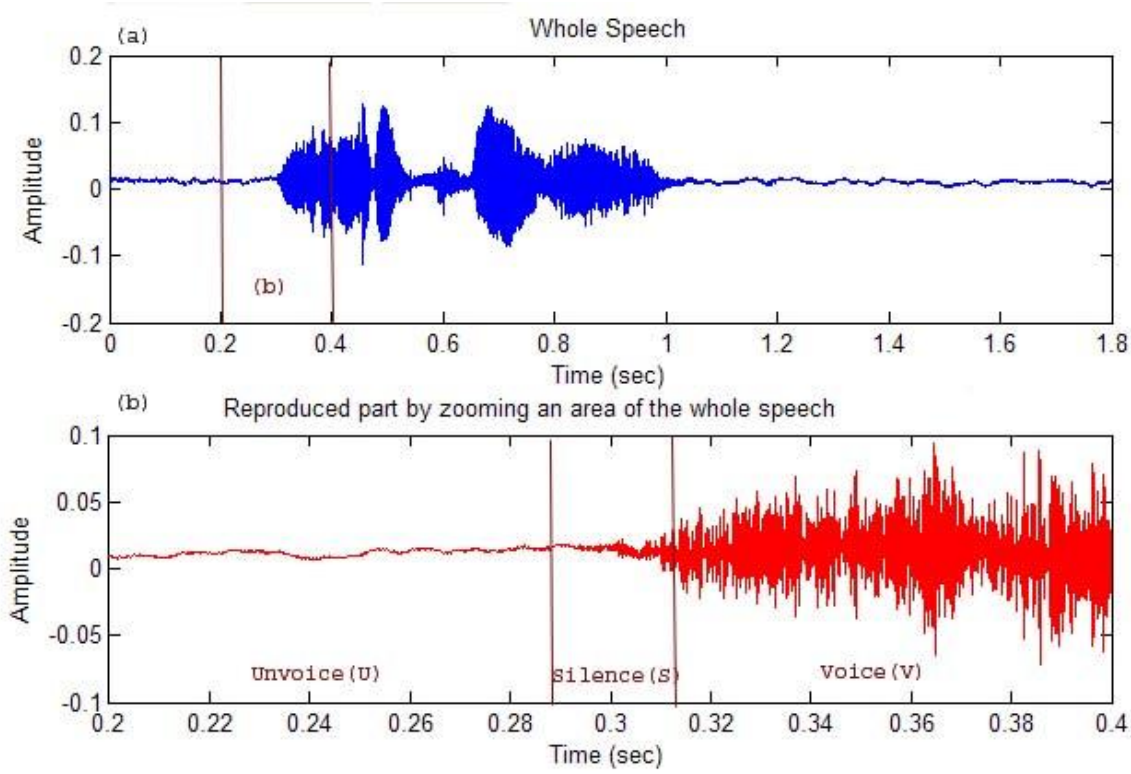

Figure 3 : Three state representation

The upper plot (a) contains the whole speech sequence and in the plot (b) a part of the upper plot (a) is reproduced by zooming an area of the whole speech sequence. At figure the segmentation into a three state representation, in relation to the different parts of the plot(b), is given.

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

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

Mixed: Simultaneously voiced and unvoiced

Plosive: Start with a short region of silence then voiced or unvoiced speech, or both build up air pressure behind the closure, then suddenly release it.

Whisper: No excitation. Air moved through vocal tract.

1.4.2 Spectral Representation

Spectral representation of the speech intensity over time is very popular and the most popular one is the sound spectrogram.

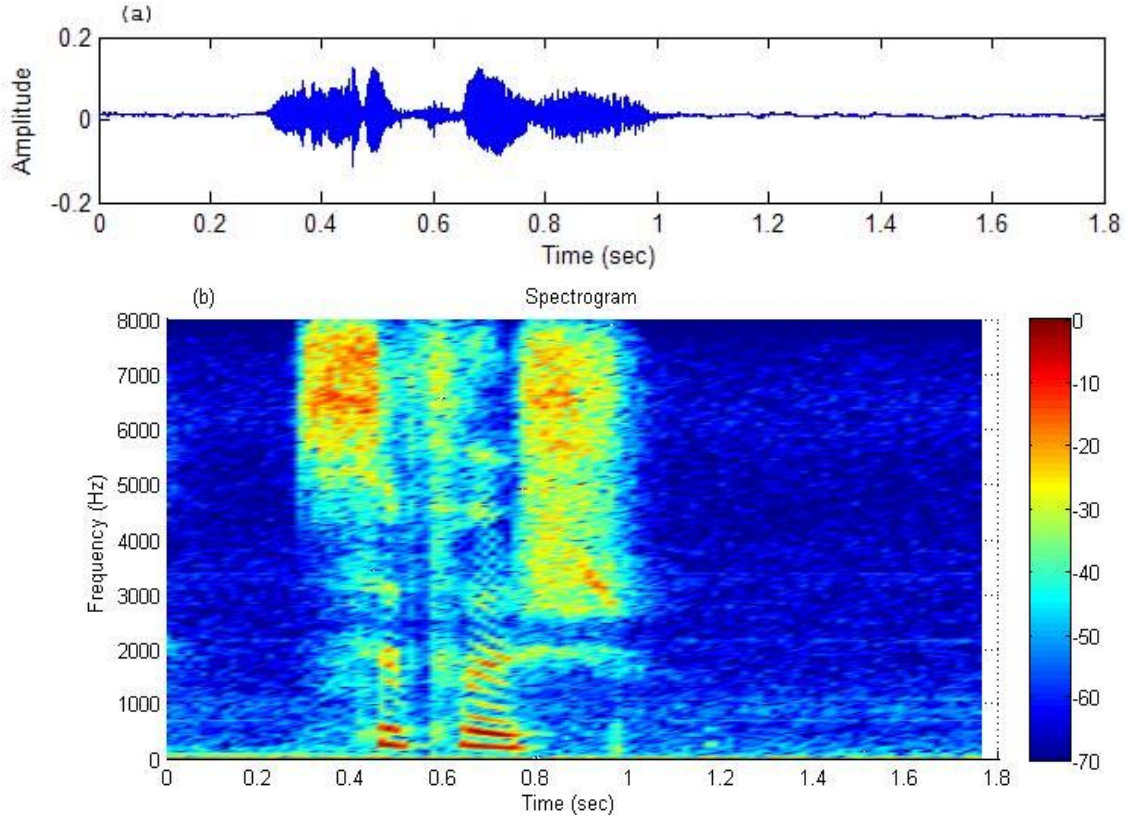

Figure 4 : Speech amplitude(a) and Spectrogram(b)

Here the darkest (dark blue) parts represents the parts of the speech waveform where no speech is produced and the lighter (red) parts represents intensity if speech is produced [17].

Figure (a) the speech waveform is given in the time domain and figure (b) shows a spectrogram in the time-frequency domain.

1.4.3 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 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.

1.5 Speech Recognition System

Recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands and control, data entry, and document preparation.

Speech recognition systems can be characterized by many parameters, some of the more important of which are speaking mode-style, vocabulary, enrolment etc. An isolated-word speech recognition system requires that the speaker pause briefly between words, connected word speech recognition similar to Isolated words, but with a minimal pause whereas a continuous speech recognition system does not. Spontaneous, or extemporaneously generated, speech contains disfluencies, and is much more difficult to recognize than speech read from script. Some systems require speaker enrolment a user must provide samples of his or her speech before using them, whereas other systems are said to be speaker-independent, in that no enrollment is necessary. Some of the other parameters depend on the specific task. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words.

1.5.1 Categories of Speech Recognition

1.5.1.1 Recognition of Isolated Words

An isolated word recognition system operates on single words at a time, requiring a distinct pause between saying each word [18]. It requires about a half second (500ms) or greater pause be inserted between spoken words [19]. This is the simplest form of recognition to perform because the end points are easier to find and the pronunciation of a word tends not to affect others. Thus, because the occurrences of words are more consistent, they are easier to recognize. The technique of the system is to compare the incoming speech signal with an internal

representation of the acoustic pattern of each word in a relatively small vocabulary and to select the best match, using some distance metric [18]. Usually the recognition success is about 100% and it is adequate for many applications but is far from being a natural way of communicating [19].

1.5.1.2 Recognition of Connected Speech

Connected word recognition is one of the most interesting problems at the present stage of speech recognition research since isolated word recognition techniques have been improved up to a practically high level [20]. Connect word systems (or more correctly 'connected utterances') are similar to Isolated words, but allow separate utterances to be 'run-together' with a minimal pause between them [21].

1.5.1.3 Recognition of Continuous Speech

A continuous speech system operates on speech in which words are connected together, i.e. not separated by pause. Continuous speech is more difficult to handle because of variety of effects [18]. First problem is to finding the utterance boundaries for allow users to speak almost naturally, while the computer determines the content. Another problem is co-articulation. The production of each phoneme is affected by production of surrounding phonemes, and similarly that the start and end of words are affected by preceding and following words. The recognition of continuous speech is also affected by the rate of speech [18]. Basically, these types of recognitions are named as a computer dictation.

CHAPTER 2: SPEECH TO TEXT SYSTEM

Speech recognition, also known as automated speech recognition (ASR) or speech-to-text (STT) is a process which converts the acoustic speech signal into written text. A typical example of speech to text application is in dictation systems. Speech to text systems may also be used as an intermediate step for making spoken language accessible to machine translation. Systems generally perform two different types of recognition: single-word and continuous speech recognition. Continuous speech is more difficult to handle because of a variety of effects such as speech rate, co-articulation, etc. Co-articulation express to changes in the articulation of a speech segment.

Speech to text systems can be separated into two categories which are speaker dependent and speaker independent. A speaker dependent system is developed to operate for a single speaker. These systems are usually easier to develop more accurate, but not as flexible as speaker independent systems. A speaker independent system is developed to recognize speech regardless of speakers, i.e. it does not need to be trained to recognize individual speakers. These systems are the most difficult to develop and their accuracy are lower than the speaker dependent systems. However, they are more flexible.

In this chapter we will explain general speech to text system frame and its modules. A general speech to text system involves several steps that can be categorized as follows: Data Preparation, Pre-Processing, Feature Extraction, Training and Recognition. Additional to this language model can be adapted to increase the performance of the system. Block Diagram of a speech to text system can be given as shown:

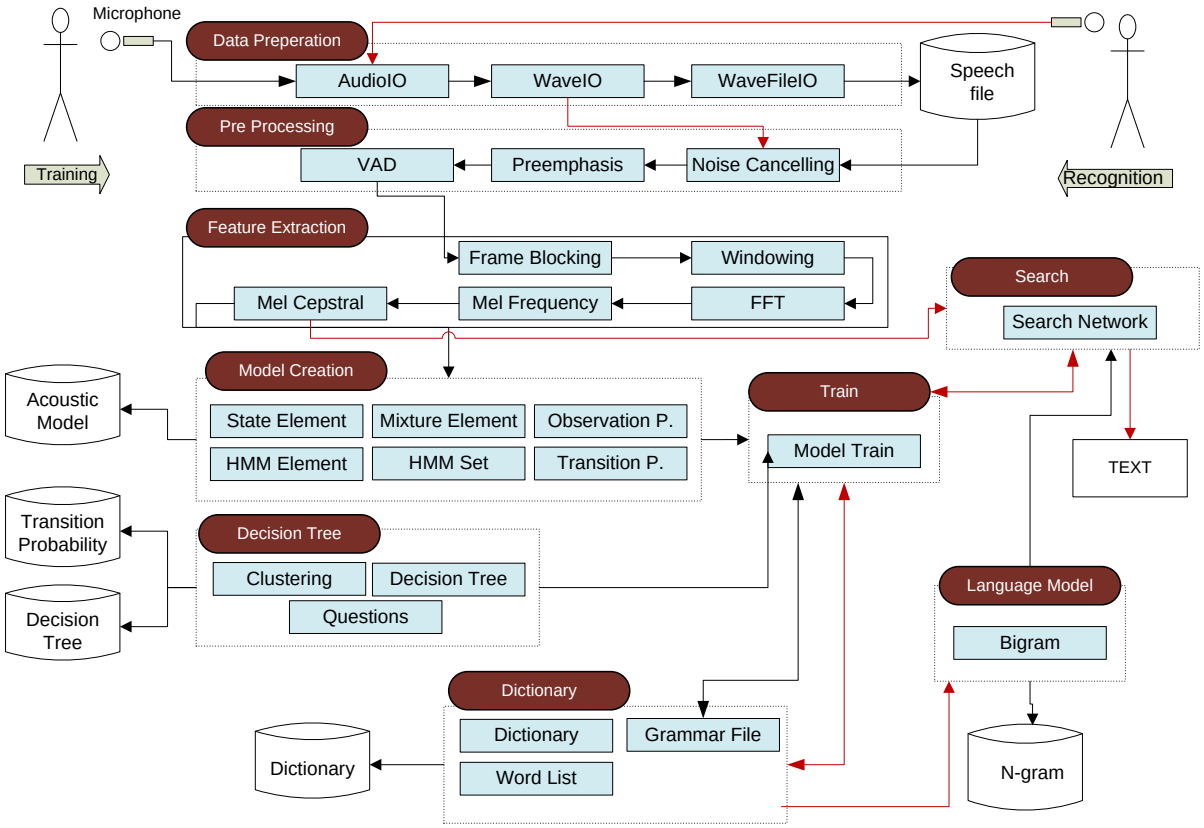

Figure 5 : Speech Recognition Process

2.6 Preprocessing

Preprocessing is a process of preparing speech signal for further processing or in the other words to enhance the speech signal. The objective in the pre-processing is to modify the speech signal in such a way that will be more suitable for the feature extraction.

Preprocessing involves three steps: noise cancelling, preemphasis and voice activation detection.

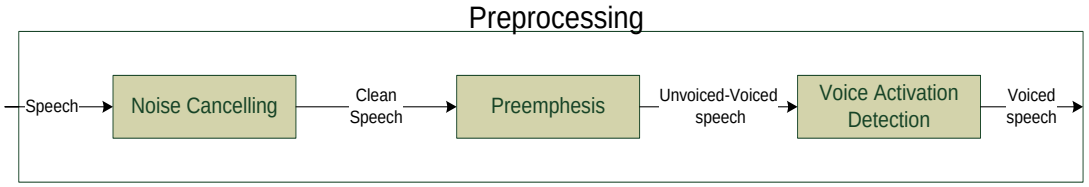

Figure 6 : Preprocessing steps

Noise Cancelling: Noise-cancelling reduces unwanted ambient sounds by a low pass filter which is a filter that passes low-frequency signals. Human can produce sounds between 100-4500Hz and for speech production the range is smaller. Thus, if telephone records (at 8000 Hz) are used, less than 120 Hz and above 3400 Hz can be a logical cut-off because over the 3400 Hz human speech has no information.

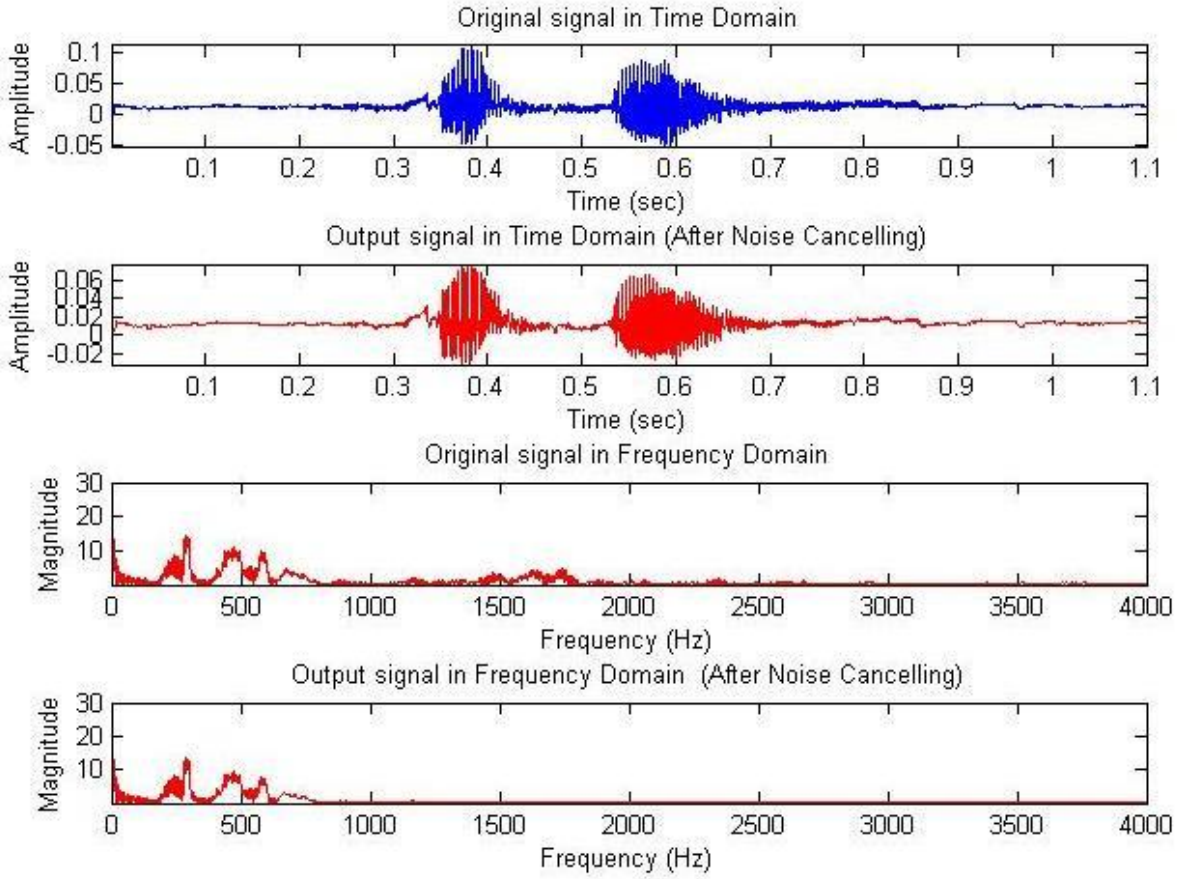

Figure 7 : Effect of the noise cancelling (word record is sIflr (zero))

Preemphasis: The first stage in MFCC feature extraction is to boost the amount of energy in the high frequencies. It turn s out that if we look at the spectrum for voiced segment like vowels, there is more energy at the lower frequencies than at the higher frequencies. This drop in energy across frequencies (which is called spectral tilt) is caused by the nature of the glottal pulse. Boosting the high frequency energy makes information from these higher formants more available to the acoustic model and improves phone detection accuracy [22].

This is usually done by a highpass filter. The most commonly used filter type for this step is the FIR filter. The filter characteristic is given by the equation (2.1):

$$H(z) = 1 - a\,z^{-1} \qquad (2.1)$$

The value (a) is chosen to be approximately 0.95. There are two reasons behind this [23]. The first is that to introduce a zero near z=1, so that the spectral contributions of the larynx and the lips have been effectively eliminated.

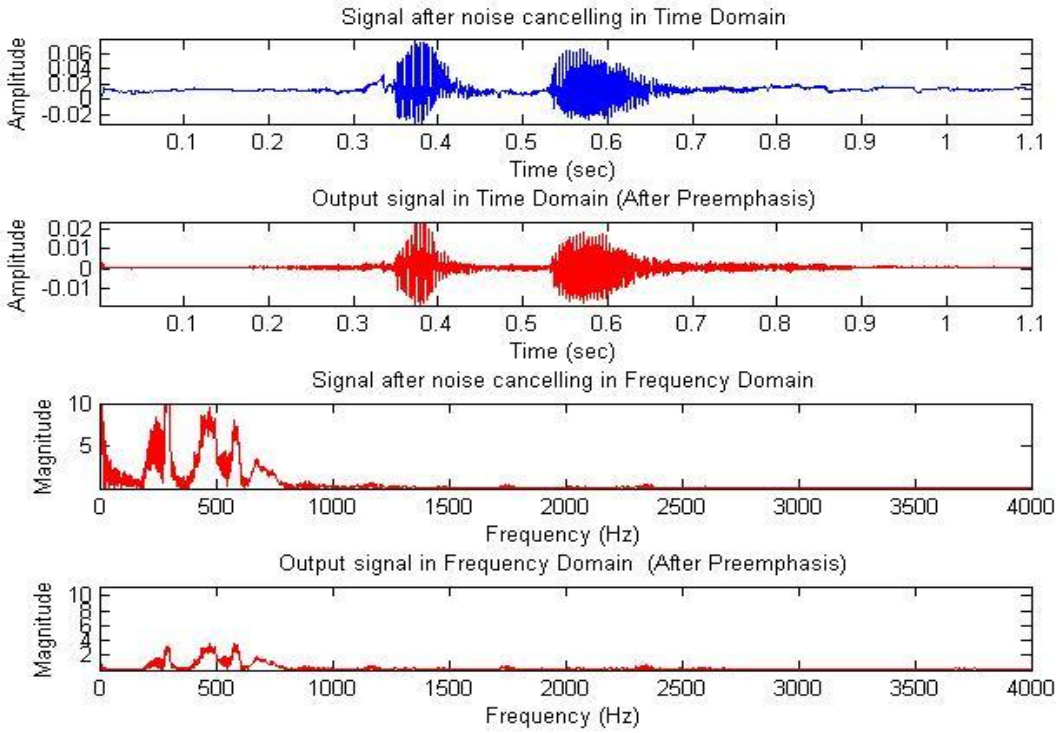

Figure 8 : The effect of Preemphasis Filter in Time and Frequency Domain (word record is slflr)

By this way, the analysis can be asserted to be seeking parameters corresponding to the vocal tract only. The second reason, if the speech signal is dominated by low frequencies, preemphasis is used to prevent numerical instability [24].

Voice Activation Detection: An important problem in speech processing is to detect the presence of speech in a background noise. This problem is often referred to as the end point location problem [25] [16].

The accurate detection of a word start and end points means that subsequent processing of the data can be kept to a minimum. In order to decide start and end points of a speech signal, the energies of each block is calculated first using sum of square energy algorithm.

$$E = \sum_{i=1}^{N} x(i)^2 \qquad (2.2)$$

Assuming that there is no speech in the first few frames, of recording, the average of the first few frames, give the maximum energy and the minimum energy which are used calculates for cutting down unvoiced parts of speech.

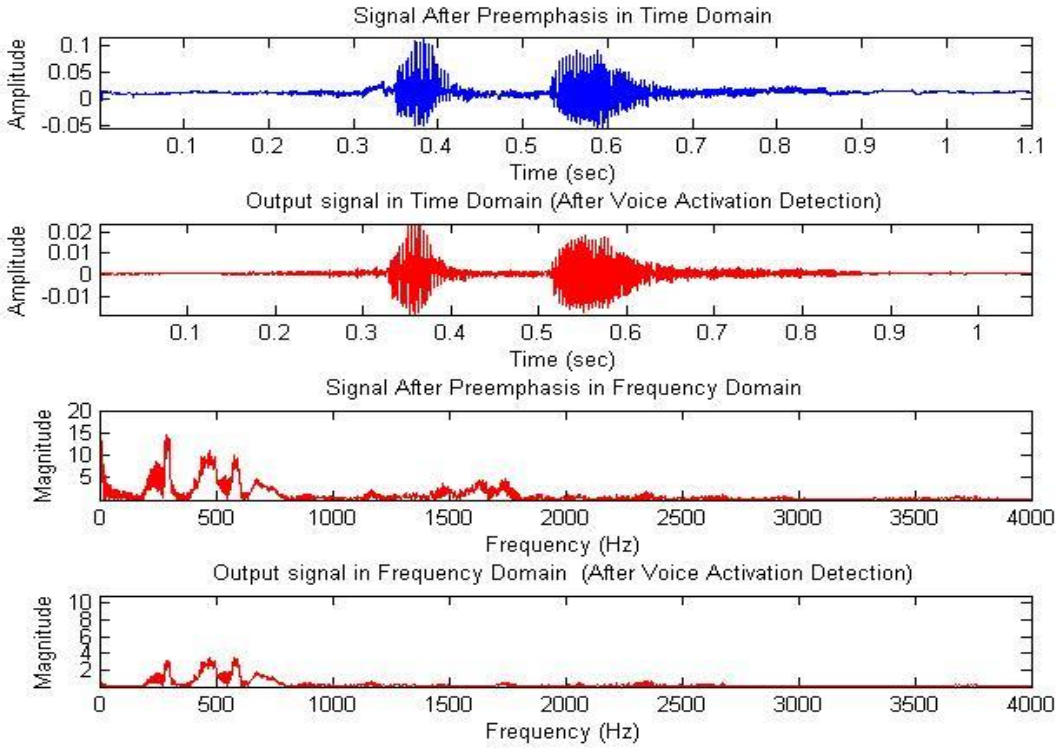

Figure 9 : The preemphasized signal cut down by the VAD.

2.7 Feature Extraction

The purpose of feature extraction is to convert the speech waveform to some type of parametric representation (at a considerably lower information rate). The speech signal is a slowly time varying signal (it is called quasi-stationary) [26]. When examined over a sufficiently short period of time (typically between 20 and 100 ms), its characteristics are fairly stationary. However, over long periods of time (on the order of 0.2s or more) the signal characteristics change to reflect the different speech sounds being spoken.

Therefore, short-time spectral analysis is the most common way to characterize the speech signal. A wide range of possibilities exist for parametrically representing the speech signal for the recognition task, such as Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), and others. MFCC is perhaps the best known and most popular, and this feature has been used in this work.

MFCC's are based on the known variation of the human ear's critical bandwidths with frequency. Linear prediction coefficients are mainly focused to model vocal tract while extracting feature coefficients from the speech, but MFCC's apply Mel scale to power spectrum of speech in order to imitate human hearing mechanism.

The MFCC technique makes use of two types of filter, namely, linearly spaced filters and logarithmically spaced filters. To capture the phonetically important characteristics of speech, signal is expressed in the Mel frequency scale. This scale has a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. Normal speech waveform may vary from time to time depending on the physical condition of speakers' vocal cord. Rather than the speech waveforms themselves, MFFCs are less susceptible to the said variations [26] [34].

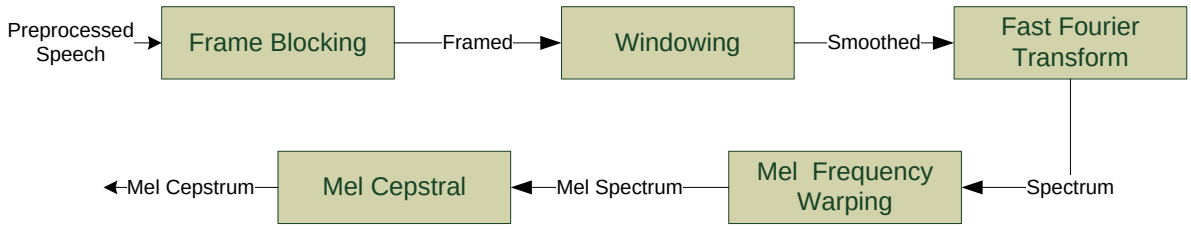

Figure 10 : Feature Extraction Steps

In a typical Feature Extraction process, the first step is windowing the speech to divide the speech into frames. Since high frequency formants have smaller amplitude than low frequency formants, high frequencies may be emphasized to obtain similar amplitude for all formants [4]. After windowing, magnitude squared FFT is used to find the power spectrum of each frame. Here we perform filter bank processing to the power spectrum, which uses mel scale. Discrete cosine transformation is applied after converting the power spectrum to log domain in order to compute MFCC.

2.7.1 Frame Blocking

Speech signals change in every few millisecond, because of this speech should be analyzed in small duration frames. Spectral evaluation of signals is reliable in case the signal is assumed to be stationary. Speech characteristics do not change much in a short time period because of this, speech signal is blocked into frames of N samples (generally between 20 and 100 millisecond), with adjacent frames being separated by M (M<N). The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N-M samples. This process continues until all the speech is accounted for within one or more frames. Typical values for N and M are N=256 (which is equivalent to ~30 ms) and M=100.

2.7.2 Windowing

Choosing a window type is an important factor to process signal correctly. Two competing factors exist in this choice.

One of them is smoothing the discontinuity at the window boundaries and other factor is not to disturb the selected points of the waveform. Typical window types are Rectangular, Hamming and Hanning.

Indeed the simplest window is the rectangular window. The Rectangular window can cause problems, because it abruptly cuts of the signal at its boundaries. These discontinuities create problems when we do Fourier analysis. For this reason, a more common window used in MFCC extraction is the Hamming window, which shrinks the values of the signal toward zero at the window boundaries, avoiding discontinuities.

Windowing is also beneficial to eliminate possible gaps between frames. Without windowing, the spectral envelope has sharp peaks and the harmonic nature of a vowel is not apparent.

Definitions Rectangular Window (2.3) and the Hamming Window (2.4); are given as follows;

$$w(n) = 1 \qquad (2.3)$$

$$w(n) = 0.53836 - 0.46164\ \cos\left(\frac{2\pi\ n}{N-1}\right),\ \ 0 \le n \le N-1 \qquad (2.4)$$

Where:

N is the number of samples in each frame (frame size)

n is the index of the frames

The length of the window determines the frequency resolution of the signal. Increasing resolution is equal to using longer window, but in this case we may violate the assumption of stationarity for signals. Typically 10-25 ms windows are used in the processing.

2.7.3 Fast Fourier Transform

The next step is to extract spectral information for our windowed signal; we need to know how much energy the signal contains at different frequency bands.

In here Fast Fourier Transform(FFT) is used to convert speech frame to its frequency domain representation, the short term power spectrum is found. The FFT is a faster version of the Discrete Fourier Transform (DFT). The FFT utilizes some clever algorithms to do the same thing as the DFT, but in much less time. The Fourier series and DFT are defined by the formulas:

$$\mathrm{f(t)} = \frac{\mathrm{a_0}}{2} + \sum_{\mathrm{n=1}}^{\infty} (\mathrm{a_n}\cos(\mathrm{w_n\ t}) + \mathrm{b_n}\ \sin(\mathrm{w_n\ t})) \qquad (2.5)$$

$$a_n = \frac{2}{T}\ \int_{-T/2}^{T/2} f(t)\cos(\mathrm{w_n\ t})\mathrm{dt} \qquad b_n = \frac{2}{T}\ \int_{-T/2}^{T/2} f(t)\sin(\mathrm{w_n\ t})\mathrm{dt}$$

$$X_k = \sum_{n=0}^{N-1} x_n\ e^{-i2\pi k \frac{n}{N}} \qquad \mathrm{k=0,...,N\text{-}1} \qquad (2.6)$$

Evaluating this definition directly requires O(N^2) operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations. More precisely, all known FFT algorithms require Θ(N log N) operations.

2.7.4 Mel Frequency Cepstrum Coefficient

The most widely used feature vector in automatic recognition is so far known as mel frequency coefficients. These coefficients apply mel scale to power spectrum of speech in order to imitate human hearing mechanism.

Mel-Frequency Cepstral Coefficients (MFCC) are used in speech recognition because they provide a decorrelated, perceptually-oriented observation vector in the cepstral domain.

2.7.5 Mel Filter Bank

Magnitude squared FFT will be an information about the amount of energy at each frequency band. Human hearing however is not equally sensitive at all frequency bands. It is less sensitive at higher frequencies, roughly above 1000 Hertz.

In general the human response to signal level is logarithmic; humans are less sensitive to slight differences in amplitude at high amplitudes than at low amplitudes. (In other words logarithm compresses the dynamic range of values, which is a characteristic of human hearing system [27].)

A mel is a unit of pitch defined so that pairs of sounds which are perceptually equidistant in pitch are separated by an equal number of mels. The mapping between frequency in Hertz and the mel scale is linear below 1000 Hz and logarithmic above 1000 Hz.

So a perceptual filter bank (a Mel-scale filter bank Eq.(2.7) and Figure below) is used to approximate the human ear's response to speech (in an attempt to receive only relevant information). Due to the overlapping filters, data in each band highly correlated. Filters are used to emphasize some of the frequency contents in power spectrum of the speech like ear does. More filter in the bank process the spectrum below 1 kHz since the speech signal contains most of its useful information such as first formant in lower frequencies.

Each filter output is the sum of its filtered spectral components [27].The central frequency of each Mel filter in the bank uniformly spaces below 1 kHz and it follows a logarithmic scale above 1 kHz. On the other hand these triangular filters are equally spaced along the Mel scale which is approximated by:

$$f_{mel} = 2595 \log_{10}(1 + \frac{F_{Hz}}{700}) \quad (2.7)$$

$f_{mel} : Mel\ Frequency$ (Mel) $f_{Hz} : Frequency\ (Hz)$

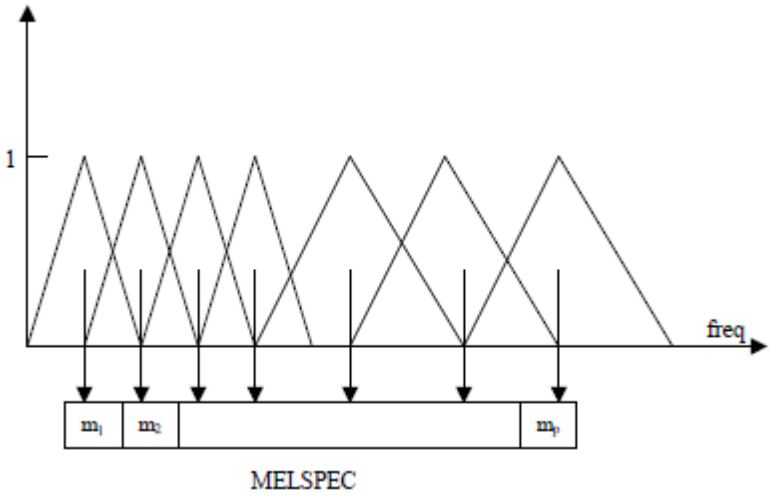

Figure 11 : Mel scale Filter Bank

The spacing between the filters in mel scale is computed using:

$$\Delta\emptyset = (\emptyset_{max} - \emptyset_{min})/(M+1)$$

$\emptyset_{max}$: The maximum frequency value in the filterbank in mels

$\emptyset_{min}$: The minimum frequency value in filterbank in mels

M: Number of desired filters

Center frequencies of the filters are found using:

$$\emptyset_c(m) = m.\Delta\emptyset\,, \qquad m = 1,2,\ldots,M$$

After converting these center frequencies to Hertz, the filter are formed using the formulae below:

$$H(k,m) = \begin{cases} 0 & for \quad f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} & for \quad f_c(m-1) \le f(k) < f_c(m) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)} & for \quad f_c(m) \le f(k) < f_c(m+1) \\ 0 & for \quad f(k) \ge f_c(m+1) \end{cases}$$

$k = 0,1,\ldots,N-1$ N, window length

$$f(k) = \frac{k.f_s}{N}$$

Then these filters applied to the magnitude spectrum and logarithm is taken:

$$X'(m) = ln\left(\sum_{k=0}^{N-1} |X(k)| \,.\, H(k,m)\right)$$

2.7.6 Discrete Cosine Transformation

After filtering, discrete cosine transform (DCT) is applied to the resulting log filter-bank coefficients to compress the spectral information into lower order ones, and also to de-correlate them. DCT is formulates as:

$$c_i = \sqrt{\frac{2}{N}} \quad \sum_{j=1}^{N} \cos\left\{m_j \ \left(\frac{i\pi}{N}\ (j-0.5)\right)\right\} \quad i = 1, \dots, M \qquad (2.8)$$

Where N is the number of filter-bank channels, m_j is the output of the j'th filter and M is the number of DCT coefficients.

For speech recognition purposes, generally the first 12 coefficients are enough for getting information about the vocal tract filter which is cleanly separated from the information about the glottal source [22].

2.8 Training and Recognition

In Training and Recognition part, different methods have been tried by different researchers so far in order to find a better method to be used in speech processing applications [28] [23]. Some of these methods are seen to work better and are used frequently we can divide these methods into two classes:

First class, deterministic models that exploit some known specific properties of the signal, e.g., that the signal is a sine wave which has parameters like amplitude, frequency [28].

Second class of signal models is the set of statistical models in which one tries to characterize only statistical properties of the signal such as Gaussian processes, Poisson processes, Markov processes and Hidden Markov processes, among others.

For speech processing, both deterministic and stochastic signal models have had good success. There are advanced techniques in speech recognition such as Dynamic Time Warping (DTW), the Hidden Markov Modelling (HMM) and Artificial

Neural Network (ANN) techniques. The DTW is widely used in the small-scale embedded-speech recognition systems such as cell phones.

2.8.1 Dynamic Time Warping

Dynamic time warping is fundamentally a feature matching algorithm between a set of reference and test features [4]. The basic idea behind it is the time alignment of two sets. During DTW, it is tried to compress or expand the test feature set in a non-linear way to match it to a reference feature set. This is a critical task in template matching because the way of saying an utterance changes depending on the condition of speaker. The distance between the test and reference set is a measure of the goodness of the matching.

The disadvantage of this approach is the computational limitations. The DTW requires a template to be available for any utterance to be recognized and it is not used for complex task involving large vocabularies. The method can be effectively used in small scale applications.

2.8.2 Artificial Neural Networks

The application of artificial neural networks [23] to speech recognition is the newest and least well understood of the recognition technologies. The ANN approach is basically an alternative computing structure for carrying out the necessary mathematical operations. The ANN strategy can also enhance the distance or likelihood computing task by learning which features are most effective [23].

2.8.3 Hidden Markov Model

Nowadays a stochastic based method, Hidden Markov Modelling (HMM), is frequently used in large scale speech recognition systems. In this thesis one type of stochastic signal model, namely the Hidden Markov Model is concerned.

HMM is a finite-state machine with a given number of states; passing from one state to another is made instantaneously at equally spaced time moments. At every pass from one state to another the system generates observations. Two process taking place: the transparent one and the hidden one, which cannot be observed, first represented by the observations string and the second, represented by the states string [9]. Each state aims to characterise a distinct, stationary part of the acoustic signal. Phone or word models are then formed by concatenating the appropriate set of states. The hidden part of the HMM is the states process, which is governed by a Markov model.

We model such processes using a hidden Markov model where there is an underlying hidden Markov process changing over time, and a set of observable states which are related somehow to the hidden states [29].

An HMM is characterized by the following parameters and these are formally defined as follows:

N, the number of states in the model. For example, the HMM model for a phonetic recognition system will have at least three states which correspond to initial state, the steady state and the ending state [30]. The states will be denoted as

$S = \{S_1, S_2, \dots, S_N\}$, and the state at time t as q_t.

M, the number of distinct observation symbols per state. In speech recognition systems, states are considered as the feature vector sequence of the model. The individual observation symbols will be denoted as $V = \{v_1, v_2, \dots, v_M\}$

$A = \{a_{ij}\}$, the state transition probability distribution for each state.

$$a_{ij} = P\left[q_{t+1} = S_j \middle| q_t = S_i\right], \quad 1 \leq i, j \leq N \tag{2.9}$$

$B = \{b_j(k)\}$, the observation symbol probability distribution in state j.

$$b_j(k) = P\left[v_k \; at \; t \middle| q_t = S_j\right], \quad 1 \leq j \leq N \quad 1 \leq k \leq M \tag{2.10}$$

$\pi = \{\pi_i\}$, the initial state distribution.

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N \tag{2.11}$$

Having defined the necessary five parameters that are required for the HMM model (λ):

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

Speech is produced by the slow movements of the articulator organ that are the tongue, the upper lip, the lower lip, the upper teeth, the upper gum ridge, the hard palate, the velum, the uvula, the pharyngeal wall and the glottis. The speech articulators taking up a sequence of different positions and consequently producing the stream of sounds that form the speech signal. Each articulator position could be represented by a state of different and varying duration. Accordingly, the transition between different articulator positions (states) can be represented by $\mathrm{A} = \{\mathrm{a_{ij}}\}$.

The observations in this case are the sounds produced in each position and due to the variations in the evolution of each sound this can be also represented by $B = \{b_j(k)\}$ [31].

In the training mode, the task of the HMM based speech recognition system is to find a HMM model that can best describe an utterance (observation sequences) with parameters that are associated with the state model A and B matrix. A measure in the form of maximum likelihood is computed among the test utterance and the HMM, and the one with the highest probability value against the test utterance is considered as the candidate of the matched pattern [30].

Figure bellow a simple example of a five state left to right HMM. The HMM states and transitions are represented by node arrows, respectively.

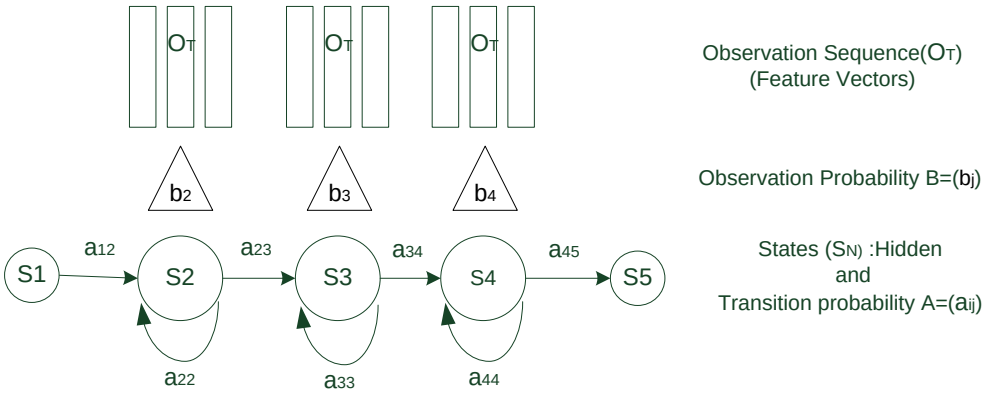

Figure 12 : A five state left to right HMM

Basic Problems of HMM: Once a system can be described as a HMM, three problems can be solved.

Problem 1 (Evaluation): Given the observation sequence $O = O_1, \dots, O_T$ and an HMM model, how do we compute the probability of O given the model?

The probability of an observation sequence is the sum of the probabilities of all possible state sequences in the HMM. Native computation is very expensive. Given T observations and N states, there are NT possible state sequences. Even small HMMs, e.g. T=10 and N=10, contain 10 billion different paths. Solution to this and problem 2 is to use dynamic programming. Solution is Forward-Backward Algorithm.

Problem 2 (Decoding): Given the observation sequence $O = O_1, \dots, O_T$ and an HMM model, how do we find the state sequence that best explains the observations?

The solution to Problem 1 (Evaluation) gives us the sum of all paths through an HMM efficiently. For Problem 2, we want to find the path with the highest probability. Solution is the Viterbi Algorithm.

Problem 3 (Learning): How do we adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $\mathrm{P}(\mathrm{O}|\lambda)$?

Up to now we have assumed that we know the underlying model. Often these parameters are estimated on annotated training data, which has two drawbacks:

1. Annotation is difficult and/or expensive
2. Training data is different from the current data

We want to maximize the parameters with respect to the current data, i.e., we're looking for a model, such that the model best describes the observation sequence. Solution is Baum-Welch Algorithm.

Solutions to Basic Problems of HMM:

2.8.3.1 The Forward-Backward Algorithm

If we could solve the evaluation problem, we would have a way of evaluating how well a given HMM matches a given observation sequence. Therefore, we could use HMM to do pattern recognition, since the likelihood $P(O|\lambda)$ can be used to compute posterior probability $P(\lambda|O)$, and the HMM with highest posterior probability can be determined as the desired pattern for the observation sequence [32].

The Problem 1 can be solved using the forward backward algorithm. The forward backward algorithm is an algorithm for computing the probability of a particular observation sequence [33].

Forward Algorithm:

The partial probability of state i at time t as $\alpha_t(i)$ - this partial probability calculates as;

$$\alpha_t(\mathrm{i}) = \mathrm{P}(\mathrm{O}_1\mathrm{O}_2 \ldots \mathrm{O}_t, \mathrm{q}_t = \mathrm{S}_\mathrm{i}|\lambda) \qquad (2.12)$$

We can solve the problem for $\alpha_t(\mathrm{i})$ inductively, using the following steps:

Initialization;

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

Induction;

$$\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_{i=1}^{n} \alpha_t(i) a_{ij}, \quad 1 \le t \le T-1, \quad 1 \le j \le N$$

Termination;

$$P(O/\lambda) = \sum_{i=1}^{N} \propto_T (i)$$

Backward Algorithm:

The partial probability of state i at time t as $\beta_t(i)$ - this partial probability calculates as;

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda) \qquad (2.13)$$

i.e., the probability of the partial observation sequence from $t+1$ to the end, given state S_i at time t and model λ. Again we can solve for $\beta_t(i)$ inductively, as follows:

Initialization;

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

Induction;

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad t = T-1, T-2, \dots, 1, \ 1 \le i \le N.$$

The backward and the forward calculations can be used extensively to help solve fundamental problems 2 and 3 of HMMs.

2.8.3.2 Viterbi Algorithm

If we could solve the decoding problem, we could find the best matching state sequence given an observation sequence, or, in other words, we could uncover the hidden state sequence [32]. The viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states called the Viterbi path that results in a sequence of observed events [34].To find the single best state sequence, Q= {$q_1\ q_2 \dots\ q_T$}, for the given observation sequence O= {$O_1\ O_2 \dots\ O_T$}, we need to define the quantity [35]

$$\delta_1(i) = \max_{q_1, q_1 \dots q_{t-1}} p[q_1\ q_2 \dots\ q_t = i, O_1\ O_2 \dots\ O_T | \lambda]$$

i.e., $\delta_1(i)$ is the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends in state S_i. By induction [28]:

$$\delta_t(j) = \left[\max_i \ \delta_t\ (i)a_{ij} \right] b_j(O_{t+1})$$

To actually retrieve the state sequence, we need to keep track of the argument which maximized (above equation), for each t and j. We do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can now be stated as follows:

1) Initialization;

   $\delta_1(i) = \pi_i\ b_i(O_1) \quad 1 \leq i \leq N$

2) Recursion;

   $\delta_t(j) = \left[\max_{1 \leq j \leq N} \delta_{t-1}\ (i)a_{ij} \right] b_j(O_t)$

   $\delta_t(j) = \left[\operatorname{argmax}_{1 \leq j \leq N} \delta_{t-1}\ (i)a_{ij} \right] \quad 2 \leq t \leq T, 1 \leq j \leq N$

3) Termination:

   $p^* = \max_{1 \leq j \leq N} \delta_T\ (i)$

   $q_T^* = \arg\max_{1 \leq i \leq N} \left[\delta_T\ (i) \right]$

4) Path (state sequence) backtracking:

   $q_t^* = \ \psi_{t+1}\ (q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1$

2.8.3.3 Baum – Welch Algorithm

If we could solve the learning problem, we would have the means to automatically estimate the model parameters from an ensemble of training data [32].

The third, and by far the most difficult, problem of HMMs is to determine a method to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model which maximizes the probability of the observation sequence. In fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. We can, however, choose

$\lambda = (A, B, \pi)$ such that p(O| λ) is locally maximized using an iterative procedure such as the Baum Welch method [28].

Firstly $\xi_t(i,j)$ defined, the probability of being in state S_i at time t and state S_j at time $t+1$,

$$\xi_t(i,j) = P\big(q_t = S_i, q_{t+1} = S_j \,\big|\, O, \lambda\big) \tag{2.14}$$

The following figure illustrates the sequence of events leading to the conditions required by (2.14).

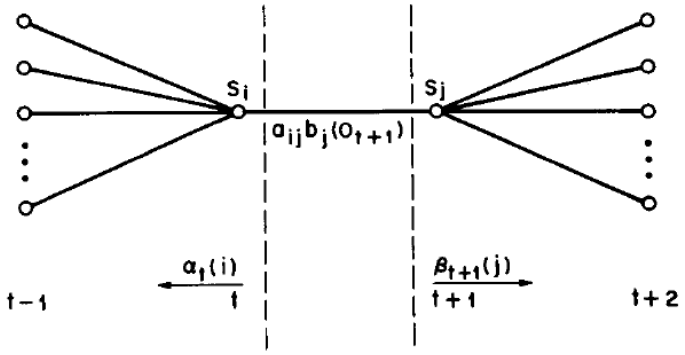

Figure 13 : Illustration of the sequence of operations required for the computation of the join event that the system is in state S_i at time t and state S_i at time t +1 [28]

Definitions of the forward and backward variables, we can write $\xi_t(i,j)$ in the form:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O \mid \lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\,a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}$$

Where the numerator term is just $P\big(q_t = S_i, q_{t+1} = S_j, O \,\big|\, \lambda\big)$ and the division by $P(O \mid \lambda)$ gives the desired probability measure.

The probability of being in state S_i at time t, given the observation sequence and the model as:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda)$$

Since $\alpha_t(i)$ accounts for the partial observations sequence $O_1 O_2 \dots O_t$ and state S_i at time t, while $\beta_t(i)$ accounts for the remainder of the observation sequence $O_{t+1} O_{t+2} \dots O_T$, given state S_i at t, its relation with forward and backward variables can be given as [28]:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

If we sum $\gamma_t(i)$ over t, we get a quantity which can be interpreted as the expected number of times that state S_i is visited and similarly the summation of $\xi_t(i,j)$ over t (from $t = 1$ to $t = T - 1$) can be interpreted as the expected number of transitions from state S_i to state S_j.

With the help of these, the formulas for re-estimating the parameters A, B and π are given as [28];

$$\overline{\pi}_i = \quad expected\ number\ of\ times\ in\ state\ S_i\ at\ time\ (t = 1) = \gamma_1(i)$$

$$\overline{a}_{ij} = \quad \frac{expected\ number\ of\ transitions\ from\ S_i\ to\ S_j}{expected\ number\ of\ transitions\ from\ S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\overline{b}_j(k) = \frac{expected\ number\ of\ times\ in\ state\ j\ and\ observing\ symbol\ V_k}{expected\ number\ of\ times\ in\ state}$$

$$= \frac{\sum_{t=1}^{T}{}_{O_{t=V_k}} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

So after re-estimation of the model parameters by using the current model $\lambda = (A, B, \pi)$, we will have a new model $\overline{\lambda} = \left(\overline{A}, \overline{B}, \overline{\pi}\right)$ which is more likely to produce the observations sequence O.

By using this procedure iteratively as $\overline{\lambda}$ in place of λ and repeat the re-estimation calculation, we then can improve the probability of O being observed from the model until some limiting point is reached. In other words the iterative re-estimation procedure continues until no improvement in $P(O|\lambda)$ is achieved [28].

CHAPTER 3: EXPERIMENTAL WORK

In this chapter, the experimental work on continuous speech recognition system for Turkish Language based on Triphone Model will be presented.

The system uses MATLAB and Hidden Markov Model Toolkit (HTK) for data preparation, model training and recognition. The HTK is a free and portable toolkit for building and manipulating HMMs primarily for speech recognition research, although it has been widely used for other topics.

MATLAB is used for data preparation and preprocessing steps and HTK is used for implementing our system which has been used after completing the preprocessing step.

3.1 Data Preparation

Data Preparation is the first stage for building speech recognition system and includes the following steps:

3.1.1 Preparing Dictionary

Two different databases are used, one of them is more commonly formed TURTEL speech database which are collected at the acoustics laboratory of TÜBİTAK-UEKAE (National Research Institute of Electronics & Cryptology, The Scientific & Technical Research Council of Turkey), that is used for speaker independent system tests and the other one is weather forecast reports database that is used for speaker dependent system tests.

The goal of the system is to recognize weather forecast reports with a rate as high as possible. First of all, a word grammar about weather forecasting has to be formed to know which words will be recorded.

Then, every report passed through some preparation processes. First numbers changed to their text equivalents. Then all characters changed to their lower case versions, after that Turkish characters are changed with uppercase English ones like ‘ş’ to ‘S’, ‘ö’ to ‘O’ etc.

3.2 Recording the Data

Recording of training and test data are made using an ordinary desktop microphone and a laptop with a sampling rate of 8000 Hz, and with 8 bit quantization. Records have speech data consisting of words.

Since these records will be used in the other steps, contents of each file must be known. For this purpose a text file is created and associated with each record file. These files contain the text version of the recorded speech data: For example text file associated with a record of word “sIfIr” can be like this:

sIfIr

By using these files:

i. Dictionary file and word list file are formed.

ii. In offline and online recognition, results of the recognition are compared with the speech data in the records text file.

3.3 Preprocessing

Preprocessing is the process of preparing speech signal for further processing, in the other words to enhance the speech signal.

In preprocessing step, after enhancing speech signal (see Chapter 2 preprocessing) dictionary and word list files are formed. Word list file consists of the words in the record text files as one word per line:

Canakkale

CankIrI

Cevreleri

Cevrelerinde.

.

This file is used in forming dictionary file and the language model. Dictionary file consists of all the words in the word list. Format: all the file is an shown below:

word [word representation] word pronunciation

For example a few lines from dictionary file can be like:

CIG [CIG] C I G

Cevreleri [Cevreleri] C e v r e l e r i

OGle [OGle] O G l e

.

Dictionary file is used in training and recognition processes.

3.4 Feature Extraction

Feature extraction is performed after preprocessing steps. One important fact about the speech signal is that it is not constant from frame to frame. For this reason, speech must be converted into the parametric form. Block diagram all the feature extraction stages can be given as follows:

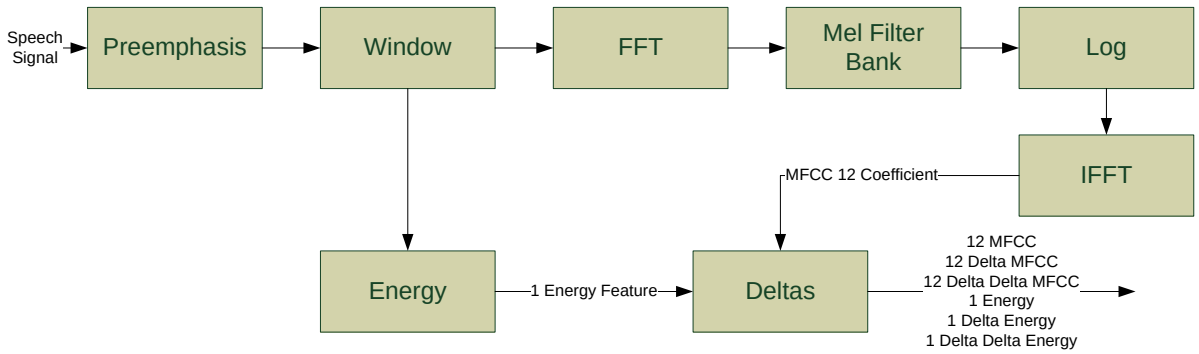

Figure 14: MFCC feature vector extraction process

Preemphasis:

This operation is used not to lose high frequency speech information of the incoming speech signal and also to make the signal spectrum flatter.

So, first, overall energy distribution of the signal is balanced in preemphasis stage by using a FIR filter ($\mathrm{H(z)} = 1 - \mathrm{a\,z^{-1}}$). The effect of this filter to the signal is:

$$y(n) = x(n) - 0.95x(n-1)$$

By this way, it emphasized the high frequencies before processing. Also, the preemphasis filter is used to remove the spectral roll off which is caused by the radiation effects of the sound from the mouth.

The value for a usually ranges from 0.9 to 1.0. For our system, we choose a=0.95. We found an optimum value by trial and error.

Frame Blocking:

In this step the reemphasized speech signal is blocked into frames of N=200 samples (25ms) and each frame overlaps with the previous frame by a predefined size that in our case separated by M=80 samples (≅10 ms, typical to be stationary signal). The first frame consists of first 200 speech samples. The second frame begins 80 samples after the first frame, and overlaps it by 200-80 samples. Similarly, the third frame begins 2*80 samples after the first frame. Each signal is now converted into a set of fixed frames, with each frame is overlapping. The goal of the overlapping scheme is to smooth the transition from frame to frame.

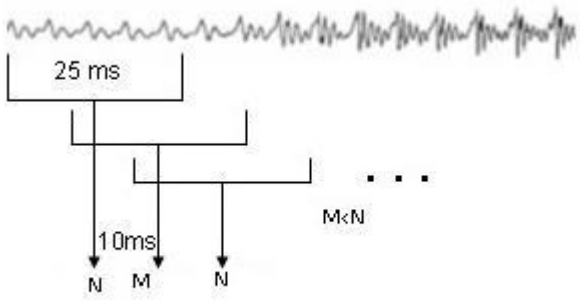

Figure 15 : Frame blocking scheme

Windowing:

Each individual frame is windowed to minimize the signal discontinuities at the borders of each frame. We used the Hamming window which is far more successful than rectangular window in attenuating window edges. Hanning window can also be a choice. As can be seen from the figure that the difference is so small between Hamming and Hanning. For not violating the assumption of being stationary for signals a 25ms Hamming window is used. In our case every window has N=200 samples. (N is the number of samples per frame)

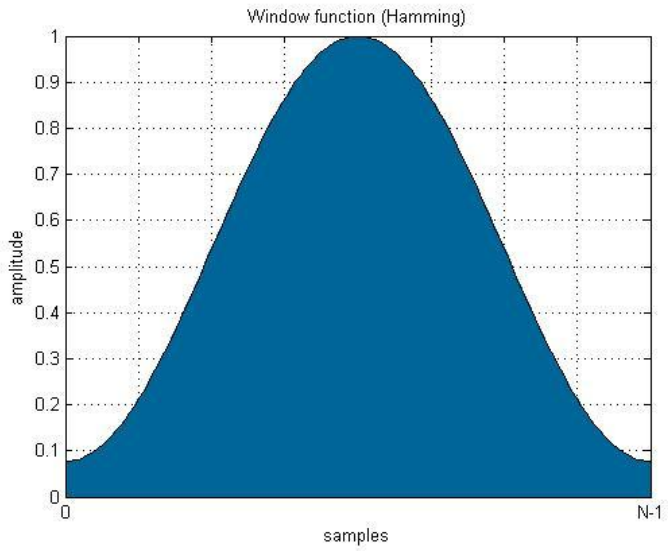

Figure 16 : Hamming Window

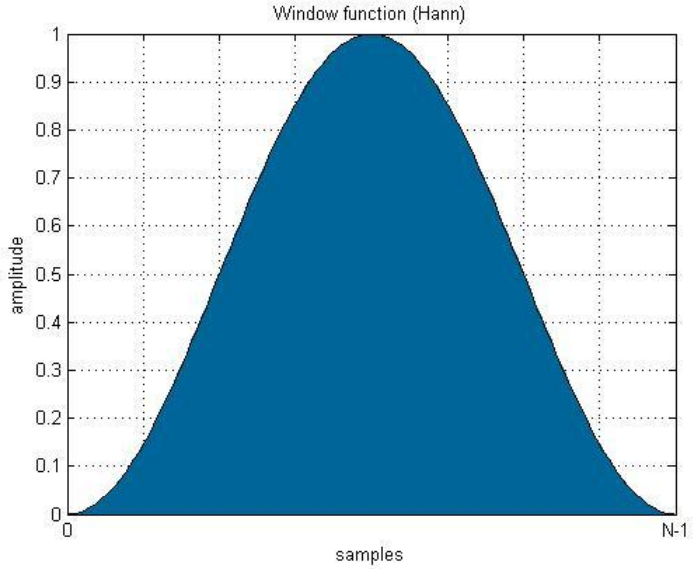

Figure 17 : Hanning Window

Fast Fourier Transform:

After the windowing step, we need to extract spectral information and know the energy amounts in each frequency band. FFT is used to convert each frame of N (200 samples) sample from the time domain to the frequency domain.

In order to lead FFT do it is job efficiently, N, point number, has to be a power of 2. Here N is selected as 256 points (200 samples $\cong 2^8 = 256$). Fast Fourier Transform is used with zero padding (256=200 samples+56 zero). Zero-padding append an array of zeros to the end of the input signal before FFT.

Zero-padding increases the number of data points to a power of 2 and provide better resolution in the frequency domain.

Mel Filter-Bank:

At this point, Speech signals have been transformed to the frequency domain and now frequency spectrum have to be converted to Mel spectrum. Because MFCC's apply Mel scale to the spectrum of speech in order to imitate human hearing mechanism.

It turns out that modelling this property of human hearing system during feature extraction improves speech recognition performance [22].

First the magnitude of the FFT components is computed, then this magnitude spectrum is multiplied with overlapping triangular filters in Mel scale.

In this work 26 triangular filters are formed to gather energies from frequenciy bands. They are all linearly spaced along Mel scale. According to the cut of 120hz-3400hz, 14 of these filter are below 1000hz and rest of them are above 1000hz.

Discrete Cosine Transform:

In the final step we use the DCT to convert the log mel-scale spectrum back to time domain.

When trying to recognizing phone identity the most useful information is the position and shape of the vocal tract. If we can know the shape of the vocal tract, we can determine which phone was produced. On the last step of the MFCC operation, by using DCT we separate the vocal tract information from the unnecessary (for phone detection) glottal source information. Generally the first 12 coefficients which we will get from DCT are enough for MFCC purposes. Higher cepstral coefficient can be used to determine pitch information.

Now we have Mel spectrum of the signal, next step is determining the Mel cepstral coefficients. To do this, logarithms of filterbank amplitudes are taken, and then using DCT, 12 DCT coefficients and 1 energy coefficient are calculated (39 size vector with 12 delta coefficients plus 1 energy and 13 double deltas).These coefficients are calculated in Mel scale and they are FFT based cepstral coefficients.

The main reason of preferring DCT rather than IFFT is IFFT requires complex arithmetic calculations compared to DCT.

3.5 Model Creation

Creating a particular model is done by HMM from the speech samples. While creating model, there are plenty of choices about what will be the basis of the recognition like phoneme-based, whole word-based, triphone-based, biphone-based etc.

In this work triphone based approach is used since triphones are phones with context information. Also in whole-word based systems to add a new word to the vocabulary, new training data and a new model creation is needed. So collecting training data becomes a problem for a large vocabulary system in whole-word models. Several record examples are needed for each vocabulary word and for a great number of speakers the numbers go beyond thousands. On the other hand because of all the large training data, the model state count stored in the memory will be large and the calculations will be slow.

In triphone based approach each phone is considered with its neighbour left and right phones. So same phones with different neighbours have different state parameters. For this reason, triphone based model will be more successful and feasible to use.

3.5.1 Monophone Model

The creation of monophone models starts with preparation of the training and testing data. The process begins a default prototype HMM for every phone. Creation of the monophone HMM however requires specifying the number of states prior to training. Our experimentations suggested utilizing 5-state HMMs for the purpose of acoustic modeling.

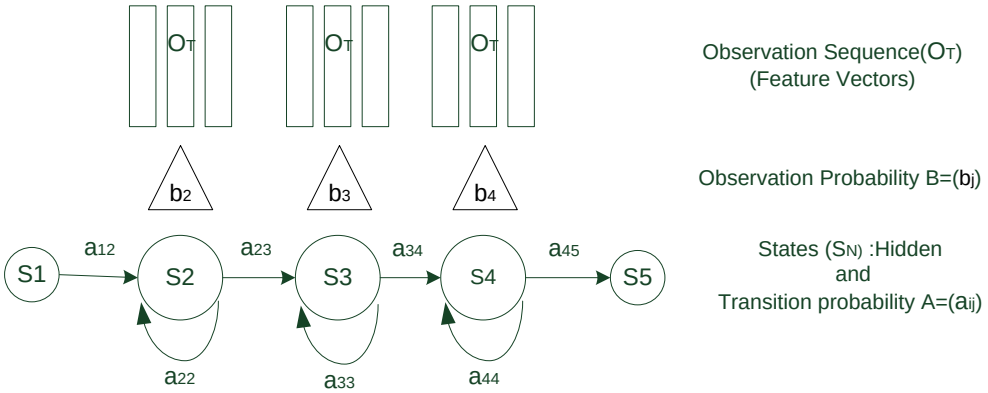

Figure 18 : Sample HMM Model (state 1 and 5 denote none-emitting states)

Monophones are modeled as 5-state left-right HMM with no state skips. Entry and exit states are non-emitting so the rest 3 states are emitting states that have probability distributions. With this approach if there is no pause between the words, this model is skipped without consuming any observation. Observation density functions are assumed to be Gaussian. Each Gaussian distribution is represented by a mean vector and a covariance matrix.

Monophone models are created from these phones. 31 monophone models are used. (29 phone models + a silence model + a short pause model which is the short pause that can occur between two words).

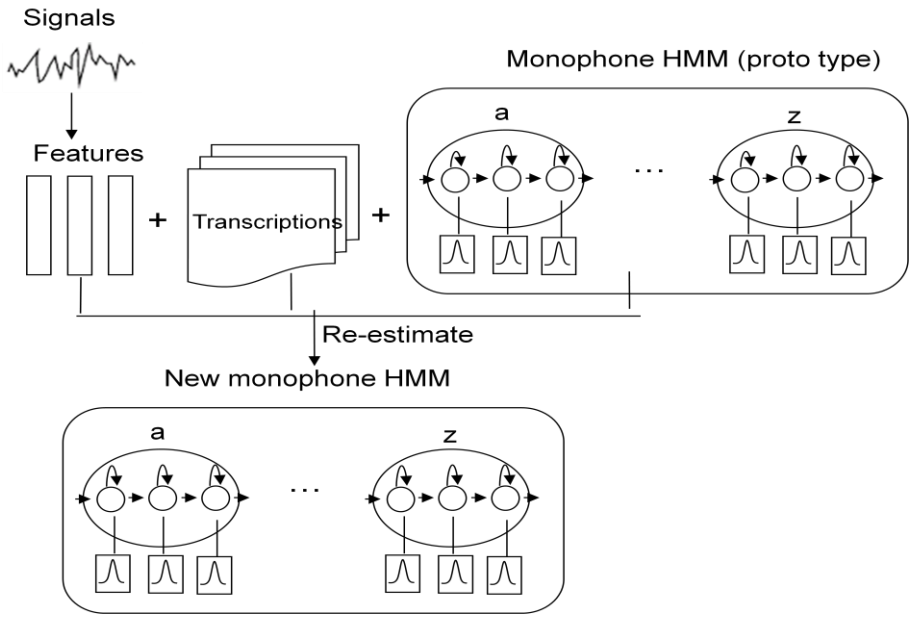

Figure 19 : Making Monophone Model

However the monophone based models cannot capture the variation of a phone with respect to the context. Phones are found to vary depending on the preceding and succeeding phones [36] and this aspect needs to be captured within the acoustic models to improve performance.

3.5.2 Triphone Model

Triphone based modeling involve capturing the context information within the phone models and it requires more training data for successful model generation [37].

Basically, triphones are phones with context. Each phone is considered with its neighbour left and right phones. The number of triphones is much greater than the number of phones [38] . Ideally, if N words are involved in the training of monophone models, training of triphone models would require N^3 words. We have special collected data that has 373 words (TURTEL) for representing 80% of the triphones contained in Turkish.

Triphone models are built up appending the monophone models according to pronunciation dictionary, for example, phones of the word “bitki” is:

b+i

b-i+t

i-t+k

t-k+i

k-i

Figure 20 : Triphone model of 'bitki'

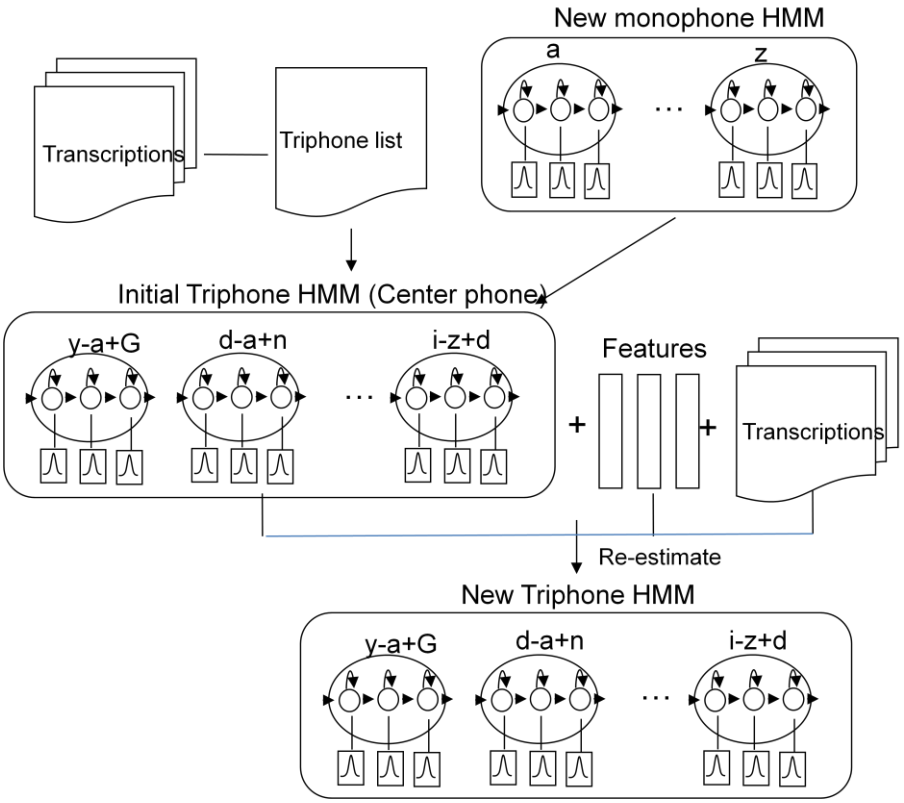

Figure 21 : Making Triphone from Monophone

3.5.3 Language Model

Since linguistic knowledge makes the recognition results better, a bigram language model with back-off smoothing is also used in this work. Bigram model is selected because it will be enough for our vocabulary size. The threshold used is 1, so the bigrams that are not observed or observed once are assigned a nonzero probability.

We have record text files that contain the utterance texts and word list file that contains all the words used in text files. Back-off bigram models are built with according to these files using the equations given:

The backed-off bigram probabilities are given by

$$p(i,j) = \begin{cases} \frac{N(i,j)-D}{N(i)} & if N(i,j) > t \\ b(i)p(j) & otherwise \end{cases} ,$$

and the unigram probabilities $p(i)$ are given by

$$p(i) = \begin{cases} \frac{N(i)}{N} & if\ N(i) > u \\ \frac{u}{N} & otherwise \end{cases}$$

Where;

$\boldsymbol{N(i,j)}$ is the number of occurrences of the word pair (w_i, w_j)

$\boldsymbol{N(i)}$ is the number of occurrences of the word w_i

$\boldsymbol{L}$ is the number of distinct words

The back-off weight $b(i)$ is calculated to ensure that $\sum_{j=1}^{L} p(i,j) = 1$.

3.6 Training

The training process contains several steps. At each step, the system is modified, and after each step re-estimation of the acoustic model parameters is done several times. Acoustic models are formed as Hidden Markov Models. The parameter estimation is implemented using a computationally efficient algorithm known as Baum-Welch reestimation (also referred to as the Forward-Backward algorithm). Baum-Welch training can be viewed as providing the system a capability to make soft decisions.

Training process start with initialization of monophone models; global mean and variance of the training data is assigned to each state by using flat start method whereby all models are given the same initial parameters. After initialization, triphone models are initialized using monophone models and retrained using embedded training method. In embedded training composite HMM is created by concatenating subword models that corresponds to the subwords in the record text files.

For example composite HMM of the word “sivas”:

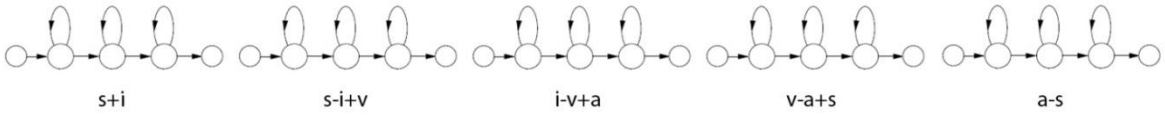

Figure 22 : Concatenating triphone HMM for a composite HMM

To limit the complexity of the triphone models and avoid the sparse data problem in acoustic training, decision tree based state clustering is used. Also decision tree clustering is used to increase robustness. Decision tree clustering method uses phonetic decision trees for tying the parameters of the context dependent HMM [30]. This process provides functionality of generating unseen triphones. A phonetic decision tree is a binary tree where each node has a question attached. The questions are used to get information about the context of investigated phone. The critical point in design of the question set is using the similarity of pronunciation. Considering each style of the pronunciation, we obtained the left and the right questions which are consisting of 12 left questions and 12 right ones, totally 24. (See. Appendix A)

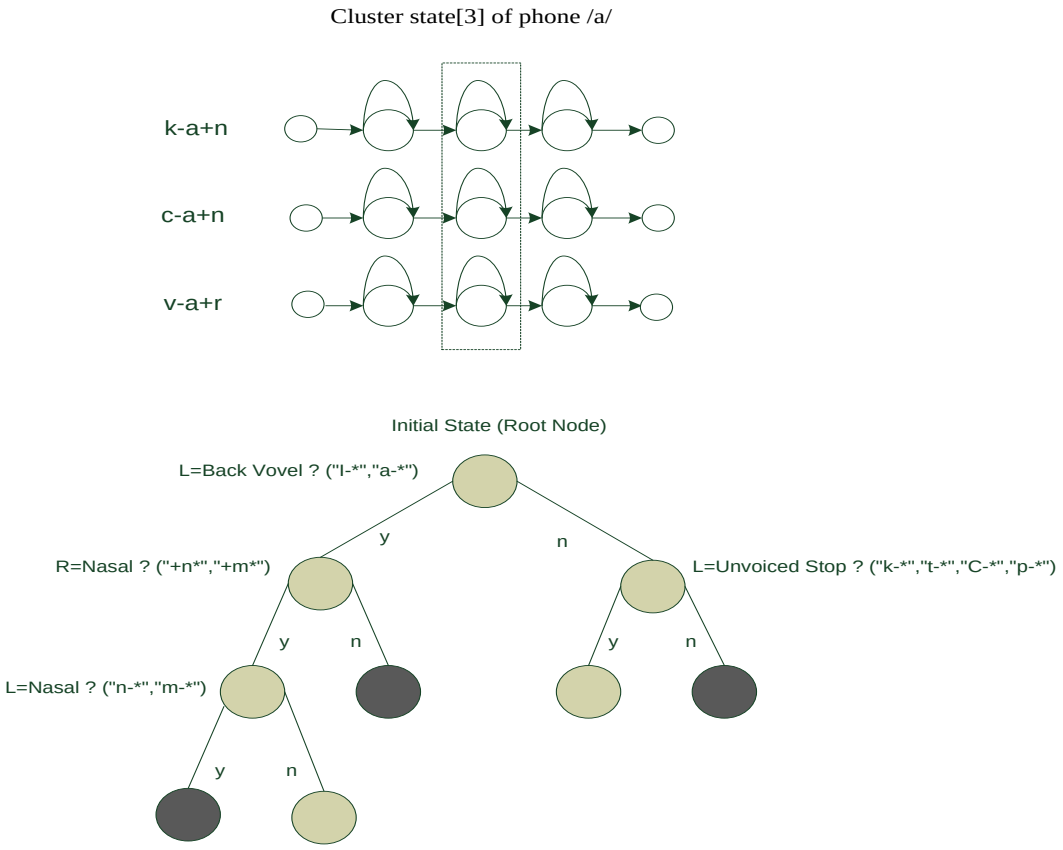

Figure 23 : A sample decision tree for phone *[a]*

Initially, one tree is constructed for each state of each phone and states with the same central unit are pooled in the root node. Then, these states are split into two groups according to the question that can result in a maximum likelihood increase on the training data. The process is repeated until the log likelihood increase is smaller than a predefined threshold.

After preparation of training modules, embedded training is performed:

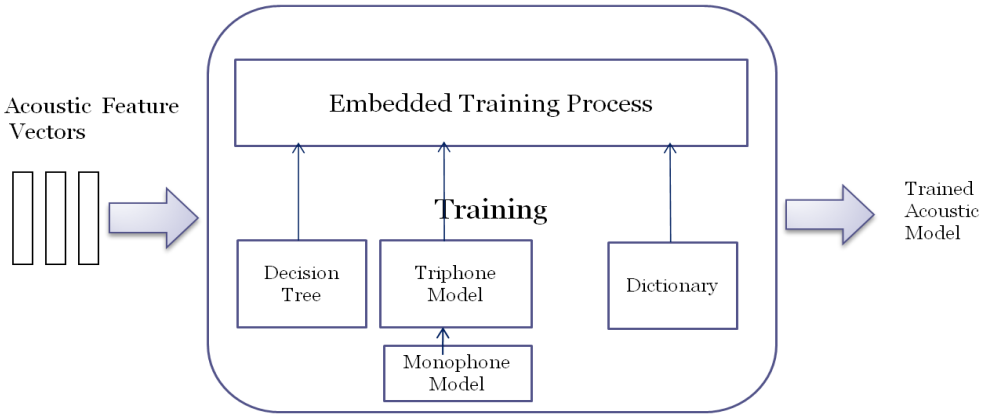

Figure 24: Training process

Traditional HMM training definition will often select one of the linear paths and update the corresponding models. Baum-Welch reestimation is performed across all networks simultaneously. Since the Baum-Welch algorithm is used at each level, this approach effectively makes soft decisions about modelling pronunciations. It leads to better generalization during recognition, since unseen pronunciations can potentially occur in the network training model.

Train is started with global mean and variance which is called flat start method, then the forward and backward probabilities are calculated for the each HMM. The forward and backward probabilities are used to compute the probabilities of state occupation. At the first iteration of embedded training each utterance is segmented for aligning the models. The models are aligned as intended on the second and following iterations.

3.7 Recognition

Goal of recognition is to find the most likely string of symbols (e.g. words) to account for the observed speech waveform:

$$\hat{W} = arg\max_W P(O|W)P(W) \quad (3.1)$$

Where:

W : word sequence

O : observation sequence

In other words it is the search for the uttered word sequence by finding the best path that goes through the state sequence that fits the feature vector sequence of the input speech signal. This is also called decoding. In this thesis Viterbi decoding with token passing is used. Viterbi algorithm finds the best sequence of states through the word. It operates on a search network that consists of nodes connected by transitions. This network efficiently allows n-gram language models to be applied during search. So the recognition network can be thought as a hierarchic structure that has 3 levels:

words -> models -> states.

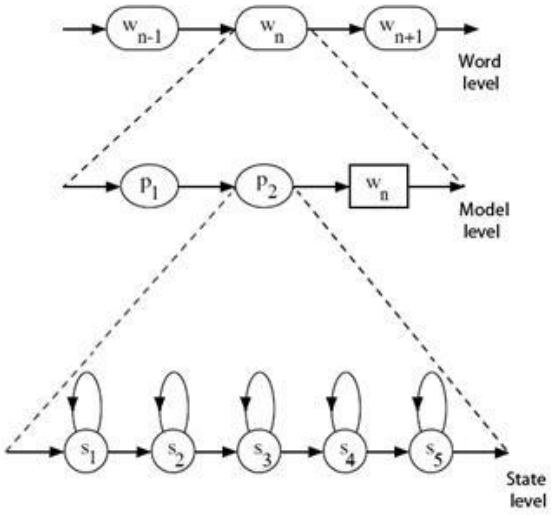

Figure 25 : Recognition Network

Decoder searches best possible path in the network. Paths are the ways from the start node to the exit node of the network that passes through the emitting HMM states. The job of the decoder is to find the paths in the network which have the highest log probability. These paths are found using tokens.

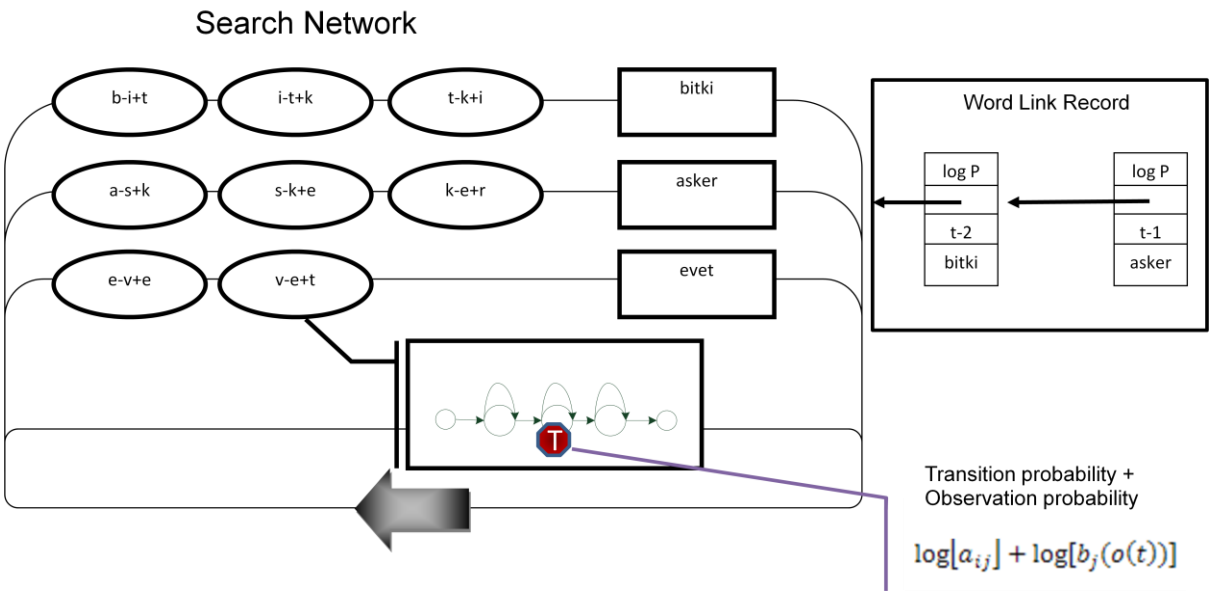

Figure 26 : Recognition search network

It works on a finite state network which is formed using acoustic model, dictionary and word lexicon. Network can be thought as a big composite HMM.

Tokens propagate on states of the inner HMMs and at every frame only the best token in each state continues its way. By using Word Link Records the best matching word sequence is found. Link Record is generated which stores the identity of the HMM from which the token has just emerged and the current value of the token's link [39].

In addition Viterbi with token passing algorithm has capability to use bigram language model probabilities efficiently. Tokens emitted from word-end nodes will have a language model probability added to them before entering the following word.

3.8 Offline Recognition

The proposed system is able to perform two types of recognition. One of them is "offline recognition" and in this type, system tries to recognize utterances that have already been recorded.

Unlike live recognition in offline recognition a much more detailed performance analysis is done. After the recognition process is carried out, the recognized words are compared with the original texts in the records text files.

3.9 Live Recognition

The other type of recognition that proposed system capable of is online recognition. In online recognition, first user is prompt to speak an arbitrary sentence to measure speech and background silence levels. Then when recognition process is ready to perform, a ready message is given to user. System accepts input until there is a longer pause in utterance. Shorter pauses between words do not break input.

If a long pause happens then the recognition is carried out and the recognized words are displayed on screen.

After the results are given another ready message is given and system is ready to recognize other utterances. This mechanism can be thought of recognition sessions separated with long pauses.

CHAPTER 4: SIMULATION RESULTS

The tests are performed on two different databases using the trained acoustic model. In the tests, three main parameters that considerably affect the performance of the system are changed to determine the optimum values for these parameters.

These parameters/factors are:

- usage of the bigram language model based on the utterance texts and word list

- pruning factor in the decision tree clustering which effects the clustering process, thus the training performance

- mean (DC offset) which can be presented by the analog to digital conversion of the original signal

4.1 Properties of the Test Data

The subjects of collected test data can be divided into two categories: general and weather.

Category	# of Words	# of sentences	Duration
General (TURTEL)	373	15	6 hours and 46 minutes
Weather	739	17	2 hours and 33 minutes
Total	1112	32	9 hours and 19 minutes

Table 1 : Collected data

TURTEL Speech Database:

Telephony read speech in Turkish, fully phonetically labeled. The data comprises 373 words and 15 sentences. These words and sentences are selected to represent 80% of the tri-phone usage in Turkish. The corpus is made of 93 speaker recordings. 36 of the speakers are female. 58% of the data is from normal telephone, %19 is from mobile and 23% is from speaker phone speech. Geographical origin, age, and other speaker dependent features are identified. From this corpus 65 of the speakers are advised to be used for system training while 28 are proposed for testing. The total length of the corpus is 6 hours and 46 minutes of speech (see Appendices B).

Weather forecast reports database:

Daily reports of the Turkish State Meteorological Service gathered from the website between dates 10.01.2009 and 25.05.2009. It has 739 words from one speaker. These words selected from daily weather news. The data comprises 739 words and 17 sentences. The corpus is made by a speaker who is female.

4.2 Parameters of the System

In the recognition, experiments are conducted for the test words and sentences described above. For the tests, HTK's tools are used.

The model topology is an HMM with one single Gaussian per state which means that each observation probability is defined as a Gaussian probability density function that can be represented by a mean vector and a covariance matrix**.** The feature extraction of the system consists of extracting a 39 dimensional feature vector.

A bigram language model is also used to increase recognition rate in our experiments.

4.3 Measures of Recognition Performance

We have used the percentage of correctly recognized words as in (4.1) and the accuracy of recognition measures as in (4.2) to analyze the speech recognition performance of the models.

$$\%Correct = \frac{H}{N} x100\% \quad (4.1)$$

$$Accuracy = \frac{H-I}{N} x100\% \quad (4.2)$$

H is the number of correctly recognized words.

I denotes the number of insertions and

N denotes the total number of words in records text files.

4.4 Experimental Results

The speech data is divided into two categories: test and train. Test speech data is not used during the training of acoustic model.

In TURTEL database tests, 10 male and 10 female speakers are used for the test of 93 speakers. In weather forecast report database test, 1 female speaker is used.

To examine the recognition performance of the system, triphone based acoustic model is tested with different decision tree pruning factors and also using a bigram language model based on the utterance text files and word list file. Another changed test parameter is the mean (DC offset). In some of the tests DC mean of the signal is also removed to observe the effect on the performance. It is known that if the analog to digital conversion has added a DC offset to the original signal then this DC mean removing operation will improve the performance of the system.

In TURTEL experiments, the tests are run on the same test corpus (15 sentences contain 68 words for each speaker).

DT Pruning	LM is used?	Mean (DC Offset)?	Sent Corr%	Word Corr.%	Word Acc.
350	No	No	26.67	60.20	59.08
350	No	Yes	25	60.49	59.18
350	Yes	No	82.33	95.42	91.69
350	Yes	Yes	83.33	95.66	92
50	No	No	28.33	62.10	60.97
50	No	Yes	28.67	63.56	62.10
50	Yes	No	83	95.81	92.15
50	Yes	Yes	84	95.97	92.47

Table 2 : Recognition performance under all conditions (For TURTEL).

The statistics given on the Table 2 shows TURTEL test results, there were 1360 words and 300 sentences in total. In the table we see that, the average percentage of test samples that are correctly recognized is in range of 60-65. However after the system tests has repeated with language model and with optimum pruning factor, these adjustments improved the performance by 25-35 percent and the word accuracy reached 92-96 percent for all tests.

For weather forecast reports, the tests are run on the same test corpus (17 sentences contain 311 words).

DT Pruning	LM is used?	Mean (DC Offset)?	Sent Corr%	Word Corr.%	Word Acc.
350	No	No	47.06	92.06	89.21
350	No	Yes	52.94	92.36	89.49
350	Yes	No	64.71	98.36	95.39
350	Yes	Yes	64.71	98.68	95.71
50	No	No	52.94	94.87	92.63
50	No	Yes	41.18	93.65	91.43
50	Yes	No	70.59	98.68	96.05
50	Yes	Yes	70.59	99.01	96.37

Table 3 : Recognition performance for weather reports under various conditions.

The statistics given on the Table 3 shows weather forecast report database test results, there were 311 words and 17 sentences in total. System performance is in between 92-98 percent at the implementation with default values. After applying the same arrangements, system accuracy has reached to 94-99 percent.

According to the table 1 and table 2, we can say that an optimum system can be introduced using a decision tree pruning factor as 50 with removing mean (DC Offset) and using the language model.

In conclusion, it is observed that for a high level continuous speech recognition performance, it is necessary to use a well defined language model with also using linguistic properties of the language.

CONCLUSION

In this thesis, a triphone model based, continuous speech recognition system is developed for Turkish language.

Besides acoustic model which is the core part of the system, a statistical language model is also used to improve recognition performance.

Turkish has an agglutinative morphology with productive inflectional and derivational suffixations. Because of these productive suffixations the number of words in vocabulary is very high. This means that a whole word recognition approach cannot be feasible for Turkish language. Therefore, triphones are used as the smallest unit in the acoustic model and they are modelled by 5-state left-to-right Hidden Markov Models which are also called Bakis model. Linear flow of the speech production process makes Bakis type of Hmm much more appropiate than ergodic type. In Bakis type HMMs there is a start state and an end state; also states are connected only with successive states. These properties of the Bakis type HMMs make them a better choice than ergodic types.

When designing large vocabulary cross-word triphone system it is unavoidable that there will be triphones which has no examples in the training data. To overcome this problem a decision tree approach is also investigated and implemented. In recognition tests it is seen that trying different values and finding an optimum for the pruning factor in the decision tree clustering is worth to work on, because a good pruning value can come with a 2-3% increase in the recognition accuracy when performing without a language model (see Table 2 and Table 3).

Training process is done with embedded training using Baum-Welch algorithm. By this way parameter estimation process for all models occurred in parallel.

During recognition process, a search network is formed from the word transcriptions and Token Passing algorithm, which is an implementation of Viterbi algorithm, is worked on this network for the search of the best state sequence.

Removing DC mean of the signal has added also as an evaluation parameter, because if the analog to digital conversion has added a DC offset to the original signal then this DC mean removing operation improves the performance of the system and test results shows that this improvement is around 2-3%.

A bigram statistical language model is also used in the recognition and proposed system tested on two different speech databases, TURTEL Speech Database and Weather Forecast Reports Database, with different pruning factors and also with/without using a language model.

The tests with these databases shown that for large vocabulary cross-word recognition systems, language models can provide a good amount of increase in the recognition accuracy.

The experimental results had shown that the proposed system worked well on both speech databases and had accuracy between 59-96% in correctly identifying a speech sentence.

FUTURE WORK

We considered that the language model can be a good topic to make improvements. A bigram language model is used in this thesis. The bigram language model gives the simplest measure of word transition probability, but ignores most of the preceding context.

It is more likely to be able to capture longer distance dependencies between words if the language model uses more contexts. So by using a trigram language model in the place of a bigram language definitely provide an increase in the recognition performance.

In trigram language model, the count of the triple of (wn-2, wn-1, and wn) is used to estimate the probability of a word given its predecessors. One of the problems that can be face to face is that for many triples, the number their occurrences can be low and so reliable estimates cannot be done. To overcome this situation smoothing techniques can be investigated.

Another topic for future work can be robustness against noise and distortion. Today usage of speech recognition applications goes beyond laboratory environments, so noise and distortion will be inevitable. Some of the noise compensation methods for Hidden Markov models in [40] can be investigated and put into practice for this work.

REFERENCES

[1] **Syrdal, Ann K., W. Bennett Raymond, Steven L. Greenspan, " Applied Speech Technology", CRC Press, 1994.**

[2] **Bristow, G., "Electronic Speech Recognition", McGraw-Hill Book Company, 1986.**

[3] **Dumitru, C. Octavian and Inge. "Continuous Speech Recognition System Based Statistical Modeling", Msc Thesis, Gavat Politehnica University of Bucharest,2002.**

[4] **Yilmaz, C., "A large vocabulary speech recognition system for Turkish", Msc Thesis, Bilkent University, 1999.**

[5] **Deller, J. R., J. G. Proakis, J. H. L. Hansen, "Discrete-Time Processing of Speech Signal", Wiley-IEEE Press, Sep 1999.**

[6] **Acar, D., "Triphone Based Turkish Word Spotting System", MSc Thesis, The Department of Electrical and Electronics Engineering, The Middle East Technical University, 2001.**

[7] **Hakkani-Tur, D., K. Oflazer, G. Tur, "Statistical Morphological Disambiguation for Agglutinative Languages", Technical Report, Bilkent Universty, 2000.**

[8] **Hayes, H. Monson, "Statistical Digital Signal Processing and Modeling", John Wiley & Sons Inc., Toronto,1996.**

[9] **Hiroaki, S., "Two-Level DPMatching A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 27, Dec 1979.**

[10] **Young S., G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, "The HTK Book (for HTK v3.4), Cambridge University Engineering Department, Dec 2006.**

[11] **20 June 2009, <http://en.wikipedia.org/wiki/Forward-backward_algorithm>.**

[12] **10 July 2009, <http://en.wikipedia.org/wiki/Viterbi_algorithm>.**

[13] **Çarki, K., P. Geutner, T. Schultz, "Turkish lvsr: Towards better speech recognition for agglutinative languages", in proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 3, June 2000.**

[14] **Man, K.F., K. S. Tang, and S. Kwong, "Genetic Algorithms", Springer, 2001.**

[15] **Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, Feb 1989.**

[16] **Bahl, L.R., R. Bakis, P.S. Cohen, A.G. Cole, F. Jelinek, B.L. Lewis, and R.L. Mercer, "Further Results on the Recognition of a Continuously Read Natural Corpus", ICASSP, 1980.**

[17] **Rabiner, L.R., M.R. Sambour, "An algorithm for determining the endpoints of isolated utterances", The Bell System Technical Journal, vol.54, no.2, Feb 1975.**

[18] **Md.Rashidul, Hasan, M. Jamil, Md.Golam Rabbani and Md.Saifur Rahman, "Speaker Identification Using Mel Frequency Cepstral Coefficients", ICECE 2004, Dec 2004.**

[19] **Mengüsoglu, E., O. Deroo, "Confidence Measures in HMM/MLP Hybrid Speech Recognition for Turkish Language", Proceedings of the ProRISC/IEEE workshop, 2000.**

[20] **Nilsson, M., M. Ejnarsson, "Speech Recognition using Hidden Markov Model performance evolution in noisy environment", Department of Telecommunications and Signal Processing Blekinge Institute of Technology, March,2002.**

[21] **NATO Research and Technology Organisation, "Use of Speech and Language Technology in Military Environments", RTo Technical Report, Information Systems Technology Panel, December 2005.**

[22] **Mitianoudis, N., "A graphical framework for the evaluation of speaker verification system", Technology and Medicine University of London, MSc Thesis, Sep 2000.**

[23] **Büyük, O., H. Erdogan, C. Bozsahin, "An outline of Turkish Morphology. Report on Turkish Natural Language Processing Initiative Project", Oct 1994.**

[24] **Çilingir, O., "Large Vocabulary Speech Recognition for Turkish", Middle East Techical University, Turkey, 2003.**

[25] **Banerjee, P., G. Garg, P. Mitra, and A. Basu, "Application of Triphone Clustering in Acoustic Modeling for Continuous Speech Recognition in Bengali", 19th International Conference on Pattern Recognition ICPR, 2008.**

[26] **Rabiner, L., R. Levison, S. E., "Isolated and Connected Word Recognition – Theory and Selected Application.", Communications, IEEE Transactions, Vol. 29, Issue 5, May 1981.**

[27] **Rabiner, L., Juang Bing-Hwang, "Fundamentals of the Speech Recognition", Prentice Hall, New Jersey, 1993.**

[28] **Boyle, R.D., <http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html>.**

[29] **Levinson, S.E., L.R. Rabiner, and M.M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. ", The Bell System Technical Journal, Vol.62, No.4, April 1988.**

[30] **Jurafsky, D., James H. Martin, "Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition", Prentice Hall Series, May 2008.**

[31] **Holmes, J. N., "Speech Synthesis and Recognition", CRC Press, 2001.**

[32] **Huang, Xuedong, A. Acero, and H. Wuen Hon, "Speech - Spoken Language Processing - A Guide to Theory, Algorithm and System Development", Prentice Hall, 2001.**

[33] **Sahin, S., "Large vocabulary modeling for Turkish continuous speech recognition", Msc Thesis, Middle East Technical University, 2003.**

[34] **Cook, Stephen, "Speech Recognition HOWTO", 19 April 2002, <http://www.faqs.org/docs/Linux-HOWTO/Speech-Recognition-HOWTO.html>.**

[35] **Young, S., J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling", in Proceedings ARPA Workshop on Human Language Technology, 1994.**

[36] **Çiloglu, T., M. Çömez, S. Sahin, "Takılı bir Dil Olarak Türkçe için Dil Modelleme", Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 2004.**

[37] **Tunali, V., "A Speaker Dependent, Large Vocabulary, Isolated Word Speech Recognition System for Turkish", Msc Thesis, Marmara University, 2005.**

[38] **Wayne, A. L., "Trends in Speech Recognition", Prentice Hall Inc., Signal Processing Series, 1980.**

[39] **Sung, Y.Hsuan, "Speech Recognition and Synthesis Lecture 9".**

[40] **Vaseghi, S.V., B.P. Milner, "Noise compensation methods for hidden Markov model speechrecognition in adverse environments", Speech and Audio Processing, IEEE Transactions, Vol. 5, Jan 1997.**

APPENDICES A

Decision Tree Questions

QS "L_front_vowel_1" {"i-*","e-*"}

QS "R_front_vowel_1" {"*+i","*+e"}

QS "L_rounded_vowel_1" {"U-*","O-*"}

QS "R_rounded_vowel_1" {"*+U","*+O"}

QS "L_back_vowel_3" {"I-*","a-*"}

QS "R_back_vowel_3" {"*+I","*+a"}

QS "L_back_vowel_2" {"u-*","o-*"}

QS "R_back_vowel_2" {"*+u","*+o"}

QS "L_nassal" {"n-*","m-*"}

QS "R_nassal" {"*+n","*+m"}

QS "L_voiced_stop" {"g-*","d-*","b-*"}

QS "R_voiced_stop" {"*+g","*+d","*+b"}

QS "L_unvoiced_stop" {"k-*","t-*","C-*","p-*"}

QS "R_unvoiced_stop" {"*+k","*+t","+C","*+p"}

QS "L_fricative_voiced" {"z-*","v-*"}

QS "R_fricative_voiced" {"*+z","*+v"}

QS "L_fricative_unvoiced" {"S-*","s-*","f-*"}

QS "R_fricative_unvoiced" {"*+S","*+s","*+f"}

QS "L_whisper" {"h-*"}

QS "R_whisper" {"*+h"}

QS "L_silence" {"sil-*","sp-*"}

QS "R_silence" {"*+sil","*+sp"}

QS "L_other_consonants" {"y-*","r-*","l-*","h-*","c-*"}

QS "R_other_consonants" {"*+y","*+r","*+l","*+h","*+c"}

APPENDICES B

1	bitki	46	içen	91	çabukluk	136	nene
2	tutulmak	47	kabartma	92	ağırlık	137	gibi
3	bir	48	acılık	93	kalınca	138	yaşar
4	gösterilen	49	süreli	94	görevliyi	139	yıl
5	nakışlarla	50	kimsenin	95	karşılamalıdır	140	dinle
6	kaza	51	testi	96	katılmış	141	yakasız
7	dervişlerin	52	boğumları	97	yarıp	142	yoluyla
8	veya	53	ilgili	98	kendi	143	varken
9	keserek	54	sonra	99	sevmeyecek	144	önce
10	yapılan	55	kadar	100	kıran	145	ambar
11	karışıyorsun	56	Için	101	yalıtkan	146	fikirli
12	bağlanarak	57	öğrenilmesi	102	matlaşmak	147	alttaki
13	getirmek	58	kaymaklı	103	el	148	mizan
14	başında	59	şaşırmak	104	türlü	149	olduğundan
15	yönteminde	60	ayrıldığı	105	sunum	150	acemileşmek
16	parçanın	61	edilme	106	şeyi	151	yazı
17	arama	62	maddeler	107	aşktır	152	daha
18	sağlayan	63	indirimden	108	bekleyen	153	nokta
19	her	64	kumaştan	109	topluluğu	154	makam
20	eve	65	güneşte	110	sıkıntı	155	değer
21	hastalıklara	66	geldiğine	111	pekişti	156	keten
22	işaretlerini	67	yayınlara	112	donanım	157	onun
23	dolaplar	68	geçirmeden	113	açık	158	kaldırım
24	buğday	69	işi	114	sandım	159	çekmek
25	tahtası	70	kuruluşu	115	artıyor	160	arasındaki

26	genellikle	71	eylem	116	duyguları	161	mübalağa
27	isimlerden	72	cariye	117	çeşit	162	kültürünü
28	bağlantısını	73	düz	118	lise	163	sardı
29	dikkatli	74	merkez	119	yüklenmek	164	durumu
30	işine	75	tuvalet	120	yoksul	165	ali
31	davranışta	76	zaman	121	çıkacak	166	halka
32	bölgesinde	77	kapanmak	122	dal	167	şaştı
33	bere	78	hissetmek	123	koruyucu	168	dönem
34	konu	79	çok	124	hayat	169	alman
35	kurallarını	80	kenetlenmiş	125	silmiş	170	familyası
36	asalak	81	etkiler	126	düşünce	171	selin
37	biçimi	82	bu	127	ortaya	172	ekte
38	tazelemek	83	mesleği	128	belirti	173	özen
39	yersiz	84	lacivert	129	aksiyon	174	ermezler
40	durumuna	85	iklim	130	büyük	175	kararsızlık
41	bakımından	86	harflerle	131	azlığını	176	abartıcı
42	alnına	87	atardamar	132	kullanılır	177	tatil
43	girebilir	88	vurmasını	133	ciddiyet	178	çalı
44	sevecenlik	89	istek	134	gereksiz	179	kavuşturmak
45	anlamı	90	bulunma	135	bütün	180	merhamet

181	canlı	218	Aydın	266	zarar	314	prizma
182	renk	219	sırtından	267	acelecilik	315	hakem
183	benzer	220	kuzey	268	deniz	316	kenter
184	başladı	221	ölümünden	269	kuş	317	asla
185	dilde	222	çivi	270	iyi	318	düğüm

186	kötü	223	yabansı	271	yaptıkları	319	evrim
187	savaş	224	afrika	272	söyledi	320	siyah
188	vakti	225	alfabetik	273	ilişki	321	delil
189	uykusuz	226	otsu	274	dünya	322	dede
190	doğru	227	garip	275	çaylak	323	özüt
191	imtiyazlar	228	ördek	276	ünlüler	324	sekiz
192	abide	229	raf	277	polis	325	çocuk
193	devletin	230	lekeleri	278	sera	326	demokrasi
194	anlatmak	231	sarmaşık	279	altı	327	ebe
195	görülmemiş	232	titizlikle	280	burun	328	eğri
196	mensup	233	esmer	281	hırka	329	kaçak
197	niteliği	234	akmak	282	hüner	330	hap
198	alışmamış	235	gelmek	283	surat	331	fena
199	namaz	236	ordu	284	aynı	332	imdat
200	ceviz	237	oku	285	o	333	bora
201	mutlu	238	demli	286	ani	334	boy
202	türk	239	yitimi	287	soru	335	ruj
203	incitici	240	adıyla	288	ses	336	füze
204	girdi	241	nitelikte	289	hata	337	sülük
205	abajur	242	çengi	290	pilav	338	aldatmak
206	başka	243	bunu	291	sahil	339	illet
207	sayı	244	ettiğin	292	küçük	340	proje
208	yüzde	245	bina	293	ülke	341	olmayan
209	hane	246	taş	294	sanki	342	jale
210	kin	247	söz	295	dönünce	343	jilet
211	emel	248	işiyle	296	kızıl	344	sakat

212	irade	249	ray	297	sükse	345	sıfır
213	aşı	250	yumuşak	298	saçlar	346	bir
214	uzunluğu	251	kıl	299	oynatma	347	iki
215	şan	252	kıvanç	300	türkiye	348	üç
216	insan	253	üst	301	varis	349	dört
217	oyun	254	kol	302	gamlı	350	beş
218	aydın	255	libas	303	kısa	351	altı
219	sırtından	256	ağaç	304	gerçek	352	yedi
220	kuzey	257	dizi	305	zorla	353	sekiz
221	ölümünden	258	kereste	306	kurt	354	dokuz
222	çivi	259	ancak	307	üzüm	355	on
223	yabansı	260	yapma	308	dış	356	yirmi
224	afrika	261	sattığım	309	tamam	357	otuz
214	uzunluğu	262	müdür	310	şam	358	kırk
215	şan	263	şimşek	311	istedi	359	elli
216	insan	264	gözler	312	göreceğim	360	altmış
217	oyun	265	beyin	313	yapacağım	361	Yetmiş
362	seksen	365	bin	368	trilyon	371	katrilyar
363	doksan	366	milyon	369	trilyar	372	evet
364	yüz	367	milyar	370	katrilyon	373	hayır