

Selecting Optimal Number of Gaussian Mixtures for Hindi Language ASR

A thesis submitted in partial fulfillment of the requirement for the award of degree of

Master of Technology

In

Computer Engineering

By

Sonali Kapoor

Roll No. 2K14/SWE/18

Under the supervision of
Prof. O. P. Verma
Head of Department



Department of Computer Engineering
Delhi Technological University
Delhi-110042, INDIA
June 2016



Department of Computer Engineering
Delhi Technological University
Delhi - 110042, India

Certificate

I, **Sonali Kapoor**, hereby declare that the work which is being presented in my M.Tech dissertation entitled “**Selecting Optimal number of Gaussian Mixtures for Hindi Language ASR**”, in partial fulfillment of the requirement for the award of the degree of **Master of Technology (Computer Engineering)** submitted to the Department of Computer Engineering, Delhi Technological University, Delhi is an authentic record of my own work carried out under the supervision of **Prof. O. P. Verma, Head of Department, Delhi Technological University, Delhi**. The work presented in this thesis has not been submitted by me for the award of the degree elsewhere.

Date:
Place: Delhi

Sonali Kapoor
Roll No. 2K14/SWE/18

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:
Place: Delhi

Prof. O. P. Verma
Head of Department
Department of Computer Engineering
Delhi Technological University
Delhi-110042

Declaration

I, **Sonali Kapoor**, hereby declare that the work which is being presented in my M.Tech dissertation entitled “**Selecting Optimal number of Gaussian Mixtures for Hindi Language ASR**”, in partial fulfillment of the requirement for the award of the degree of **Master of Technology (Computer Engineering)** submitted to the Department of Computer Engineering, Delhi Technological University, Delhi is an authentic record of my own work carried out under the supervision of **Prof. O. P. Verma, Head of Department, Delhi Technological University, Delhi**. The work presented in this thesis has not been submitted by in any other University/Institute for the award of the degree elsewhere.

Date:
Place: Delhi

Sonali Kapoor
Roll No. 2K14/SWE/18

Acknowledgements

Foremost, I would like to express my sincere gratitude to my Supervisor **Prof. O. P. Verma**, Head of Department of Computer Engineering for his continuous encouragement, patience, motivation, enthusiasm, and immense knowledge. His guidance and insightful comments helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for this thesis.

The thesis has been kept on track and been seen through to completion with the support of numerous people including my friends, well-wishers. At the end of my thesis, I would like to express my thanks to all those who contributed in many ways to the success of this study. Last but not the least, I would like to thank my parents for their unconditional support, both financially and emotionally throughout my research.

Sonali Kapoor

Roll No: 2K14/SWE/18

M.TECH (SOFTWARE ENGG.)

Delhi Technological University,

DELHI

Abstract

Automatic Speech Recognition (ASR) is one of the classic example of an automatic pattern classification problem. Speech recognition is a typical alternative to typing on keyboard, based on sound-analysis and converting the acoustic data into a text sequence. It is a cybernated speech to text process, in which speech is usually recorded with microphone or a mike by capturing the changes in air pressure. It has several applications in various areas of day-to-day life like movie and train schedules information, voice control of house hold applications mostly home appliances, inquiry of bank balance, dialing telephone numbers by digit or name pronunciation and especially for physically challenged persons. Although enormous progress has been made during the last four decades in the field of Automatic Speech Recognition (ASR) systems, still there is a substantial gap in performance between human and machine.

In India if it would have been possible to provide human like interaction with machine, the commoners will be able to get the benefits of the advanced information and communication technologies. In this scenario the acceptance and usability of the advances in information technology by the masses will be staggeringly increased. Moreover, 70% of the country's population lives in rural areas, so it becomes even more advantageous for them to have speech enabled computer applications accessible in their native languages. In the past decades, remarkable research has been done on isolated as well as continuous, large vocabulary speech processing and recognizing systems for English and other European languages; Indian languages as Hindi and other state languages were not being emphasized. So in this dissertation, a Hindi Speech Recognition system has been built and Gaussian Mixture Models (GMMs) is used to find the optimal number of Gaussian Mixtures that exhibits maximum accuracy for a small vocabulary of Hindi speech recognition system.

For the implementation work, we mainly used three speech processing and recognition tools like Audacity, Wavesurfer and HTK. For speech recording we used Audacity, for labeling the recorded audio files Wavesurfer is used, and HTK is a popular tool used for speech processing and recognition for handling HMMs and GMMs. As soon as the speaker

utters some word or phrase in the unidirectional mike or a microphone, the speech signal is captured and pre-processing is done at front-end for feature extraction, and evaluated at back-end using the GMM and hidden Markov model. In this statistical approach, since the evaluation of Gaussian likelihoods dominates the total computational load, the selection of appropriate number of Gaussian mixtures is very important. This selection of GMM depends upon the amount of training data provided. As to train Indian languages ASR system the small databases are available, the higher range of Gaussian mixtures (i.e. 64 and above), normally used for English or European languages, is not required for them. This thesis reviews the statistical framework and introduces an iterative procedure to select an optimum number of Gaussian mixtures that exhibits maximum accuracy for small database in Hindi speech recognition system. We have also varied the number of HMMs with varied number of GMMs and studied their effect on the speech recognition.

List of Figures

Figure 2.1 Components of Traditional Recognition Method.....	8
Figure 2.2 Architecture of ASR System	8
Figure 2.3 Recognition using Template Matching	12
Figure 3.1 Steps of Feature Extraction in Speech Recognition	13
Figure 3.2 Frame Based Feature Extraction in Speech Recognition	14
Figure 3.3 Block Diagram of Speech Analysis Procedure	15
Figure 3.4 Word Model for the Word “need”.....	16
Figure 3.5 A typical structure of a word based HMM.....	20
Figure 4.1 Waveform as viewed by wavesurfer	27
Figure 4.2 Labeled Speech waveform	28
Figure 4.3 Configuration file	28
Figure 4.4 Prototype for “jal” HMM with 2 gaussian mixtures	30
Figure 4.5 HMM training for “jal” HMM with 2 gaussian mixtures.....	31
Figure 4.6 Grammar file for connected words.....	31
Figure 4.7 Flowchart to convert speech signal to sequence of text	32
Figure 4.8 Recognized sentence based upon our transliteration for connected system	33
Figure 4.6.1 Recognition statistics for Gaussian 2 HMM6.....	33
Figure 4.6.2 Result for Isolated Word Recognition: HMM6.....	38
Figure 4.6.3 Result for Connected Word Recognition: HMM6	38

List of Tables

Table I Vowels with equivalent Matras	25
Table II Hindi Consonant Set	26
Table III Transcription for used vocabulary set.....	27
Table IV Result for GMM based HMM model with 6 states	34
Table V Result for GMM based HMM model with 7 states.....	35
Table VI Result for GMM based HMM model with 8 states	35
Table VII Result for GMM based HMM model with 9 states	36
Table VIII Result for GMM based HMM model with 10 states.....	36
Table IX Table of comparison	37

List of Abbreviations

AM	Acoustic Model
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
BP	Back-Propagation
BW	Baum Welch
C	Consonant
CD	Continuous Density
CSR	Continuous Speech Recognition
CUED	Cambridge University Engineering Department
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DMC	Discrete Model Combination
DT	Discriminative Training
DTW	Dynamic Time Warping
DWPT	Discrete Wavelet Packet Transform
FB	Filter Bank
FDLP	Frequency Domain Linear Prediction
FFNN	Feed Forward Neural Network
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HAT	Hidden Activation TRAP
HMM	Hidden Markov Model
HTK	Hidden Markov Model Tool Kit
IID	Independent and Identically Distributed
IPA	International Phonetic Alphabet
IWR	Isolated Word Recognition
LDA	Linear Discriminant Analysis
LM	Language Model

LME	Large Margin Estimation
LP	Linear Prediction
LPC	Linear Prediction Coefficient/Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficient
LSTF	Localized Spectro-Temporal Feature
LVCSR	Large Vocabulary Continuous Speech Recognition
MB	Multi Band
MCE	Minimum Classification Error
MFB	Mel Filter Bank
MFCC	Mel Frequency Cepstral Coefficient
ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
MMSE	Minimum Mean Square Error
MPE	Minimum Phone Error
MS	Multi Stream
MWE	Minimum Word Error
OFB	Optimized Filter Bank
OOV	Out Of Vocabulary
PDF	Probability Density Function
PLP	Perceptual Linear Prediction
RASTA	Relative Spectra
RNN	Recurrent Neural Networks
ROVER	Recognized Output Voting Error Reduction
SA	Speaker Adaptation
SD	Speaker Dependent
SI	Speaker Independent
SMC	Stochastic Model Combination
SME	Soft Margin Estimation
SSM	Stochastic Segment Model
STFT	Short Time Fourier Transform
SVM	Support Vector Machine

TDNN	Time Delay Neural Network
TRAP	Temporal Pattern
V	Vowel
VAD	Voice Activity Detector
VQ	Vector Quantization
VS	Vowel Sign
WER	Word Error Rate
WTN	Word Transition Network
ZCR	Zero Crossing Rate

TABLE OF CONTENTS

Certificate	ii
Declaration.....	iii
Acknowledgements	iv
Abstract	v
List of Figures.....	vii
List of Tables	viii
List of Abbreviations	ix
Chapter 1	1
Introduction.....	1
1.1 Overview of Automatic Speech Recognition	1
1.2 Motivation.....	1
1.3 Related Work	2
1.3.1 Early Attempts for Automatic Speech Recognizer	2
1.3.2 Improved Technology in 1980's.....	3
1.3.3 Milestones in Speech Recognition over the 1990's.....	4
1.3.4 Latest Technology Drivers.....	5
1.4 Problem Statement and Contribution.....	5
1.5 Dissertation Organization	6
Chapter 2	7
System Architecture for ASR	7
2.1 Automatic Speech Recognition.....	7
2.2 System Architecture for Speech Recognizer	8
2.2.1 Feature Extraction.....	9
2.2.2 Back End/ Training & Testing Mainly Covers	9
Chapter 3	13
Front End and Back End Analysis.....	13
3.1 Feature Extraction.....	13
3.2 Mel Frequency Cepstrum Coefficient (MFCC).....	15
3.3 Hidden Markov Model.....	15

3.4 Gaussian Mixture Model Definition	16
3.5 Gaussian Mixture Based HMM	19
3.6 Mixtures of Multivariate Gaussian	20
Chapter 4	22
Hindi Speech Recognition System Using HTK	22
4.1 Introduction.....	22
4.2 Related Work	23
4.3 Hidden Markov Model and HTK.....	24
4.4 Hindi Character Set.....	24
4.5 Implementation	26
4.5.1 System description.....	26
4.5.2 Data preparation.....	27
4.5.3 Front-end design	28
4.5.4 Back-end design.....	30
4.5.5 Performance evaluation	32
4.6 Results.....	33
Chapter 5	39
Conclusion and Future Directions.....	39
5.1 Conclusion	39
5.2 Suggestions for Future Research	39
References	41

Introduction

1.1 Overview of Automatic Speech Recognition

Automatic Speech Recognition (ASR) is a typical example of an automatic pattern classification problem. The aim of ASR is to find the most probable sequence of words (text) corresponding to a stream of acoustic information. ASR takes place in two phases, i.e. pattern training and pattern classification. It has various applications, such as voice command and control, dictation, dialog systems, audio indexing, speech-to-speech translation, etc. In the ASR system, variability due to different speakers, environment and channels degrade the performance of the system and are undesirable. It is desired that an ASR system is robust to these unwanted variability.

1.2 Motivation

Speech is the primary means of communication between humans and it would be ideal if we can use them for communicating with a computer. Speech recognition, for example, is a task of converting human's speech into text which enables us to talk to a computer. Speech recognition is a very popular research goal in the field of machine intelligence. There are many reasons for automatic speech recognition being widely developed by engineers and scientists around the world. Human-machine interaction is one of the most important reasons. We always dream of ordering machines such as the TV to turn itself on and change channels per our orders, thermostats to adjusting the temperature by them to adapt to a human's preferences, or even a robot babysitter to do all the house tasks fast and efficiently. The basic sensory stages of the human-machine interaction are vision recognition and speech recognition. Voice recognition, which is a special kind of speech recognition, is widely used in high security locations. Due to the high demand in the current market, many corporations have already built some Automatic Speech Recognition (ASR) systems: like the dictation system used by IBM and the telephone transaction system used by T-

Mobile, AT&T and Philips. Although these systems have been used in commercial area for years already, they still have many problems. First, these systems can only accomplish limited tasks such as recognizing numbers from 0 to 9, or isolated commands (e.g. transfer to customer service, balance request, pay bill, and etc.). Second, they all lack robustness, i.e. these systems have very poor performance in a noisy environment. In order to resolve the noise robustness, noise robust approach for ASR is explored in this dissertation. This approach is known as Multi Stream (MS) hybrid HMM/ANN. In Multi Stream systems, information from more than one source is combined to improve the performance, assuming that different sources considered for combination carry complementary information. In addition to this approach Multi Band (MB) hybrid HMM/ANN is also investigated in the dissertation. Multi Band processing which belongs to the generic class of Multi Stream processing techniques divides the entire frequency band into a set of frequency sub-bands and processes each band separately as an individual feature stream.

1.3 Related Work

Automatic speech recognition (ASR) is the task of transforming the intended message content of the speech into text with the help of a machine. Designing a machine that mimics human behavior, particularly the capability of speaking naturally and responding properly to spoken language, has intrigued engineers and scientists for centuries. Since the 1930s, when Homer Dudley of Bell Laboratories proposed a system model for speech analysis and synthesis [1,2], the problem of automatic speech recognition has been approached progressively, from a simple machine that responds to a small set of sounds to a sophisticated system that responds to fluently spoken natural language and takes into account the varying statistics of the language in which the speech is produced. The first speech recognizer was appeared in 1952 and consisting of a device for the recognition of single spoken digits[3].

1.3.1 Early Attempts for Automatic Speech Recognizer

Early attempts to design systems for automatic speech recognition were mostly guided by the theory of acoustic-phonetics, which describes the phonetic elements of speech (the basic sounds of the language) and tries to explain how they are acoustically realized in a spoken utterance. In 1952, Davis, Biddulph, and Balashek of Bell Laboratories built a system for isolated digit recognition for a single speaker [3], using the formant frequencies measured (or estimated) during vowel regions of

each digit. In the 1960's, several Japanese laboratories demonstrated their capability of building special purpose hardware to perform a speech recognition task. Most notable were the vowel recognizer of Suzuki and Nakata at the Radio Research Lab in Tokyo [4], the phoneme recognizer of Sakai and Doshita at Kyoto University [5]. Speech recognition, based on LPC methods, was proposed by Itakura [6], Rabiner and Levinson [7] and others. Today, most practical speech recognition systems are based on the statistical framework and results developed in the 1980's, with significant additional improvements in the 1990's.

1.3.2 Improved Technology in 1980's

The LPC models the input signal with constant weighting for the whole frequency range. However human perception does not have constant frequency perception in the whole frequency range. Thus LPC was not able to provide significant performance in speech recognition. MelScale was introduced by Davis and Mermelstein [8] in 1980. In MFCC the spectrum is warped according to the Mel Scale by taking human perception sensitivity with respect to frequencies into consideration.

Speech recognition research in the 1980's was characterized by a shift in methodology from the more intuitive template-based approach (a straightforward pattern recognition paradigm) towards a more rigorous statistical modeling framework. Hidden Markov Models (HMM) [9] are the most popular (parametric) model at the acoustic level. The hidden Markov model, which is a stochastic process, models the intrinsic variability of the speech signal (and the resulting spectral features) as well as the structure of spoken language in an integrated and consistent statistical modeling framework [10].

The formalism of the HMM is a probability measure that uses a Markov chain to represent the linguistic structure and a set of probability distributions to account for the variability in the acoustic realization of the sounds in the utterance. Given a set of known (text-labeled) utterances, representing a sufficient collection of the variations of the words of interest (called a training set), one can use an efficient estimation method, called the Baum-Welch (BW) algorithm to obtain the "best" set of parameters that define the corresponding model or models.

But HMM suffers from some major limitations [11] which led to the idea of Artificial Neural Network (ANN) in the late 1980's. Neural networks, such as the Multilayer Feed-forward Networks (MLPs) or the Recurrent Neural Networks (RNN) can be trained to associate unknown input data to

learned words. To consider the temporal relationships of speech signal, time delay neural network (TDNN) and recurrent neural networks (RNN) have been proposed [12]. These neural networks are usually trained by using Back propagation algorithm [13].

1.3.3 Milestones in Speech Recognition over the 1990's

In the 1990's, a number of innovations took place in the field of speech recognition. The Perceptual Linear Prediction (PLP) method proposed by Wheatley and Picone (1991) demonstrated a further resolution of the critical band, equal loudness curve adjustment and application of intensity-loudness power law [14]. In the noisy environment PLP was not able to provide significant performance as noise in the signal cause mismatch between the trained and testing data due to which performance of the system degrades. In fact, Hermansky and Morgan (1994) showed that the reduction of this irrelevant information in the parametric representation of speech signals significantly improves the performance of the recognition system.

The Relative Spectral Technique (RASTA) [15] is one of the pioneering techniques developed in this context. As an alternative to spectrum-based feature vectors as discussed above we can use Temporal Patterns (TRAPs) for phonetic classification. In this technique, we substitute a conventional spectral feature vector in ASR by a 1 sec long temporal vector [16] because the information about the underlying sub-word classes (phonemes) spreads at least over the interval 200-300 ms as demonstrated by Bilmes [17]. Early attempts at using neural networks for speech recognition centered on simple tasks like recognizing a few phonemes or a few words (e.g., isolated digits), with good success [18]. However, as the problem of speech recognition inevitably requires handling of temporal variation, neural networks improvement over the LPC which takes advantage of three principal characteristics derived from the psychoacoustic properties of the human hearing viz., spectral in their original form have not proven to be extensible to this task. On-going research focuses on integrating neural networks with the essential structure of a hidden Markov model to take advantage of the temporal handling capability of the HMM. Bourlard et al. [19] proposed HMM/ANN hybrids for continuous ASR in which a MLP was trained to estimate the posterior probabilities of HMM states, with the ultimate objective of maximizing the posterior probability of a given (left-to-right) Markov model M_i given an acoustic observation sequence X .

1.3.4 Latest Technology Drivers

We can extend our feature extraction technique [20] for Tandem processing in ASR. The TANDEM refers to a way of converting the frequency-localized evidence to features for the HMM-based ASR system. It has been observed that combining evidence from more than one source improves the performance of ASR systems if the different sources carry complementary information and are given importance according to their reliability [21]. Multi Stream combination is one of the ways to improve the robustness of a system. A Multi Stream system is a step towards fail-safe systems from an engineering perspective as well. In the case of failure of one of the streams, the system can still work, perhaps with a reduced performance. The underlying principle of Multi Stream combination is to obtain a better estimate of the optimal decision rule by combining outputs of several classifiers having complementary source of information [22]. Multi Band speech recognition, a special case of Multi Stream combination, has been studied in detail in the recent past, leading to some important contributions to the field of ASR. The approach of Multi Band ASR was motivated by Fletcher's product of errors rule. In Multi Band approaches, the full-band spectrum is divided into sub-bands and separate models are trained for features extracted from each sub-band. Multi Band combination was found useful in the case of band limited noise but for wide-band noise, the scheme often failed to perform better than a full-band system and the combination led to a degraded performance. In most of the studies, 4 sub-bands defined by critical bands were used for band division. Increasing the number of sub-bands reduced the information content in each sub-band and the performance of individual sub-bands degraded. However, when the outputs of the sub-band models were combined, no significant difference was observed between the systems using 4 and 7 sub-bands. R. K. Aggarwal and M. Dave also published some papers which are milestones in the field of Gaussian Mixture optimization and feature extraction for Hindi speech recognition [23,24].

1.4 Problem Statement and Contribution

The main objective of the dissertation is to implement speech recognition system for Hindi language that can recognize a connected sequence of words from the predefined set of vocabulary. The dissertation also aims at providing a theoretically complete description of statistical techniques used in speech recognition.

The specific contributions of the dissertation may be summarized as follows:

- A review of the reported research work on speech recognition approaches has been performed. From the performed review, some important open issues have been identified and observations have been taken.
- A theoretical complete description of statistical techniques used for speech recognition has been presented.
- A Hindi speech recognition system for isolated words with fixed HMMs and varying Gaussian has been implemented using HTK that will recognize words from the set of vocabulary for which system has been trained.
- A Hindi speech recognition system for connected words with fixed HMMs and varying Gaussian has been implemented using HTK that will recognize a continuous sequence of words from the set of vocabulary for which system has been trained.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows:

Chapter 2: System Architecture of ASR

This chapter describes the basics of Automatic Speech Recognition. In this chapter, system architecture for speech recognizer is discussed in detail.

Chapter 3: Front End and Back End Analysis

This chapter describes the term “feature extraction” in detail. It introduces the feature extraction method (MFCC) in detail. HMM and GMM based HMM are also discussed in this chapter

Chapter 4: Hindi Speech Recognition System Using HTK

This chapter presents that how HMM based speech recognition system is simulated using the HTK. In this chapter, we evaluate our system for isolated and connected speech recognition. We also studied the effect of varying GMM and HMM and their effect on the accuracy of our system.

Chapter 5: Conclusions and Future Directions

In this chapter, we have concluded this dissertation with our research contributions and discussion about future research directions.

System Architecture for ASR

2.1 Automatic Speech Recognition

Automatic Speech Recognition is the process by which a machine finds the most probable sequence of words (text) corresponding to the speech utterance [25]. The machine takes human utterance as an input and returns a string of words or phrases (in the form of text) as output. To resolve the speech recognition problem, it can be divided into two categories [26]:

- Isolated Word Recognition (IWR)
- Continuous Speech Recognition (CSR)

In IWR, the recognizer takes an observation sequence of any utterance at a time (spoken in isolation and belonging a fixed dictionary) as input and outputs the word which has the highest probability corresponding to that observation sequence [27]. In these systems, the speech containing the words can be easily isolated and recognized. But, speech communication used in the real world has a different pattern, where the word boundaries are not so clearly defined. Moreover, the segmentation of the speech signal into different words is difficult and sometimes considered impossible too. The frame work of IWR cannot handle these complexities and statistical techniques need to be employed. These systems fall into the realm of CSR systems. It is important to note that even though IWR has a very limited recognition power; it has many applications in the real world. The traditional speech recognition method comprises of two steps: acoustic signal processing and recognition [25] as shown in Figure 2.1.

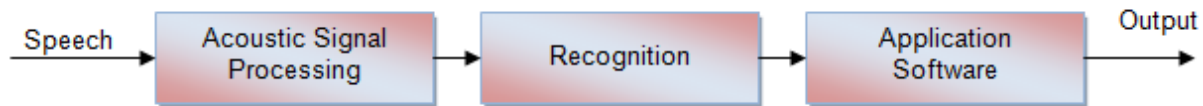


Figure 2.1 Components of Traditional Recognition Method

Pre-processing: Converts the spoken input into a form that a recognizer can process.

Recognition: Identifies what has been said and sends the recognized input to the software/hardware system that needs it.

Data for Speech Recognition

In speech recognition we concentrate on different types of sounds. The basic unit of these sounds is phoneme which is represented as phoneme linguistically. Phonemes are used to distinguish words.

2.2 System Architecture for Speech Recognizer

The basic model of ASR system is divided into two ends [26], front end and back end as shown in Figure 2.2.

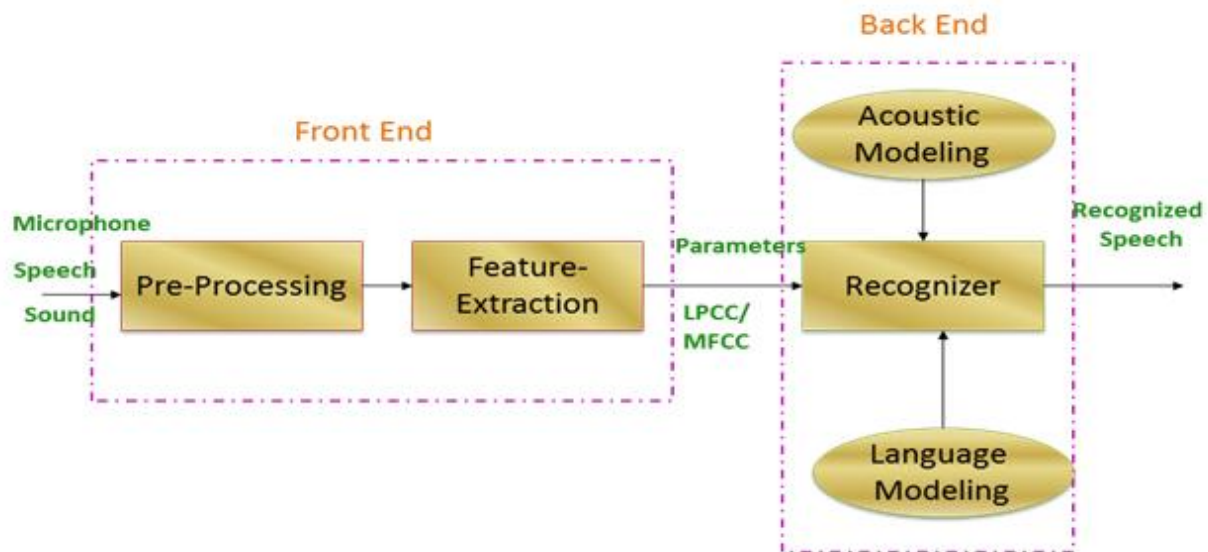


Figure 2.2 Architecture of ASR System

2.2.1 Feature Extraction

The goal of feature extraction is to find a set of properties of an utterance that have acoustic correlation to the speech signal, that is, parameters that can somehow, be computed or estimated through processing of the signal waveform. Such properties are termed as features. (Detailed discussion will be made in chapter3)

Some of them are:

- Linear Predictive Cepstral Coefficients(LPCC)
 - Based on properties of vocal tract/speech production system.
- Mel Frequency Cepstral Coefficients(MFCC)
 - Based on properties of the human auditory system.

2.2.2 Back End/ Training & Testing Mainly Covers

- Acoustic Modeling (HMM)
- Language Modeling
- Recognition

Acoustic Modeling

In this subsystem, the connection between the acoustic information and phonetics is established. The connection can be established either at word or at phoneme level giving rise to word based system and phoneme based system, respectively. There are many models for this purpose like

- i. Hidden Markov Model(HMM)
- ii. Artificial Neural Network(ANN)s
- iii. Support Vector Machine(SVM)
- iv. Fuzzy Logic

Hidden Markov Model

It is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters. This algorithm is often used due to its simplicity and feasibility of use. Hidden Markov models (HMM) are the most popular (parametric) model at the acoustic level (explained in later chapters).

Artificial Neural Network

Automatic (machine) recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, artificial intelligence, and remote sensing. A pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. Given a pattern, its recognition/classification may consist of one of the following two tasks: 1) supervised classification (e.g., discriminant analysis) in which the input pattern is identified as a member of a predefined class, 2) unsupervised classification (e.g., clustering) in which the pattern is assigned to a hitherto unknown class. The recognition problem here is being posed as a classification or categorization task, where the classes are either defined by the system designer (in supervised classification) or are learned based on the similarity of patterns (in unsupervised classification).

Support Vector Machine

Hidden Markov Models (HMMs) are, undoubtedly, the most employed core technique for Automatic Speech Recognition (ASR). During the last decades, research in HMMs for ASR has brought about significant advances and, consequently, the HMMs are currently accurately tuned for this application. Nevertheless, we are still far from achieving high-performance ASR systems. Some alternative approaches, most of them based on Artificial Neural Networks (ANNs), were proposed during the last decade. Some of them tackled the ASR problem using predictive ANNs, while others proposed hybrid (HMM-ANN). However, despite some achievements, none of these approaches could outperform the results obtained with HMMs and, nowadays, the preponderance of Markov Models is a fact.

Fuzzy Logic

Fuzzy logic, proposed by Lotfi A. Zadeh in 1965, is a theory that allows natural linguistic problems to be solved rather than using numerical values. Fuzzy logic formalizes and operates the same way human arguing does. It's based on the mathematical theory of the fuzzy groups. This theory is a classic extension of the theory for the consideration of groups defined in an indistinct way. Furthermore, Fuzzy Logic was conceived as a reliable method for sorting and handling data but has proven, also, to be an excellent choice for many control system applications since it mimics human control logic. According to classical logic, decisions are binary having two-valued logic:

true or false. This point distinguishes fuzzy logic from classical logic. In fuzzy logic, a decision can be true and false with a degree of membership in each of these two cases. Fuzzy logic describes uncertain and imprecise declaration using that each element belongs partially or gradually to defined sets. For example, the statement "Today, it is a nice day" according to fuzzy logic, is 100% true if there are no clouds, 80% true if there are a few clouds, 50% true if there are a lots of clouds and 0% true if it rains all day. To conclude, in fuzzy logic an element belongs to a “fuzzy” set (not strictly to one set).

Language Modeling

A language model contains the structural constraints available in the language to generate the probabilities of occurrence. Intuitively speaking, it determines the probability of a word occurring after a word sequence. It is easy to see that each language has its own constraints for validity. The method and complexity of modeling language would vary with the speech application. For example, a simple speech enabled call-dialing system which would have a very limited vocabulary and constrained input will have a simple language model. On the other hand, the task of transcribing broadcast news data would require a large vocabulary of the order of thousands with sentence structure that is much less constrained. This gives rise to mainly two approaches for language modeling as described below. The appropriateness of the approach is problem specific. Generally, small vocabulary constrained tasks like call-dialing can be modeled by grammar based approach where as large applications like broadcast news transcription require stochastic approach.

Recognition

Once thepre-processing of a user’s input is complete the recognizer is ready to perform its primary function i.e. to identify what the user has said. Thecompeting recognition technologies found in commercial speech recognition systems are:

- Template Matching
- Stochastic Processing

These approaches differ in speed, accuracy and requirements.

Template Matching

Template matching is a form of pattern recognition. It represents speech data as sets of feature/parameter vectors called templates [25]. Each word or phrase in an application is stored as

separate template. Spoken input by end users is organised in to templates prior to performing the recognition process. The input is the compared with stored templates as shown in Figure 2.5.

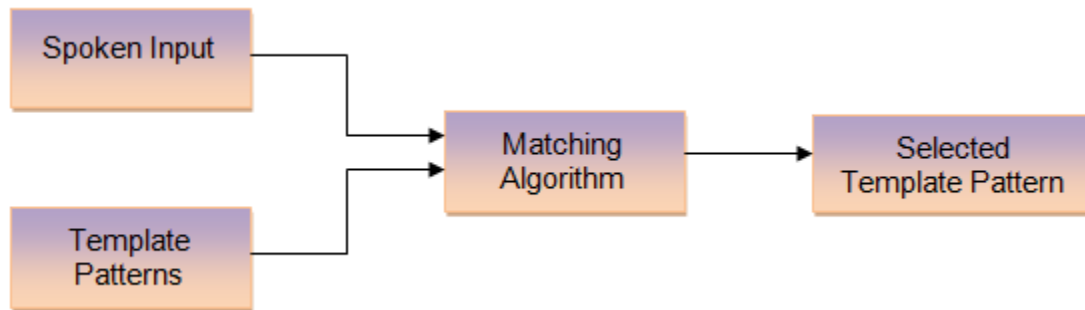


Figure 2.3 Recognition using Template Matching

The stored template most closely matching the incoming speech pattern is identified as the input word or the phrase. The selected template is performed at word level and contains no reference to phonemes within the word.

Stochastic Processing

The ASR problem can be formulated as a statistical classification problem, according to classical pattern recognition. Once the classes have been defined as sequences W of allowable words from a “closed” dictionary, a parametric representation of the speech signal has been chosen (e.g. a sequence of acoustic feature vectors X), and a Maximum a Posteriori (MAP) criterion has been adopted, the classification problem can then be stated as finding the sequence of words W which maximizes the quantity $Pr(W|X)$ [30]. The latter is usually factorized using Bayes' theorem as:

$$Pr(W|X) = \frac{Pr(X|W)Pr(W)}{Pr(X)}$$

Given an acoustic observation sequence X , the efforts on the maximization of $Pr(W|X)$ can be moved to the search for the class W which maximizes the numerator of the right-hand side of equation i.e. $Pr(X|W)Pr(W)$. The quantity $Pr(W)$, usually referred to as the **Language Model (LM)** depends on high-level constraints and linguistic knowledge about allowed word strings for the specific task. The quantity $Pr(X|W)$ is known as the **Acoustic Model (AM)**. It describes the statistics of sequences of parameterized acoustic observations in the feature space given the corresponding uttered words (e.g. certain phonemes).

Front End and Back End Analysis

3.1 Feature Extraction

One of the first decisions in any pattern recognition system is the choice of what features can be used and how exactly to represent the basic signal that is to be classified, in order to make the classification task easiest. Speech recognition is a typical example of pattern recognition system. The feature extraction process [31] is expected to discard irrelevant information of the task while keeping the useful one as depicted in Figure 3.1. The following properties are required for a good feature extractor:

- Compact features to enable real time analysis
- Minimize the loss of Discriminant information

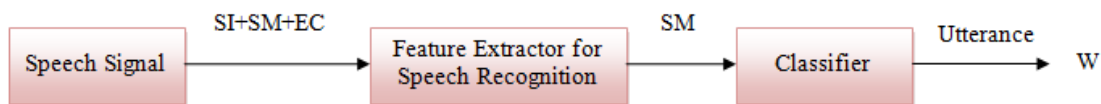


Figure 3.1 Steps of Feature Extraction in Speech Recognition

A Feature Extractor (FE) for ASR performs a specific task. As shown in figure1, the speech signal contains the characteristic information of the speaker (SI) and Environment (EC) in addition to Signal Message(SM). An FE for speech recognition needs to maximally discard the SI and EC information and only allow the SM information to filter from the speech signal. The ability of FE for speech recognition improves depending upon how well SI and EC are filtered out:

- SI \Rightarrow Speaker Independence
- EC \Rightarrow Noise Robustness

The speech signal is processed in frames with frame size ranging from 15 to 25 milliseconds and an overlap of 50%-70% between consecutive frames as shown in Figure 3.2. Hence, the speech

is processed on a frame by frame basis. The overlap between two consecutive frames is necessary in order to account for the possibility of a split of an acoustic unit. Broadly, the feature extraction techniques are classified as temporal analysis and spectral analysis techniques. In temporal analysis the speech waveform itself is used for analysis. In spectral analysis spectral representation of speech signal is used for analysis. Through more than 30 years of recognizer’s research, many different feature extractions of the speech signal have been suggested and tried. The most popular feature representation currently used is the MEL Frequency Perceptual Linear Prediction (MF-PLP). Another popular speech feature representation is known as Relative Spectral Perceptual Linear Prediction (RASTA-PLP).

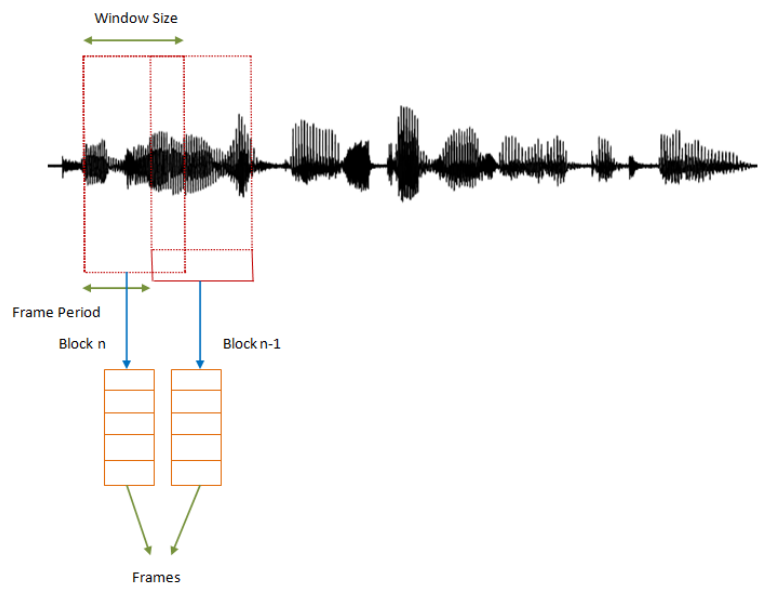


Figure 3.2 Frame Based Feature Extraction in Speech Recognition

Mel scale Cepstral analysis is very similar to perceptual linear predictive analysis of speech, where the short term spectrum is modified based on psychophysically based spectral transformations. In this method, however, the spectrum is warped according to the Mel Scale, whereas in PLP the spectrum is warped according to the Bark Scale. Both MFCC and PLP take human perception sensitivity with respect to frequencies into consideration, and therefore are best for speech/speaker recognition.

3.2 Mel Frequency Cepstrum Coefficient (MFCC)

In a physical or communication theory sense, signal processing in the ear can be described by a succession of several signal processing stages that model the “effective” processing in the auditory system without taking the actual biological implementation into account. An important element of such a functional description is a filter bank as a first stage that distributes the sound according to its frequency into different frequency bands and hence mimics some aspects of the basilar membrane, as an example in MFCC the spectrum is warped according to the Mel Scale by taking human perception sensitivity with respect to frequencies into consideration. We shall explain the step-by-step computation of MFCC [32] in this section as depicted by Figure 3.3.

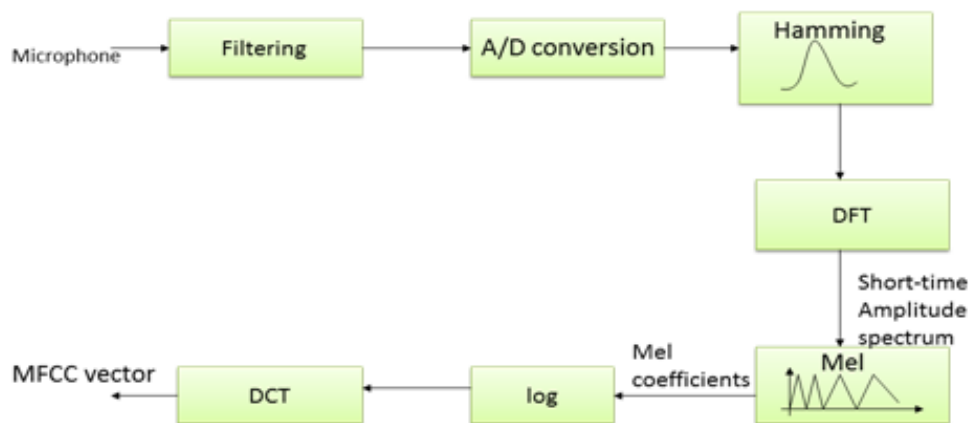


Figure 3.3 Block Diagram of Speech Analysis Procedure

3.3 Hidden Markov Model

Hidden Markov models (HMMs) are stochastic models widely used in speech recognition during recent years. The number of states in classical HMMs is usually predefined and fixed during training, and may be quite different from the real number of hidden states of the signal source. The underlying assumption of the HMM is that the speech can be well characterized as a parametric random process and that the parameters of the stochastic process can be determined in a precise, well-defined manner. HMMs are the natural extension to the markov chain that produces output observation symbols in any given state [34,35]. Therefore, the observation is a probabilistic function of the state. For a given observation sequence, the state sequence is not observable and

therefore hidden. This is why the word hidden is placed before Markov Models. Formally, a Hidden markov model is defined as $\lambda (S, M, A, B, \pi)$ where,

- **S**: Set of states $S = S_1, S_2, \dots, S_n$
- **M**: Number of distinct observation symbols per states
Individual symbols are denoted by $V = \{v_1, v_2, \dots, v_k\}$.
- **A**: a_{ij} : State Transition Probability
Each a_{ij} represents the probability of transitioning from state S_i to S_j .

$$a_{ij} = P(T_{t+1} = S_j | T_t = S_i)$$

- **B**: $b_j(k)$: Emission Probability or Observation Symbol Probability distribution
 $b_j(k) = P(v_k | T_t = S_j)$
- π : Initial State Distribution: the probability that S_i is a start state

Given the observation sequence $O = o_1, o_2, \dots, o_T$ and an HMM model $\lambda = (A, B, \pi)$, we compute the probability of O given the model i.e. $P(O|\lambda)$ as depicted by Figure 3.6. Unfortunately HMM suffers from some major limitations too. One major limitation of conventional HMM is that it does not provide an adequate representation of the temporal structure of speech. Secondly, HMM relies on first order Markov assumption, following which the duration of each stationary segment captured by single state is inadequately modeled. Finally, because of conditional independence assumption, all observation frames are dependent only on the state that generated them, and not on neighboring observation frames, which makes it hard to handle non-stationary strongly correlated frames [11]

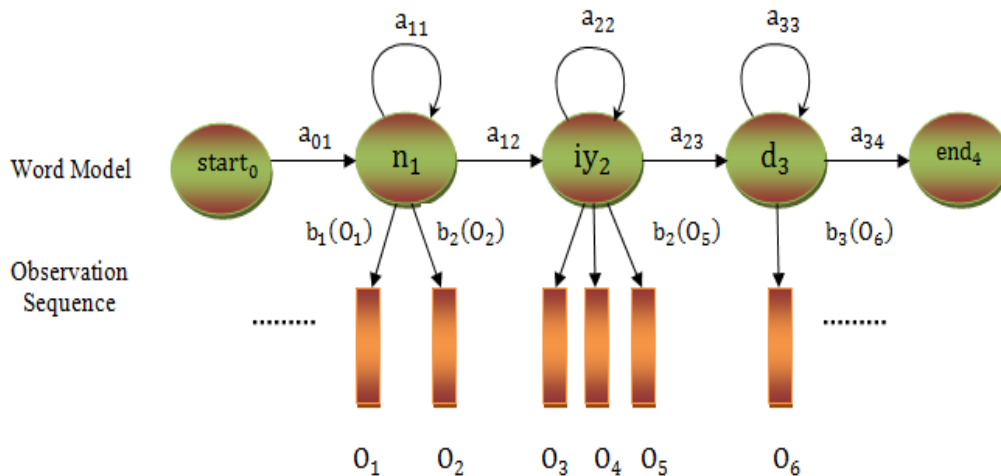


Figure 3.4 Word Model for the Word “need”

3.4 Gaussian Mixture Model Definition

Before understanding the GMMs, basic knowledge of Covariance and correlation Matrix is required, so firstly these are explained and after that their use to GMM is mentioned.

Covariance

In probability theory and statistics, covariance is a measure of how much two random variables change together. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the smaller values, i.e., the variables tend to show similar behavior, the covariance is positive. In the opposite case, when the greater values of one variable mainly correspond to the smaller values of the other, i.e., the variables tend to show opposite behavior, the covariance is negative. The sign of the covariance therefore shows the tendency in the linear relationship between the variables. The magnitude of the covariance is not easy to interpret. The normalized version of the covariance, the correlation coefficient, however, shows by its magnitude the strength of the linear relation.

Covariance matrix

In probability theory and statistics, a covariance matrix (also known as dispersion matrix or variance–covariance matrix) is a matrix whose element in the i, j position is the covariance between the i^{th} and j^{th} elements of a random vector (that is, of a vector of random variables). Each element of the vector is a scalar random variable, either with a finite number of observed empirical values or with a finite or infinite number of potential values specified by a theoretical joint probability distribution of all the random variables.

Intuitively, the covariance matrix generalizes the notion of variance to multiple dimensions. As an example, the variation in a collection of random points in two-dimensional space cannot be characterized fully by a single number, nor would the variances in the x and y directions contain all of the necessary information; a 2×2 matrix would be necessary to fully characterize the two-dimensional variation.

Correlation

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people.

Correlation matrix

The correlation matrix of n random variables $X_1 \dots X_n$ is the $n \times n$ matrix whose i, j entry is $\text{corr}(X_i, X_j)$. If the measures of correlation used are product-moment coefficients, the correlation matrix is the same as the covariance matrix of the standardized random variables $X_i / \sigma(X_i)$ for $i = 1, \dots, n$. This applies to both the matrix of population correlations (in which case " σ " is the population standard deviation), and to the matrix of sample correlations (in which case " σ " denotes the sample standard deviation). Consequently, each is necessarily a positive-semi definite matrix. The correlation matrix is symmetric because the correlation between X_i and X_j is the same as the correlation between X_j and X_i .

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system.

GMMs are used in speaker recognition systems, due to their capability of representing a large class of sample distributions. One of the powerful attributes of the GMM is its ability to form smooth approximations to arbitrarily shaped densities. The classical uni-modal Gaussian model represents feature distributions by a position (mean vector) and an elliptic shape (covariance matrix) and a vector quantizer (VQ) or nearest neighbor mode represents a distribution by a discrete set of characteristic templates. A GMM acts as a hybrid between these two models by using a discrete set of Gaussian functions, each with their own mean and covariance matrix, to allow a better modeling capability.

3.5 Gaussian Mixture Based HMM

Speech, a unique medium of communication is produced by air pressure waves emanating from the mouth and nostrils of a speaker. When we speak, our articulatory apparatus (like lips, jaw, tongue and velum) modulates the air pressure and flow to produce an audible sequence of sounds. Due to the physical constraints, the articulatory organs cannot move with drastic changes and this characteristic of speech production is very useful for acoustic modeling. During a short time interval (typically 10-20 ms) while the articulatory configuration stays relatively constant, speech signals can be termed as quasi-stationary. Statistical framework of continuous density hidden Markov models is a popular choice for modeling these time varying quasi-stationary speech signals.

HMM is a doubly stochastic process generated by two interrelated mechanisms, an underlying Markov chain having a finite number of states, i.e. $\{q_1, q_2, \dots, q_N\}$, and a set of random functions, one of which is associated with each state to describe output symbols. First process is based on the transitions among the states that are governed by a set of probabilities called transition probabilities to model the temporal structure of speech. Second process is based on the state output observations, $O = \{o_1, o_2, \dots, o_T\}$, governed by multivariate Gaussian mixture distributions, $b_j(o_t), 1 \leq j \leq N$, to model the spectral variability of speech[36,37]. At each discrete instance of time, one process is assumed to be in some state and an observation is generated by the other process corresponding to the current state. The underlying Markov chain then changes states according to its transition probability matrix, $A = [a_{ij}]$, $1 \leq i, j \leq N$ and a_{ij} is the probability of making a transition from state i to state j denoted as $a_{ij} = P[q_{t+1} = j | q_t = i]$. When symbols are generated at the time of entering to the next state, it is known as state-output HMM and when symbols are generated at the time of transition between states, it is known as edge-output HMM. Usually the model outputs are associated with states rather than with the transition of the states. A typical structure of a word based HMM is shown in Fig.3.7.

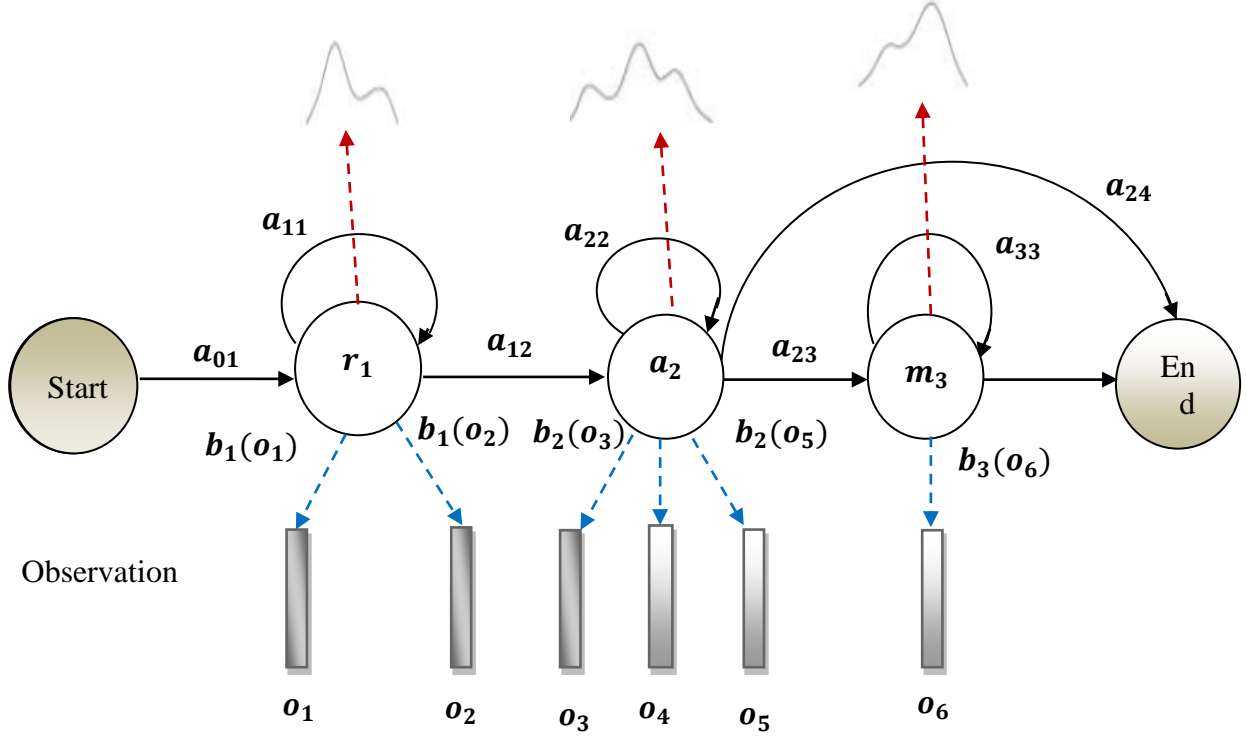


Figure 3.5 A typical structure of a word based HMM

3.6 Mixtures of Multivariate Gaussian

To model the complex speech signal, mixtures of Gaussian have been used as emission pdfs in the hidden Markov models. In such systems, the output likelihood of a HMM state S for a given observation vector, X_n can be represented as a weighted sum of probabilities:

$$P(X_n|S) = \sum_{k=1}^K w_k P_k(X_n) \quad (1)$$

Where, parameters of the state pdf are number of mixtures K ; their weighing factors, w_k which satisfies $w_k > 0$ and $\sum_{k=1}^K w_k = 1$; mean vector μ_k and the variance covariance matrix Σ_k of the k^{th} mixture component. Each mixture component belongs to a D -dimensional multivariate Gaussian density function defined as:

$$p_k(X_n) = \frac{1}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \exp \left[-\frac{(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)}{2} \right] \quad (2)$$

In practice the full covariance matrices are reduced to diagonal covariance due to computational and data sparseness reasons [38]. By substituting the value of probabilities defined in Equation (2), the state model in Equation (1), becomes the Gaussian Mixture model (GMM) defined as:

$$p(X_n|S) = \sum_{k=1}^K Z_k \exp \left[-\frac{1}{2} \sum_{q=1}^D \frac{(x_{nq} - \mu_{kq})^2}{\sigma_{kq}^2} \right] \quad (3)$$

Where Z_k is a constant for each Gaussian i.e.,

$$Z_k = \frac{w_k}{(2\pi)^{\frac{D}{2}} (\prod_{q=1}^D \sigma_{kq}^2)^{1/2}} \quad (4)$$

In order to compute efficiently and to avoid underflow, probabilities are computed in log domain. Therefore the log likelihood can be expressed as:

$$\log p(X_n|S) = \log \text{add}_{k=1}^K \left[\log(Z_k) - \frac{1}{2} \sum_{q=1}^D \frac{(x_{nq} - \mu_{kq})^2}{\sigma_{kq}^2} \right] \quad (5)$$

Here, the function $\log \text{add} []$ is defined as follows:

$$\log \text{add}_{k=1}^K [y_k] = \log \left[\sum_{k=1}^K \exp(y_k) \right] \quad (6)$$

In a typical HMM-based LVCSR system, the number of model states ranges from 2000 to 6000, each of which is a weighted sum of typically 8-64 multidimensional Gaussian distributions as in Equation (3). For each input frame, the output likelihoods should be evaluated against each active state [38]. Therefore, the state likelihoods estimation is computationally intensive and takes about 30-70% of the total recognition time. This kind of likelihood-based statistical acoustic decoding is so time consuming that it is one of the most important reasons why the recognition is slow. Some LVCSR systems might even decode speech several times slower than real time; that is to say, these systems are not practical for most spontaneous applications, such as man-machine dialogue. Therefore, it is necessary to develop efficient techniques in order to reduce the time consumption of likelihood computation without a significant degradation of recognition accuracy.

Hindi Speech Recognition System Using HTK

Speech recognition is the process of converting an acoustic waveform into the text similar to the information being conveyed by the speaker. In the present era, mainly hidden Markov model (HMMs) based speech recognizers are used. This chapter aims to build a speech recognition system for Hindi language. Hidden Markov model toolkit (HTK) is used to develop the system. It recognizes the isolated words using acoustic word model. The system is trained for 51 Hindi words.

4.1 Introduction

Speech interfacing provides a convenient and user friendly way of man-machine communication. Their absence makes the man-machine interaction obsolete. Unadventurously, transfer of information between man and machine is carried out via keyboards, pointing devices like mouse, touchpad for input and visual display units, monitors, plotters or printers for output [38]. However, one cannot reassure their usage as they require certain amount of skills for their use. Their usage is time consuming too. On the other hand, speech interfacing offers high bandwidth information and relative ease of use. It also permits the user's hands and eyes to be busy with a task, which is particularly valuable when the user is in motion or in natural field settings [37]. One can speak hastily instead of typing. Similarly speech output is more impressive and understandable than the text output.

Speech interfacing involves speech synthesis and speech recognition. Speech synthesis is the process of converting the text input into the corresponding speech output i.e. it act as text to speech converter. Conversely, Speech recognition is the way of converting the spoken sound into the text similar to the information being conveyed by sounds.

This chapter aims to develop and implement an isolated word speech recognition system for Hindi language. Hidden Markov model (HMM) is used to train and recognize the speech that uses MFCC to extract the features from the speech utterances. To accomplish this, hidden Markov model toolkit (HTK) [39,40] designed for speech recognition is used. HTK is developed in 1989 by Steve Young at the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED).

4.2 Related Work

In the past decade, much work has been done in the field of speech recognition for Hindi language. Among others, some of the reported works performed in the literature are [41, 42].

Pruthi et al. (2000) [42] have developed a speaker-dependent, real-time, isolated word recognizer for Hindi. Developed System uses a standard implementation. Linear predictive cepstral coefficients are used for feature extraction and recognition is carried out using HMM. System was designed for two male speakers. The recognition vocabulary consists of Hindi digits (0, pronounced as “shoonya” to 9, pronounced as “nau”).

An isolated word speech recognition system for Hindi language is designed by Gupta (2006) [41]. System uses continuous HMM and consists of word based acoustic. Recognition system uses this word model while recognizing. Again the word vocabulary contains Hindi digits. Recognizer gives good results when tested for sound used for training the model. For other sounds too, the results are satisfactory.

LVCSR systems are usually based on continuous density HMMs, which are typically implemented using Gaussian mixture distributions. Such statistical modeling systems tend to operate slower than real-time, largely because of the heavy computational overhead of the likelihood evaluation [43].

Gales et al, [44] basically this paper investigates the use of Gaussian Selection (GS) to increase the speed of a large vocabulary speech recognition system. Typically 30-70% of the computational time of a continuous density HMM-based speech recognizer is spent calculating probabilities. The aim of GS is to reduce this load by selecting the subset of Gaussian component likelihoods that should be computed given a particular input vector.

4.3 Hidden Markov Model and HTK

Hidden Markov model (HMM) [34] is a doubly stochastic process with one that is not directly observable. This hidden stochastic process can be observed only through another set of stochastic processes that can produce the observation sequence. HMMs are the so far most widely used acoustic models. The reason is just it provides better performance than other methods. HMMs are widely used for both training and recognition of speech system.

HMM are statistical frameworks, based on the Markov chain with unknown parameters. Hidden Markov model is a system which consists of nodes representing hidden states [45]. The nodes are interconnected by links which describes the conditional transition probabilities between the states. Each hidden state has an associated set of probabilities of emitting particular visible states.

HTK is a toolkit for building hidden Markov models. It is an open source set of modules written in ANSI C which deal with speech recognition. HTK mainly runs on the Linux platform. However, to run it on Windows, interfacing package Cygwin [46] is used.

In order to install the HTK on Ubuntu platform, the first step is to configure the HTK with respect to the operating system. Linux command *./configure* is used for this purpose. Once the HTK is configured, *make all* command is used to compile the separate source-code files and then to generate a final program. Finally, Linux command *make install* is used to create the installation directories, executable files, configuration files, libraries, documentation, various data, and so on.

4.4 Hindi Character Set

Hindi is mostly written in a script called Nagari or Devanagari which is phonetic in nature. Hindi sounds are broadly classified as the vowels and consonants.

Vowels:

In Hindi, there is separate symbol for each vowel. There are 12 vowels in Hindi language. The consonants themselves have an implicit vowel + (ॐ) to indicate a vowel sound other than the implicit one (i.e. ॐ) a vowel-sign (Matra) is attached to the consonant. The vowels with equivalent Matras are given in table 4.1.

Table I: Vowels with equivalent Matras

Vowel	अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	ऋ	ॠ
Matra	-	ा	ि	ी	ु	ू	े	ै	ो	ौ	ृ	ॄ

Consonants:

The consonant set in Hindi is divided into different categories according to the place and manner of articulation. There are divided into 5 Vargs (Groups) and 9 non-Varg consonants. Each Varg contains 5 consonants, the last of which is a nasal one. The first four consonants of each Varg, constitute the primary and secondary pair. The primary consonants are unvoiced whereas secondary consonants are voiced sounds. The second consonant of each pair is the aspirated counterpart of the first one. Thus four consonants of each Vargs are [unvoiced], [unvoiced, aspirated], [voiced], [voiced, aspirated] respectively. Remaining 9 non Varg consonants are divided as 5 semivowels, 3 sibilants and 1 aspirate [47]. The complete Hindi consonant set with their phonetic property is given in table 4.2.

Other Characters:

Apart from consonants and vowels, there are some other characters used in Hindi language are: anuswar (ं), visarga (ः), chanderbindu (ँ), क्ष, व्र, ज्ञ. Anuswar indicates the nasal consonant sounds. Anuswar sound depends upon the character following it. Depending upon the Varg of following character, sound wise it represents the nasal consonants of that Vargs.

Tables II: Hindi Consonant Set

Phonetic Property (Category)	Primary Consonants (Unvoiced)		Secondary Consonants (Voiced)		Nasal
	Unaspirated	Aspirated	Unaspirated	Aspirated	
Gutturals(कवर्ग)	क	ख	ग	घ	ङ
Patatals(चवर्ग)	च	छ	ज	झ	ञ
Cerebrals(टवर्ग)	ट	ठ	ड	ढ	ण
Dentals(तवर्ग)	त	थ	द	ध	न
Labials(पवर्ग)	प	फ	ब	भ	म
Semivowels	य, र, ल, व				
Sibilants	श, ष, स				
Aspirate	ह				

4.5 Implementation

In this section, implementation of the speech system based upon the developed system architecture has been presented.

4.5.1 System description

Hindi speech recognition system is developed using HTK toolkit on the Linux platform. HTKv3.4 and ubuntu10.04 are used. Firstly, the HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. The observation function that is being used in traditional HMM model is Single-gaussian observation function, with diagonal matrices. In this work, we are introducing an iterative procedure to select an optimum number of Gaussian mixtures by using different number of Gaussian mixtures in observation probability function. Secondly, unknown utterances are transcribed using the HTK recognition tools like HVite. System is trained for 51 Hindi words. Word model designed using GMM is used to recognize the speech.



Figure 4.1 Waveform as viewed by wavesurfer

4.5.2 Data preparation

Training and testing a speech recognition system needs a collection of utterances. System uses a data set of 51 words as given in column 1 and 4 of table 4.3. The data has been recorded using unidirectional microphone. Recording of the speech-sounds were done using the *Audacity*. The recorded waveform can be viewed using the *Wavesurfer* as shown in figure 4.1.

Distance of approximately 5-10 cm has been used between mouth of the speaker and microphone. Recording has been performed at room environment. Sounds were recorded at a sampling rate of 16000 Hz. Voices of eight people (5 male and 3 female) have been used to train the system. Each one was asked to utter each word four times. Thus giving a total of 1632 ((8*4)*51) speech files. Speech files have been stored in *.wav* format. Table III shows the ourtransliteration corresponding to the vocabulary words.

Table III Transcription for used vocabulary set

Word	Based upon our transliteration	Word	Based upon our transliteration
जल	Jal	होता	Hota
ही	Hi	तो	To
जीवन	Jeevan	संभव	Sambhav
है	Hai	पानी	Pani
एक	Ek	चारो	Chaaro
दुसरे	Dusre	ओर	Or
के	Ke	हाहाकार	Hahakar

पूरक	Poorak	मच	Mach
पृथ्वी	Prithvi	जाता	Jata
पर	Pr	व	Va
अगर	Agar	हर	Har
नही	Nahi	जगह	Jagah
लेकिन	Lekin	उपयोगी	Upyogi
की	Ki	सकता	Sakta
सागर	Sagar	मात्रा	Matra
कम	Km	पीने	Peene
योग्य	Yogya	बहुत	Bahut
केवल	Keval	तीन	Teen
प्रतिशत	Pratishat	शेष	Sesh
खारा	Khara	आज	Aaj
इस	Is	लिए	Liye
यह	Yeh	ने	Na
जा	Ja	देखा	Dekha
गया	Gaya	बन	Ban
दायी	Dayi	हम	Hum
बन	Ban		

4.5.3 Front-End Design

Once the data was recorded, each speech file was labeled with corresponding word. The developed system uses manual labeling. *Wavesurfer* is used for labeling the speech files. Figure 4.2 shows the labeled speech-waveform.

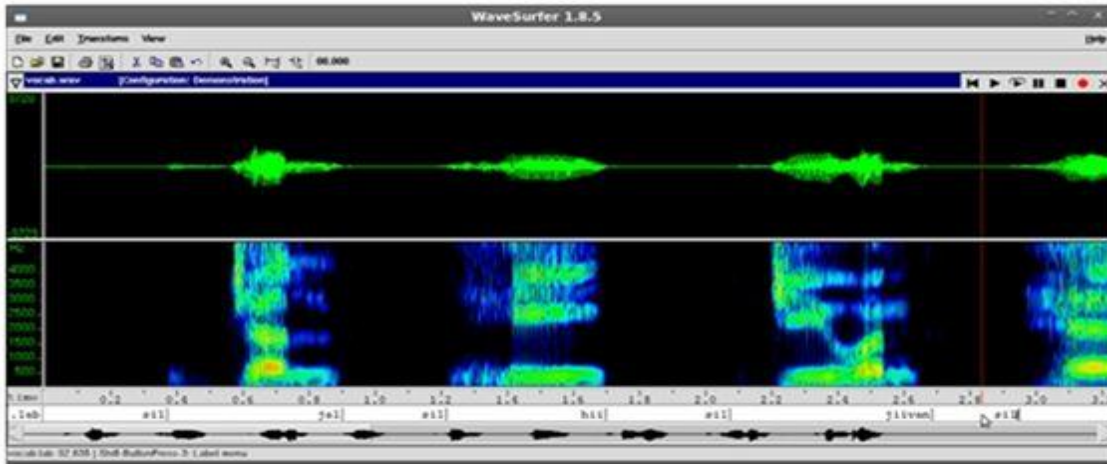


Figure 4.2 Labeled Speech waveform

Once the files were labeled, the recorded data is parameterized into a sequence of features. For this purpose, HTK tool *HCopy* was used. This command uses a configuration file *analysis.conf* which includes all the parameters used for acoustic analysis of the speech waveform. Figure 4.3 shows the configuration file used. Mel frequency cepstral coefficient (MFCC) has been used for parameterization of data. The acoustic parameters were 39 MFCCs with 12 Mel cepstrum plus log energy and their first and second order derivatives.

```

analysis.conf (~/isolated) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
analysis.conf x
SOURCEFORMAT = WAVE # Gives the format of the speech files
TARGETKIND = MFCC_0_D_A # Identifier of the coefficients to use

# Unit = 0.1 micro-second :
WINDOWSIZE = 250000.0 # = 25 ms = length of a time frame
TARGETRATE = 100000.0 # = 10 ms = frame periodicity

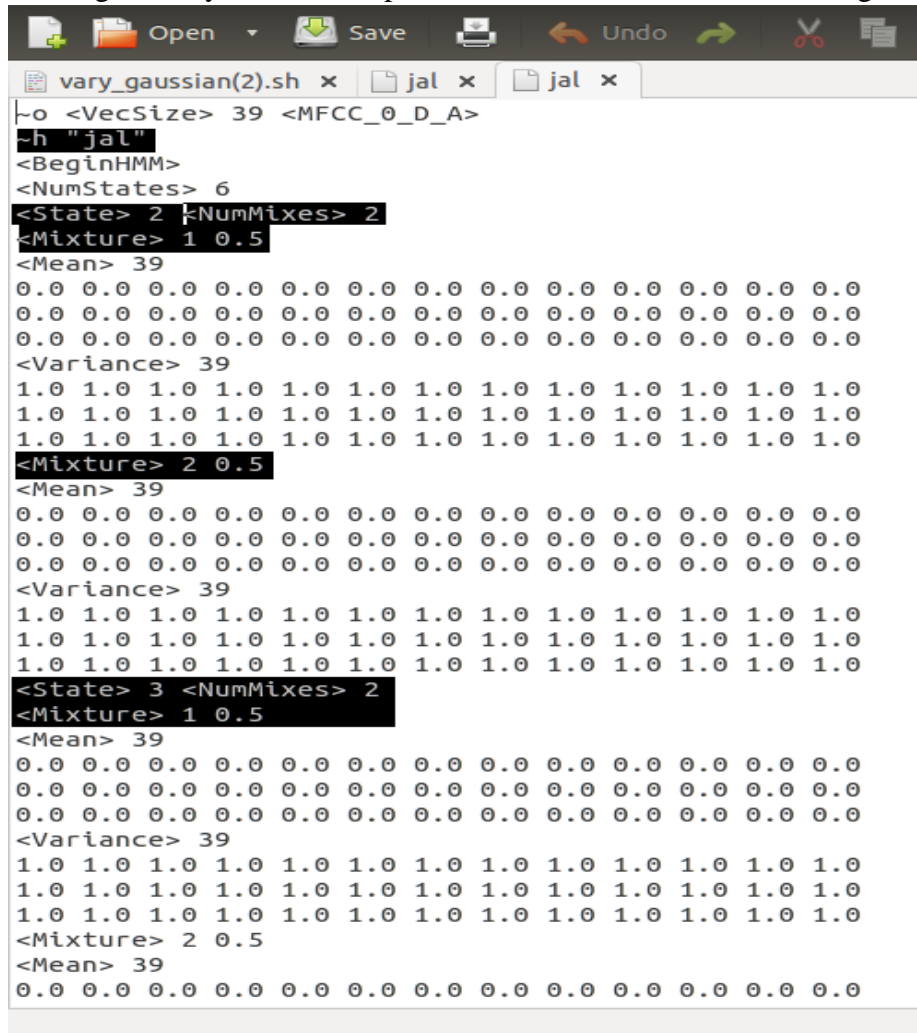
NUMCEPS = 12 # Number of MFCC coeffs (here from c1 to c12)
USEHAMMING = T # Use of Hamming function for windowing
frames
PREEMCOEF = 0.97 # Pre-emphasis coefficient
NUMCHANS = 26 # Number of filterbank channels
CEPLIFTER = 22 # Length of cepstral liftering

Plain Text Tab Width: 8 Ln 11, Col 37 INS
  
```

Figure 4.3 Configuration file

4.5.4 Back-End Design

For training the HMM, a prototype HMM model was created with varying number of Gaussian mixtures. HMM definition is done in model/gaussian/proto folder for all files using vary_gaussian.sh file. Prototype for word “jal” with 2 gaussian mixtures is shown in figure 4.5. Apart from the models of vocabulary words, model for silent (sil) was also included. For prototype models, 6-10 state HMMs with 1-5 Gaussian mixtures have been used in which the first and last were non-emitting states. The prototype models were initialized using the HTK tool *HInit* which initializes the HMM model based on one of the speech recordings. The prototype and the corresponding initialization is shown in figure 4.4 and 4.5. Then *HRest* was used to re-estimate the parameters of the HMM model based on the other speech recordings in the training set. Task grammar for the recognition system based upon own transliteration is shown in figure 4.6.



```
vary_gaussian(2).sh x jal x jal x
|_o <VecSize> 39 <MFCC_0_D_A>
-h "jal"
<BeginHMM>
<NumStates> 6
<State> 2 <NumMixes> 2
<Mixture> 1 0.5
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<Mixture> 2 0.5
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3 <NumMixes> 2
<Mixture> 1 0.5
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<Mixture> 2 0.5
<Mean> 39
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

Figure 4.4 Prototype for “jal” HMM with 2 gaussian mixtures

```

vary_gaussian(2).sh x  jal x
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
-h "jal"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<NUMMIXES> 2
<MIXTURE> 1 3.545719e-01
<MEAN> 39
-1.578218e+01 1.076989e+01 -7.668446e-01 2.516137e-02 4.085817e+00 -6.686939e+00 4.399506e+00 -7.710737e+00 -1.215896e+00 -5.080512e+00
-4.013512e-01 -3.213497e+00 5.742287e+01 7.722217e-01 2.323366e-01 8.118663e-01 -1.480974e+00 -1.313265e+00 -2.845610e+00 2.378247e-01
1.134980e+00 8.905055e-01 8.790101e-01 1.620268e+00 1.054044e+00 3.305383e+00 1.449071e+00 -5.072758e-01 1.074812e+00 6.937715e-01
-9.739601e-01 9.247819e-02 -1.059078e+00 3.841498e-01 2.947642e-02 -3.725854e-01 5.411496e-02 7.390668e-02 -3.063373e-01
<VARIANCE> 39
1.038081e+01 1.922555e+01 1.951312e+01 9.742307e+00 1.043914e+01 1.987814e+01 1.052370e+01 2.446949e+01 1.684380e+01 2.073364e+01 1.556303e
+01 1.260404e+01 1.773444e+01 9.720391e+00 2.612648e+00 8.261589e+00 2.101844e+00 3.281144e+00 1.487861e+00 3.832731e+00 2.848338e+00 2.735935e
+00 2.490837e+00 1.896953e+00 1.139368e+00 1.045802e+00 4.913912e-01 1.078665e+00 9.850616e-01 2.005668e-01 6.077700e-01 4.731217e-01
4.287116e-01 9.802848e-01 8.078709e-01 1.078272e+00 3.377527e-01 3.484980e-01 1.807780e-01
<GCONST> 1.115591e+02
<MIXTURE> 2 6.454282e-01
<MEAN> 39
-6.364822e+00 1.300020e+01 7.989774e+00 9.528796e+00 1.563161e+00 -2.865957e-01 -5.583628e+00 -6.552521e+00 -6.704908e+00 -5.664888e+00
-6.503641e+00 -3.795092e+00 4.433134e+01 -8.051294e-01 -4.190952e-01 -6.995011e-01 -2.804835e-01 6.941449e-01 1.711425e-01 9.623531e-01
-2.621576e-01 7.238091e-02 -5.521938e-01 1.058708e-02 -3.890342e-01 4.911930e-01 -5.220193e-01 -3.254710e-01 -5.568092e-01 -4.558284e-01
6.507070e-02 -1.323125e-01 4.444850e-01 1.112313e-01 3.868566e-01 1.833423e-01 3.765624e-01 1.705689e-01 3.229808e-01
<VARIANCE> 39
7.008319e+00 4.209572e+00 5.434771e+00 4.473987e+00 1.648727e+01 1.332191e+01 1.835688e+01 1.700472e+01 8.887227e+00 1.091589e+01 7.059617e
+00 8.315748e+00 1.947578e+00 1.495901e+00 1.274975e+00 2.272208e+00 1.739933e+00 1.637485e+00 2.534847e+00 2.371824e+00 3.333159e+00 2.198626e
+00 1.459777e+00 1.413224e+00 9.629490e-01 1.991995e+00 2.444187e-01 3.964794e-01 3.164140e-01 1.812437e-01 4.389942e-01 8.368410e-01
4.333907e-01 8.821360e-01 3.249283e-01 3.070734e-01 2.943045e-01 2.112810e-01 5.594686e-01
<GCONST> 9.355788e+01
<STATE> 3
<NUMMIXES> 2
<MIXTURE> 1 6.960135e-01
<MEAN> 39

```

Figure 4.5 HMM Training for “jal” HMM with 2 gaussian mixtures

```

grammer.txt (-/Desktop/speech_recognition/connected/english) - gedit
/*
 * Grammar
 */
$WORD = JAL | HI | JEEVAN | HAI | VA | EK | DUSRE | KE | POORAK | PRITHVI | PR | AGAR |
NA | HOTA | TO | SAMBHAV | NAHI | PANI | DAYI | ABHAV | ME | CHARO | OR | HAHAKAR | MACH
| JATA | HAR | JAGAH | LEKIN | UPYOGI | KI | MATRA | KM | JIS | PRAKAAR | SAGAR | REH |
KAR | BHI | KA | PI | SAKTE | USI | PEENE | YOGYA | BAHUT | KEVAL | TEEN | PRATISHAT |
SHESH | KHARA | sil ;

( {<sil> [$WORD] {sil}} )

```

Figure 4.6 Grammar file for connected words

The complete functioning of the system can be represented by the following flowdiagram which explains each and every step to convert the speech into a sequence of text files.

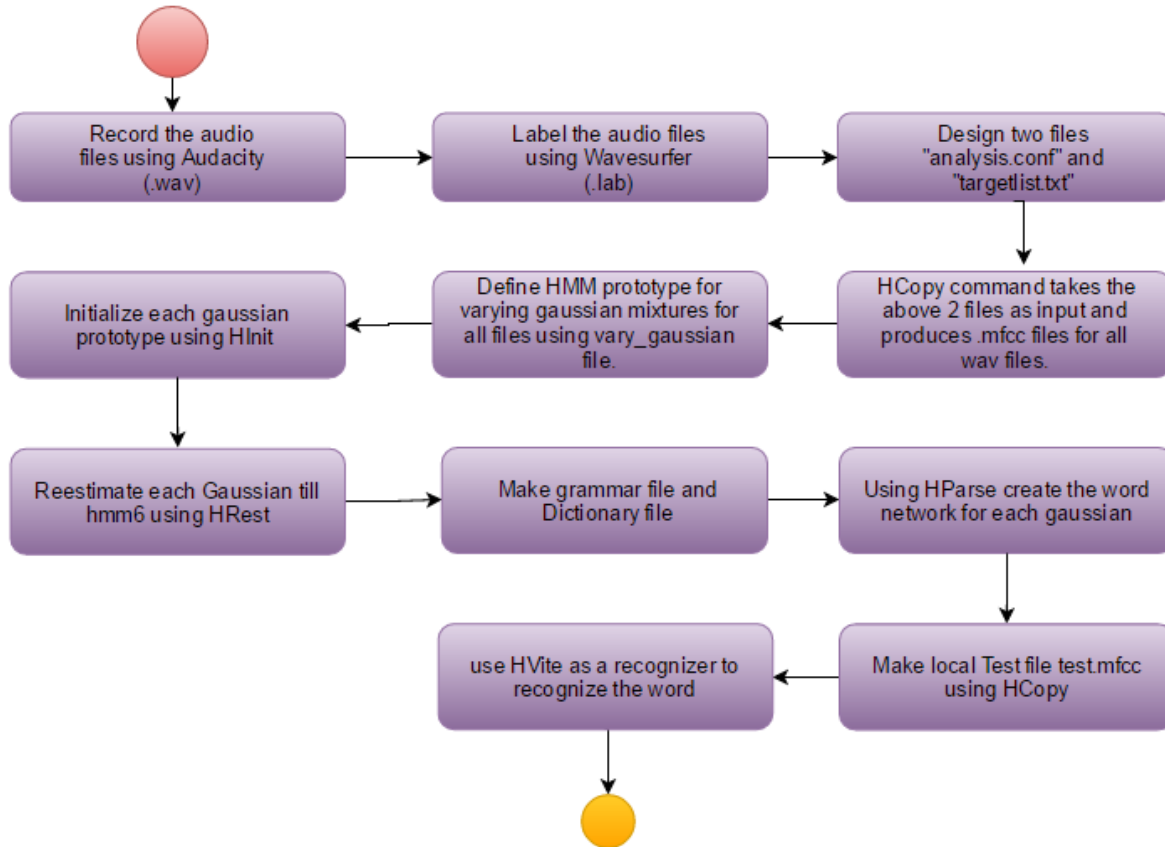


Figure 4.7 Flowchart to convert speech signal to sequence of text

4.5.5 Performance Evaluation

During evaluation, system is responsible for generating the transcription for an unknown utterance. The model generated during the training phase is responsible for evaluation. In order to evaluate the system performance, speakers were asked to utter some words of the vocabulary. For testing five speakers were used. Figure 4.8 displays the recognized word during testing.

```

*log.txt (~/Desktop/speech_recognition/connected/english/test) - gedit
HVite -A -D -T 1 -S mfcclist.txt -H ../Hmndef_hmm5/10states/hmmsdef_g1_hmm5.txt -i rec_perf.mlf -w ../net.slf ../
dict.txt ../hmmlist.txt

No HTK Configuration Parameters Set

Read 52 physical / 52 logical HMMs
Read lattice with 57 nodes / 113 arcs
Created network with 113 nodes / 169 links
File: test_mfcc/test1.mfcc
sil sil JAL sil HI sil JEEVAN sil HAI sil == [998 frames] -60.8323 [Ac=-60710.7 LM=0.0] (Act=110.1)
File: test_mfcc/test2.mfcc
sil VA sil EK sil sil DUSRE sil sil KE sil POORAK sil PRITHVI sil PR == [998 frames] -68.0351 [Ac=-67899.1 LM=0.0]
(Act=110.1)
File: test_mfcc/test4.mfcc
sil AGAR sil NA sil HOTA sil TO sil SAMBHAV sil NAHI sil sil sil == [998 frames] -70.6709 [Ac=-70529.6 LM=0.0]
(Act=110.1)
File: test_mfcc/test7.mfcc
sil DAYI sil sil ABHAV sil sil KE sil CHARO sil OR sil HAHAKAR sil MACH sil == [998 frames] -70.7701 [Ac=-70628.5
LM=0.0] (Act=110.1)
File: test_mfcc/test8.mfcc
sil JATA sil HAR sil JAGAH sil LEKIN sil UPYOGI sil HI sil == [998 frames] -68.3011 [Ac=-68164.5 LM=0.0] (Act=110.1)
No HTK Configuration Parameters Set

```

Figure 4.8 Recognized sentence based upon our transliteration for connected system

4.6 Results and Discussions

HResults is the HTK performance analysis tool. When used to calculate the sentence accuracy the basic output is recognition statistics for the whole file set in the format. Recognition statistics for 6 states GMM based HMM model with 2 Gaussian mixtures is shown in Figure

```

HResults -A -D -T 1 -f -e ??? sil -l ref.mlf ../hmmlist.txt rec_perf.mlf

No HTK Configuration Parameters Set

----- Sentence Scores -----
===== HTK Results Analysis =====

Ref : ref.mlf
Rec : rec_perf.mlf

----- File Results -----
test1.rec: 100.00[100.00] [H= 4, D= 0, S= 0, I= 0, N= 4]
test2.rec: 100.00[100.00] [H= 4, D= 0, S= 0, I= 0, N= 4]
test3.rec: 85.71[ 71.43] [H= 6, D= 0, S= 1, I= 1, N= 7]
test4.rec: 62.50[ 62.50] [H= 5, D= 1, S= 2, I= 0, N= 8]
test5.rec: 57.14[ 28.57] [H= 4, D= 0, S= 3, I= 2, N= 7]
test6.rec: 71.43[ 71.43] [H= 5, D= 0, S= 2, I= 0, N= 7]
test7.rec: 85.71[ 71.43] [H= 6, D= 0, S= 1, I= 1, N= 7]
test8.rec: 100.00[100.00] [H= 7, D= 0, S= 0, I= 0, N= 7]
test9.rec: 83.33[ 83.33] [H= 5, D= 0, S= 1, I= 0, N= 6]
test10.rec: 83.33[ 83.33] [H= 5, D= 0, S= 1, I= 0, N= 6]
test11.rec: 100.00[100.00] [H= 6, D= 0, S= 0, I= 0, N= 6]
test12.rec: 100.00[ 83.33] [H= 6, D= 0, S= 0, I= 1, N= 6]
test13.rec: 100.00[ 66.67] [H= 6, D= 0, S= 0, I= 2, N= 6]
test14.rec: 100.00[ 66.67] [H= 6, D= 0, S= 0, I= 2, N= 6]
test15.rec: 71.43[ 71.43] [H= 5, D= 0, S= 2, I= 0, N= 7]
test16.rec: 71.43[ 71.43] [H= 5, D= 0, S= 2, I= 0, N= 7]

----- Overall Results -----
SENT: %Correct=25.00 [H=4, S=12, N=16]
WORD: %Corr=84.16, Acc=75.25 [H=85, D=1, S=15, I=9, N=101]
=====

```

Figure 4.6.1 Recognition statistics for Gaussian2 HMM6

The first line in overall results gives the sentence-level accuracy based on the total number of label files which are identical to the transcription files. The second line is the word accuracy matches between the label files and the transcriptions. In this second line, H is the number of correct labels, D is the number of deletions, S is the number of substitutions, I is the number of insertions and N is the total number of labels in the defining transcription files. The percentage number of labels correctly recognized is given by

$$\%Correct = \frac{H}{N} \times 100\% \quad (1)$$

And the accuracy is computed by

$$Accuracy = \frac{H - I}{N} \times 100\% \quad (2)$$

The overall accuracy of each HMM with varying Gaussians can be best understood by the following tables:

Table IV: Result for GMM based HMM model with 6 states

Number of Gaussians	Sentence-level Accuracy (in percentage)	Word-level Accuracy (in percentage)	Overall Accuracy (in percentage)
1	37.50	87.13	85.15
2	25.00	84.13	75.25
3	25.00	74.26	66.34
4	12.50	66.34	52.48
5	12.50	65.35	51.49

Table V: Result for GMM based HMM model with 7 states

Number of Gaussians	Sentence-level Accuracy (in percentage)	Word-level Accuracy (in percentage)	Overall Accuracy (in percentage)
1	31	85.15	85.15
2	25.00	77.23	76.24
3	18.75	71.29	66.34
4	12.50	63.37	61.39
5	18.75	61.39	59.41

Table VI: Result for GMM based HMM model with 8 states

Number of Gaussians	Sentence-level Accuracy (in percentage)	Word-level Accuracy (in percentage)	Overall Accuracy (in percentage)
1	37.50	87.13	86.14
2	25.00	73.27	73.27
3	12.50	69.31	69.31
4	18.75	56.44	56.44
5	18.75	58.42	58.42

Table VII: Result for GMM based HMM model with 9 states

Number of Gaussians	Sentence-level Accuracy (in percentage)	Word-level Accuracy (in percentage)	Overall Accuracy (in percentage)
1	37.50	87.13	87.13
2	25.00	74.26	74.26
3	12.50	66.34	66.34
4	12.50	49.50	49.50
5	12.50	47.52	47.52

Table VIII: Result for GMM based HMM model with 10 states

Number of Gaussians	Sentence-level Accuracy (in percentage)	Word-level Accuracy (in percentage)	Overall Accuracy (in percentage)
1	37.50	86.14	86.14
2	18.75	67.33	67.33
3	18.75	60.40	60.40
4	12.50	48.51	48.51
5	12.50	42.57	42.57

Comparison with Other Techniques:

The result of our Proposed work is compared with techniques DTW (Dynamic Time warping) [48] and HMM (Hidden Markov Model)[48] at back-end.

The efficiency of connected word recognition is defined in terms of WER % (word error rate) given as

$$WER\% = \frac{100(I+S+D)}{\text{total word in correct transcript}} \quad (1)$$

where I = insertions, S = substitutions and D = deletions

The comparison of the results between DTW, HMM and the proposed model with 10 states is shown in the table IX.

Table IX: WER% Connected Speech Recognition

	DTW	HMM	Proposed GMM
%WER	13.33	13.33	12.87

As it can be seen in the table IX that the technique proposed in this dissertation is better than other techniques as it has less error rate. [48]

Result for Isolated Word Recognition: HMM6

With increase in number of Gaussians (in word model for Hindi ASR for 51 words), accuracy decreases and WER increases. Moreover, lower no. of states shows lesser WER with increasing Gaussians.

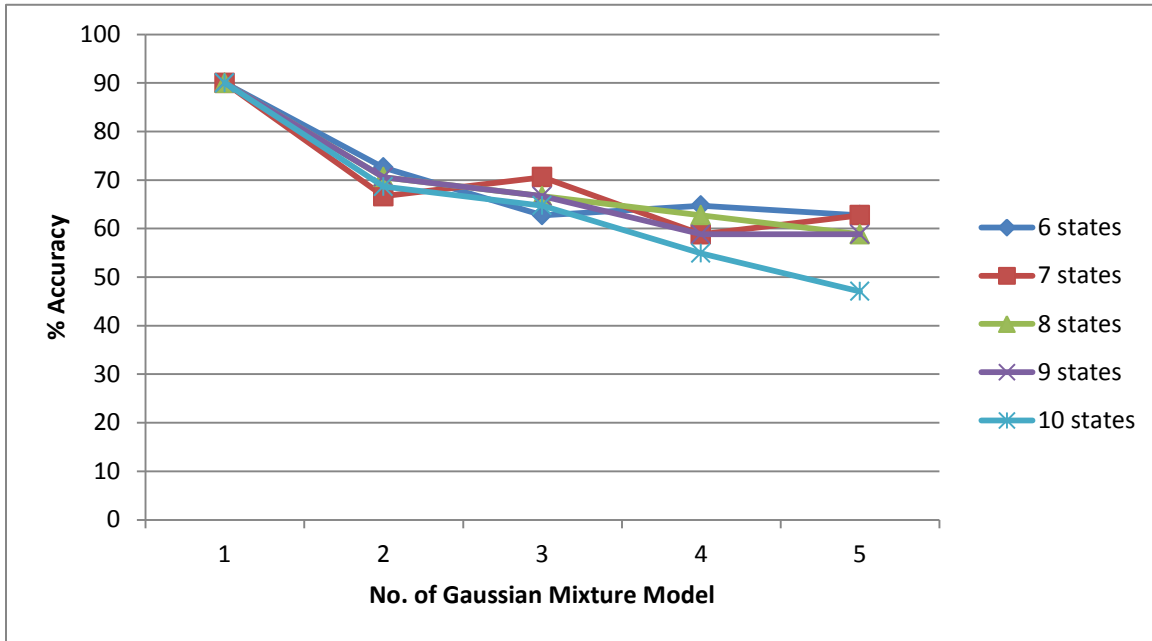


Figure 4.6.2 Result for Isolated Word Recognition: HMM6

Result for Connected Word Recognition: HMM6

With increase in number of Gaussians (model for Hindi ASR for 51 words), accuracy decreases

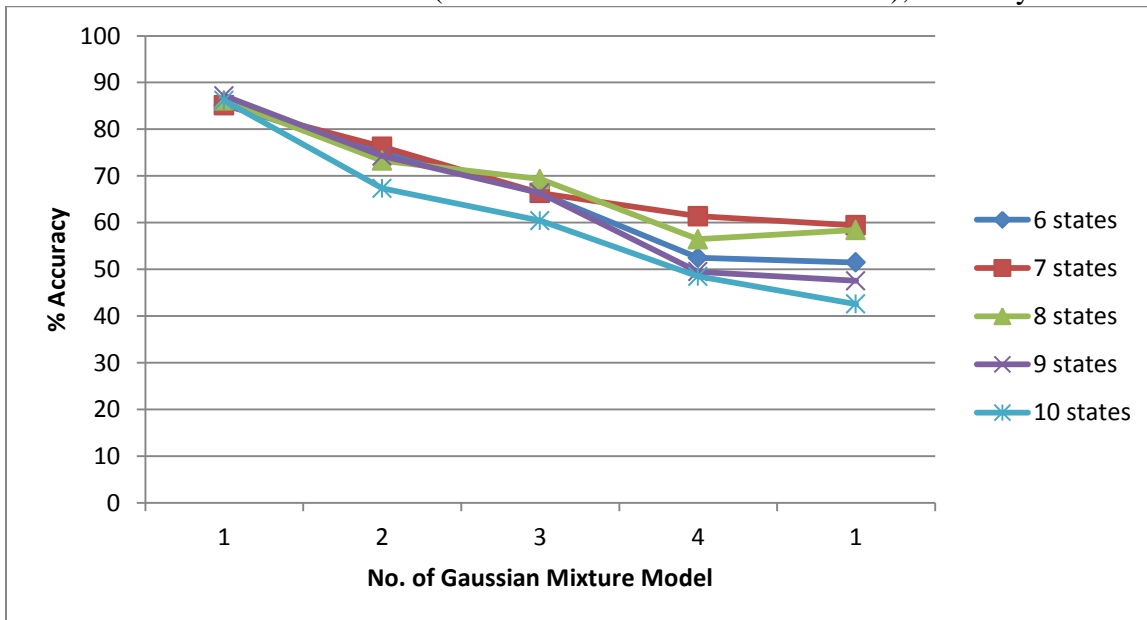


Figure 4.6.3 Result for Connected Word Recognition: HMM6

Conclusion and Future Directions

5.1 Conclusion

The automatic speech recognition system for Hindi language has been developed successfully. The developed system recognizes the connected words using acoustic word model. The training of the system has been done using 51 Hindi words vocabulary. This dissertation has mentioned and explained the detailed procedure to select an optimum number of Gaussian mixtures that exhibits maximum accuracy in the context of Hindi speech recognition system for small database. We have also varied the number of HMMs and studied their effect on the speech recognition with varied number of GMMs. For small database (51 words vocabulary) the best results can be observed at single Gaussian, and as we increase the vocabulary the accuracy and efficiency in case of single GMM reduces and increases at relatively higher value of GMMs.

5.2 Suggestions for Future Research

In this section some suggestions for further research are provided.

- This project can also be implemented using phone model.
- This work can also be tried for other Indian languages like Punjabi, Bengali, Marathi, etc.
- The number of Gaussian Mixtures can further be increased for larger vocabulary for the same Hindi Language.
- Open source tools like SPHINX, designed mainly for English language, can be used for Hindi language and compared with HTK's results for Gaussian's variation. Such tools should be trained directly with Hindi language fonts.

- In our work, close talking microphones were used. They should be replaced with phone-sets kept at some distance.
- In our system MFCC is used for feature extraction. To improve the accuracy of the system we can use other feature extraction techniques also like PLP or MF-PLP.

References

- [1] H. Dudley, “The Vocoder”, *Bell Labs Record*, Vol. 17, pp. 122-126, 1939.
- [2] H. Dudley, R. R. Riesz, and S. A. Watkins, “A Synthetic Speaker”, *J. Franklin Institute*, Vol. 227, pp. 739-764, 1939.
- [3] K.H. Davies , R. Biddulph, and S. Balashek, “Automatic Speech Recognition of Spoken Digits”, *Journal of Acoustic Signal, Speech Processing* , vol. 24(6), pp.637 – 642 , 1952.
- [4] J. Suzuki and K. Nakata, “Recognition of Japanese Vowels—Preliminary to the Recognition of Speech”, *J. Radio Res. Lab*, Vol. 37 (8), pp. 193-212, 1961.
- [5] J. Sakai and S. Doshita, “The Phonetic Typewriter, Information Processing”, *In Proceedings of IFIP Congress*, Munich, 1962.
- [6] F. Itakura, “Minimum Prediction Residual Principle Applied to Speech Recognition”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol. ASSP-23, pp. 57-72, 1975.
- [7] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg and J. G. Wilpon, “Speaker Independent Recognition of Isolated Words Using Clustering Techniques”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol. Assp-27, pp. 336-349,1979.
- [8] S. B. Devis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28 (4), 1980.
- [9] J. G. Wilpon, L. R. Rabiner, C. H. Lee and E. R. Goldman, “Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol. 38 (11), pp. 1870-1878, 1990.
- [10] L. R. Rabiner and B. H. Juang, “Statistical Methods for the Recognition and Understanding of Speech”, *Encyclopedia of Language and Linguistics*, 2004.
- [11] H. Bourlard, N. Morgan and S. Renals, “Neural Nets and Hidden Markov Models: Review and generalizations”, *Speech Communication*, Vol. 11 (2-3), pp. 83-92, 1992.
- [12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, “Phoneme recognition using time delay neural networks”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol.37, pp.328-339, 1989.
- [13] D.E. Rumelhart, G.E. Hinton, R.J. Williams, “Learning Internal representations by error propagation”, Vol. 1, *MIT Press, Cambridge*, 1986, pp. 318-362 .

- [14] B. Wheatley and J. Picone, "Voice across America: Toward Robust Speaker Independent Speech Recognition for Telecommunications Applications", *Digital Signal Processing: a Review Journal*, Vol. 1(2), pp. 45–64.
- [15] H. Hermansky, "Perceptual Linear Predictive Analysis", *Journal of Acoustic Soc. Am.*, pp. 1738- 1752, 1990.
- [16] H.Hermansky, S.Sharma, "Temporal Patterns (TRAP) in ASR of Noisy Speech", *IEEE Transaction of Acoustics, Speech and Signal Processing (ICASSP-99)*, Vol. 1, pp. 289-292, 1999.
- [17] J. Bilmes, "Maximal Mutual Information Based Reduction Strategies for Cross-Correlation Based Joint Distributional Modeling", *IEEE Transaction of Acoustics, Speech and Signal Processing (ICASSP-98)*, SP 14.6, Seattle, 1998.
- [18] R. P. Lippmann, "Review of Neural Networks for Speech Recognition, Readings in Speech Recognition", A. Waibel and K. F. Lee, Editors, *Morgan Kaufmann Publishers*, pp. 374-392, 1990.
- [19] H. Bourlard, C. Wellekens, "Links between Hidden Markov Models and Multilayer Perceptrons", *IEEE Transaction of Pattern Analysis Machine Intelligence*, Vol. 12, pp. 1167-1178, 1990.
- [20] H. Hermansky, S. Sharma, "Feature Extraction using Non-Linear Transformation for Robust Speech Recognition on the Aurora Database" *IEEE Transaction of Acoustics, Speech and Signal Processing*, 2000.
- [21] A. Hagen, J. Neto, "Multi-Stream Processing Using Context-Independent and Context-Dependent Hybrid Systems", *IEEE Transaction of Acoustics, Speech and Signal Processing (ICASSP - 03)*, pp. 277–280, 2003.
- [22] P. Jancovic, J. Ming, "A Multi- Band Approach Based on the Probabilistic Union Model and Frequency Filtering Features for Robust Speech Recognition", *In: Euro Speech' 01*. pp. 579–582, 2001.
- [23] R. K. Aggarwal and M. Dave, "Performance evaluation of sequentially combined heterogeneous feature streams for Hindi speech recognition system," *Telecommunication Systems Journal (Special issue on Signal Processing Applications in Human Computer Interaction)*, Springer (online first), Sept. 2011.

- [24] R.K. Aggarwal and M. Dave, “Fitness Evaluation of Gaussian Mixtures in Hindi Speech Recognition System”, *ICIIC Published by IEEE Computer Society*, 2010.
- [25] B. H. Juang and Lawrence R. Rabiner, “Automatic Speech Recognition – A Brief History of the Technology Development”, 2004.
- [26] Madhav Pandya , “Data Driven Feature Extraction and Parameterization for Speech Recognition” , *M.Tech Thesis , IIT Kanpur*, 2005
- [27] C. H. Lee, F. K. Soong, and K. K. Paliwal, “Automatic Speech and Speaker Recognition Advanced Topics”, *Kluwer Academic Publisher*, 1995.
- [28] B. Yegnanarayana and R. N. J. Veldhuis, “Extraction of Vocal-Tract system Characteristics from Speech Signals”. *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol. 6 (4), pp. 313-327, 1998.
- [29] L.R. Rabiner and R.W. Schafer, “Digital Process of speech signals”, *Englewood Cliffs New-Jersey, Prentice –Hall*, 2000.
- [30] Edmondo Trentin, Marco Gori, “A Survey of Hybrid ANN/HMM Models for Automatic Speech Recognition”, *Neurocomputing*, Vol. 37, pp. 91-126, 2001.
- [31] J. W. Picone, “Signal Modeling Technique in Speech Recognition”, *Proceedings of the IEEE*, Vol. 81 (9), pp. 1215-1247, 1993.
- [32] Claudio Becchetti and Lucio Prina Ricotti, “Speech Recognition Theory and C++ Implementation”, *John Wiley & Sons*.
- [33] Vrijendra Singh and Narendra Meena, “Engine Fault Diagnosis using DTW, MFCC and FFT”, *IIT Allahabad*.
- [34] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition”, *IEEE Transaction of Acoustics, Speech and Signal Processing*, Vol.77, pp. 257–286, 1989.
- [35] X.D. Huang, Y. Ariki and M. Jack. “Hidden Markov Models for Speech Recognition”, *Edinburgh University Press*, Edinburgh, 1990.
- [36] R.K. Aggarwal and M. Dave (2010b) “Fitness Evaluation of Gaussian Mixtures in Hindi Speech Recognition System”. *Paper Presented at First International Conference on Integrated Intelligent Computing. SJB Institute of Technology, Bangalore, India. August 05-07,2010.*

- [37] R.K. Aggarwal and M. Dave (2010a) “Effects of Mixtures in Statistical Modeling of Hindi Speech Recognition System”. *Paper Presented at Second International conference on Intelligent Human Interaction. Allahabad, India.* January 16-18, 2010.
- [38] R.K. Aggarwal and M. Dave “Using Gaussian Mixtures for Hindi Speech Recognition System”. *International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 4, No: 4, December 2011.*
- [39] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland “The HTK Book”, *Microsoft Corporation and Cambridge University Engineering Department, 2009*
- [40] HTK (2013). Hidden Markov Model Toolkit. Obtained through the Internet: <http://htk.eng.cam.ac.uk>, [accessed on Mar. 10, 2013].
- [41] R. Gupta “Speech Recognition for Hindi”. *Master’s Project Report. Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, Mumbai, India, 2006.*
- [42] T. Pruthi, S. Saksena and P. K. Das “Swaranjali: Isolated Word Recognition for Hindi Language using VQ and HMM”. *Paper Presented at International Conference on Multimedia Processing and Systems (ICMPS).* IIT Madras, India. August 13-15, 2000.
- [43] Cai et al., “Efficient likelihood evaluation and dynamic Gaussian selection for HMM-based speech recognition”. *Computer Speech and Language, Elsevier, May 12, 2008.*
- [44] Gales et al., “State-Based Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMM’s”, *IEEE Transactions on Speech and Audio Processing, Vol.7, No. 2, 1999, pp.152–161.*
- [45] H. Bourlard, C. Wellekens, “Links between Hidden Markov Models and Multilayer Perceptrons”, *IEEE Transaction of Pattern Analysis Machine Intelligence, Vol. 12, pp. 1167-1178, 1990.*
- [46] Cygwin (2013), Obtained through the internet: www.cygwin.com.
- [47] N. Rai “Isolated word speaker Independent Speech recognition for Indian Languages”, *Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, 2005.*
- [48] Shweta Singhal, Dr. Rajesh Kumar Dubey “Automatic Speech Recognition for Connected Words using DTW /HMM for English/ Hindi Languages”, *2015 International Conference on Communication, Control and Intelligent Systems (CCIS).*

