HMM BASED HINDI SPEECH RECOGNITION

A Dissertion

Submitted in partial fulfillment of the requirement
for the award of the degree of

MASTER OF ENGINEERING
(COMPUTER TECHNOLOGY & APPLICATIONS)

By
SHOBHA BHATT
College Roll No. 18/CTA/04
Delhi University Roll No. 8515

Under the guidance of
Mr. Rajeev Kumar

Department Of Computer Engineering
Delhi College of Engineering
Bawana Road, Delhi-110042
(University of Delhi)

July -2006

CERTIFICATE

This is certify that the major project on "**HMM BASED HINDI SPEECH RECOGNITION**" submitted by Shobha Bhatt towards partial fulfillment of the requirements for a degree of M.E. (Computer Technology and applications) of University of Delhi through Delhi College of Engineering is a bonafide original record of her work carried out under the guidance and supervision of the undersigned.

(Dr. Amita Dev)
Principal,
Ambedkar Polytechnic

(Sh. Rajiv Kumar)
Lecturer
Computer Deptt. DCE

ACKNOWLEDGMENT

I feel pride in placing on record my deep gratitude to my guide, Sh.. Rajeev Kumar who guided me throughout this project. I am also heartily grateful to Dr. Amita Dev for her resourceful help.Whether it was review of literature or clearing doubts, they spared quality time to guide me despite their extremely busy schedule. They not only helped in solving any problem with comfortable ease but also encouraged and motivated me to sail through in difficult times. It is hard to imagine successful completion of such a project without their guidance and care.

I am very thankful to Professor Dr. Goldie Gabrani for her constant encouragement to accomplish an innovative project.

I am fortunate to receive attention of Prof D. Roy Choudhary , Mrs. Rajni Jindal, and Mr. N.S. Raghava . They constantly boosted my morale and were always there to help whenever needed.

I owe my gratitude to our honorable Principal Dr. P. B. Sharma for providing us all the pre-requisite facilities.

I also want to pay my sincere thanks to the staff Delhi College of Engineering. I extend my thanks and appreciation to friends, family for their patience and support for completing this project.

Shobha Bhatt.

ABSTRACT

Many times key board acts as a barrier between computer and the user. This is true especially for rural areas. We face big challenge in applying this high level specialist knowledge of computers to Community development projects like “Sarva Siksha Abhiyan” so as to quickly bridge the “digital divide”.This barrier can be major hindrance in delivering computer literacy efficiently. Fascinated by the “Hole in the wall” initiative by NIIT, I decided to work on Hindi Speech Recognition so that the users can explore a computer’s potential by interacting with it in their own language thus unlocking the flood gates of development especially for our teeming rural population.

This led to start of my work on “HMM BASED HINDI SPEECH RECOGNITION”.

The main purpose of this study is to perform speech recognition for isolated words in Hindi language. The railway speech corpus has been included for the study.

Two different corpus of speech samples are used for Hindi speakers, one set for training data and other one for testing the data. This study was implemented using the HMM toolkit HTK by training HMMs of, the training data, words. The trained system was tested on data other than the training data . A new technique for mel cepstral coefficients has been developed. Auto Labeling program has been developed for isolated word recognition.

The developed system can be used by researchers interested in the field of Hindi language speech recognition. The findings of the study can be generalized to cater for large vocabularies and for continuous speech recognition

LIST OF ACRONYMS/ ABBREVIATIONS

LVCSR	Large Vocabulary Continuous Speech Recognition.
ASR	Automatic speech recognition
TTS	Text- to-speech
IVR	Interactive Voice Response
HCI	Human Computer Interaction
I/O	Input and Output
SU	Speech Understanding
GUI	Graphical User Interface
DVI	Direct Voice Input
HMM	Hidden Markov Models
HTK	Hidden Markov Model Toolkit.
BNF	Backus-Naur form
SLF	Standard Lattice Format
MLF	Master Label Files
MFCC	Mel Frequency Cepstral Coefficients.

Content

Chapter 1

INTRODUCTION

Technology is constantly searching for ways to accommodate workers with various types of disabilities, while decreasing the number of Worker's Compensation claims paid by creating an ergonomically correct work environment. Speech Recognition software has the capability of meeting these needs and students will be exposed to it in their future careers.

Speech recognition can be defined as the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words . The Voice/speech recognition is a field of computer science that deals with designing computer systems that recognize spoken words. It is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone. Automatic speech recognition (ASR) is a field in the framework of speech science and engineering. In the new generation of computing technology, it comes as the next major innovation in man-machine interaction, after functionality of text-to-speech (TTS), supporting interactive voice response (IVR) systems.

Applications of speech recognition can be in public areas like train stations, airports or tourist information centres might serve the customer with answers to their spoken query. Physically handicapped or elderly people might also be able to access services in a more natural way, since the use of a keyboard is not required

Speech technology is the technology of today and tomorrow with a developing number of methods and tools for better implementation. Speech recognition has a number of practical implementations for both fun and serious works. Automatic speech recognition has an interesting and useful implementation in expert systems, a technology whereby computers can act as a substitute for a human expert.

An intelligent computer that acts, responds or thinks like a human being can be equipped with an automatic speech recognition module that enables it to process spoken information. Medical diagnostic systems, for example, can diagnose a patient by asking him a set of questions, the patient responding with answers, and the system responds with what might be a possible diagnosis.

There is an urgent need for development of a convenient, multi-modal human computer interface that enables a wider population of the country to reap the benefits of computers. Here, the role of speech interface cannot be overemphasised as it is the most natural and convenient mode of communication among human beings.

The first attempts was (during the 1950s) to develop techniques in ASR.

The concept of a machine that can recognize the human voice has long been an accepted feature in Science Fiction. From ‘Star Trek’ to George Orwell’s ‘1984’ - *“Actually he was not used to writing by hand. Apart from very short notes, it was usual to dictate everything into the speakwriter.”* - it has been generally assumed that one day it will be possible to converse naturally with an advanced computer-based system. Indeed in his book ‘The Road Ahead’, Bill Gates (co-founder of Microsoft Corp.) hails ASR as one of the most important innovations for future computer operating systems.

Different companies such as Advanced Recognition Technologies, Inc (ART), Microsoft, as well as other companies have been integrating/ implementing speech recognition systems in their software. These voice command based applications will be expected to cover many of the communicational aspects of our daily lives ranging from telephones to the Internet.

Isolated word recognition system uses a single word at a time. In continuous speech recognition works best if phrases and sentences are spoken in a natural flowing manner. Although the software can recognise individual words, it prefers to look at the sequence of sounds in a whole phrase, having been taught that certain word sounds usually occur in predictable combinations.

There are speaker dependent and speaker independent programmes on the market.

The first type, the speaker dependent speech recognition system, is to be trained by the purchaser of the programme with his or her own voice. Long lists of words and sentences are delivered with the programme that have to be pronounced by the owner to allow the programme to create a speaker-specific database. An advantage of such programmes is the possibility of better recognition because no average database has to be formed. At the same time the absence of an average database necessitates the disadvantage that for each different speaker a separate dataset will have to be generated.

Speaker-independent speech recognition systems make no distinction between speakers. A high voice is recognized no less accurately than a low voice, and differences in pronunciation (if not too large) between speakers are not problematic. So only one database suffices. The disadvantage is, however, that the dataset will have to be very large.

1.1 Importance of the Study

Speech is one of the oldest and most natural means of information exchange between human beings. We as humans speak and listen to each other in human-human interface. For centuries people have tried to develop machines that can understand and produce speech as humans do so naturally . Obviously such an interface would yield great benefits . Attempts have been made to develop vocally interactive computers to realise voice/speech recognition. In this case a computer can recognize text and give out a speech output .The first positive results of spoken word recognition came into existence in the 1970s, when general pattern matching techniques were introduced.

As the extension of their applications was limited, the statistical approach to ASR started to be investigated, at the same period. Nowadays, the statistical techniques prevail over ASR applications. Common speech recognition systems these days can recognize thousands of words. The last decade has witnessed dramatic improvement in speech recognition technology, to the extent that high performance algorithms and systems are becoming available. In some cases, the transition from laboratory demonstration to commercial deployment has already begun

The reason for the evolution of ASR, hence improved is that it has a lot of applications in many aspects of our daily life, for example, telephone applications, applications for the physically handicapped and illiterates and many others in the area of computer science. Speech recognition is considered as an input as well as an output during the Human Computer Interaction (HCI) design. HCI involves the design implementation and evaluation of interactive systems in the context of the users' task and work..

The list of applications of automatic speech recognition is so long and is growing; some of known applications include virtual reality, Multimedia searches, auto-attendants, travel information and reservation, translators, natural language understanding and many more applications .

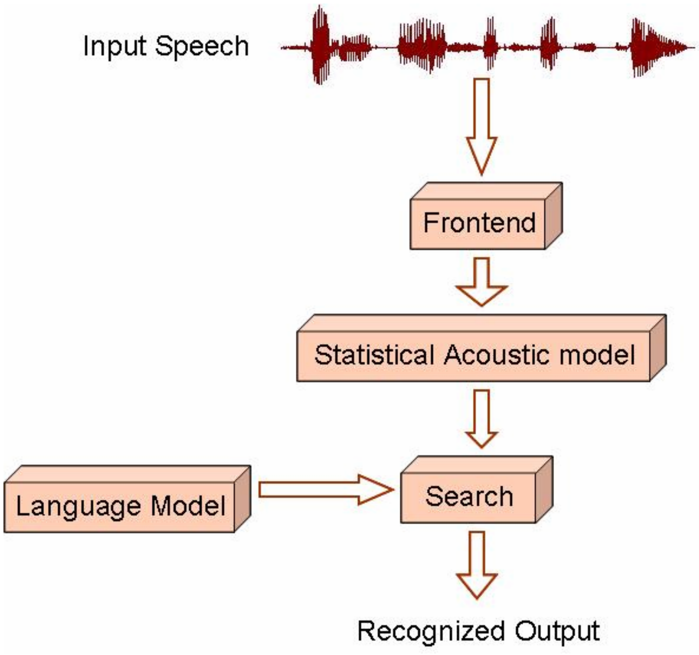


Figure 1: Basic components of a large vocabulary speech recognition system.

A speech recognition system, shown in Figure 1, consists of the following four blocks: feature extraction, language model, acoustic model and search. Feature extraction converts the incoming signal to a stream of vectors, and typically uses an MFCC

Satisfactory speech recognition accuracy can be obtained using sophisticated statistical model such as HMM that adequately characterises the temporal aspect of the speech signal in addition to its spectral properties. Utilisation of more training data together with detailed modelling of speech signal can raise the system performance to a level adequate for actual deployment in appropriate task domains.

1.2 Statement of the Problem (Problem Definition)

There are many people, especially in rural areas, who can only read and write in their mother-tongue Hindi language thus making it difficult for them to use computer tools that are designed for International language like English .

Therefore the purpose of this project was to design and train a speech recognition system that could be used by application developers to develop applications that will take indigenous Hindi language speakers aboard the current information and communication technologies to fast-track the benefits of computer technology.

1.3 Objectives of the Study

1.3.1 General Objective

The general objective of the project was to perform speech recognition for isolated words in Hindi language.

1.3.2 Specific Objectives

The specific objectives of the project are:

1. To critically review literature related to ASR.

11. To identify speech corpus elements exhibited in Hindi language.

111. To build a Hindi language speech corpus for a voice operated system.

IV. To implement an isolated whole word speech recognizer that is capable of recognizing and responding to speech.

V. To train the above developed system in order to make it speaker independent.

1.5 Significance of the Study

The proposed research has theoretical, practical, and methodological significance:

1. The speech corpus developed will be very useful to any researcher who may wish to venture into Hindi language automatic speech recognition.
2. By developing and training a speech recognition system in Hindi language, the semi

illiterates would be able to use it in accessing **IT** tools. Since Speech technology is the technology of today and tomorrow, the results of this research will help many indigenous Hindi language to take advantage of the many benefits of computer technology.

IV. The technology will find applicability in systems such as banking, telecommunications, transport, Internet portals, accessing PC, emailing, administrative and public services, cultural centres and many others.

VI. By developing and training a speech recognition system in Hindi language, it would mark the first step towards making IT tools become more usable by the visually challenged people.

Chapter 2

Literature Review

Human computer interactions as defined in the background is concerned about ways Users (humans) interact with the computers. Some users can interact with the computer using the traditional methods of a keyboard and mouse as the main input devices and the monitor as the main output device. Due to one reason or another some users cannot be able to interact with machines using a mouse and keyboard device, hence the need for special devices. Speech recognition systems help users who in one way or the other can not be able to use the traditional Input and Output devices. For about four decades human beings have been dreaming of an "intelligent machine" which can master the natural speech. In its simplest form, this machine should consist of two subsystems, namely automatic speech recognition (ASR) and speech understanding. The goal of ASR is to transcribe natural speech while SU is to understand the meaning of the transcription. Recognizing and understanding a spoken sentence is obviously a knowledge-intensive process, which must take into account all variable information about the speech communication process, from acoustics to semantics and pragmatics.

2.1 Current State of ASR Technology and its Implications for Design

The design of user interfaces for speech-based applications is dominated by the underlying ASR technology. More often than not, design decisions are based more on the kind of recognition the technology can support rather than on the best dialogue for the user**.** The type of design will depend, broadly, on the answer to this question:

What type of speech input can the system handle, and when can it handle it? When isolated words are all the recognizer can handle, then the success of the application will depend on the ability of designers to construct dialogues that lead the user to respond using single words. Word spotting and the ability to support more complex grammars opens up additional flexibility in the design, but can make the design more difficult by allowing a more diverse set of responses from the user. Some current systems allow a limited form of natural language input, but only within a very specific domain at any particular point in the interaction.

Even in these cases, the prompts must constrain the natural language within acceptable bounds. No systems allow unconstrained natural language interaction, and it's important to note that most human-human transactions over the phone do not permit unconstrained natural language either. Typically, a customer service representative will structure the conversation by asking a series of questions.

It is especially frustrating when a system makes the same mistake twice, but when the active vocabulary can be updated dynamically, recognizer choices that have not been confirmed can be eliminated, and the recognizer will never make the same mistake twice. Also, when more than one choice is available (this is not always the case, as some recognizers return only the top choice), then after the top choice is disconfirmed, the second choice can be presented.

2.2 Types of ASR

ASR products have existed in the marketplace since the 1970s. However, early systems were expensive hardware devices that could only recognize a few isolated words (i.e. words with pauses between them), and needed to be trained by users repeating each of the vocabulary words several times. The 1980s and 90s witnessed a substantial improvement in ASR algorithms and products, and the technology developed to the point where, in the late 1990s, software for desktop dictation became available 'off-the-shelf' for only a few tens of dollars. From a technological perspective it is possible to distinguish between two broad types of ASR: 'direct voice input' (DVI) and 'large vocabulary continuous speech recognition' (LVCSR). DVI devices are primarily aimed at voice command-and-control, whereas LVCSR systems are used for form filling or voice-based document creation. In both cases the underlying technology is more or less the same. DVI systems are typically configured for small to
medium sized vocabularies (up to several thousand words) and might employ word or phrase spotting techniques.

Also, DVI systems are usually required to respond immediately to a voice command. LVCSR systems involve vocabularies of perhaps hundreds of thousands of words, and are typically configured to transcribe continuous speech. Also, LVCSR need not be performed in real-time - for example, at least one
vendor has offered a telephone- based dictation service in which the transcribed document is e- mailed back to the user.

Specific examples of application of ASR may include .

1. large vocabulary dictation

2. Interactive voice response

111. Telecom assistants .

IV. Process and factory management

2.3 Speech Recognition Techniques

Speech recognition techniques are the following:

1. Knowledge based approaches: An expert knowledge about variations in speech is hand coded into a system. This has the advantage of explicit modelling variations in speech; but unfortunately such expert knowledge is difficult to obtain and use successfully. Thus this approach was judged to be impractical and automatic learning procedure was sought instead.

2. Statistical based approaches. In which variations in speech are modelled statistically, using automatic, statistical learning procedure, typically the Hidden Markov Models, or HMM. The approach represent the current state of the art. The main disadvantage of statistical models is that they must take priori modelling assumptions which are liable to be inaccurate, handicapping the system performance. In recent years, a new approach to the challenging problem of conversational speech recognition has emerged, holding a promise to overcome some fundamental limitations of the conventional Hidden Markov Model (HMM) approach

This new approach is a radical departure from the current HMM-based statistical modeling approaches. Rather than using a large number of unstructured Gaussian mixture components to account for the tremendous variation in the observable acoustic data of highly coarticulated spontaneous speech, the new speech model that have developed provides a rich structure for the partially observed (hidden) dynamics in the domain of vocal-tractresonances.

IV. Learning based approaches. To overcome the disadvantage of the HMMs machine learning methods could be introduced such as neural networks and genetic algorithm/

programming. In those machine learning models explicit rules or other domain expert knowledge) do not need to be given they a can be learned automatically through emulations or evolutionary process.

v. The artificial intelligence approach attempts to mechanise the recognition procedure according to the way a person applies its intelligence in visualizing, analysing, and finally making a decision on the measured acoustic features. Expert system are used widely in this approach

2.4 Matching Techniques

Speech-recognition engines match a detected word to a known word using one of the following techniques **.**

1. Whole-word matching. The engine compares the incoming digital-audio signal against a prerecorded template of the word. This technique takes much less processing than sub-word matching, but it requires that the user (or someone) prerecord every word that will be recognized - sometimes several hundred thousand words. Whole-word templates also require large amounts of storage and are practical only if the recognition vocabulary is known when the application is developed.

.

3.1 Sources of information

A speech recognizer makes use of three different sources of information for transcribing a speech signal into written text. The same sources are used by people during the process of recognition: (1) acoustic models, (2) a vocabulary, and (3) a language model [Strik, 2001].

Acoustic models contain information on variation in separate sounds.

.

2.5 Corpora

2.6 Problems in Designing Speech Recognition Systems

1. Number of speakers: With more than one speaker, an ASR system must cope with the difficult problem of speech variability from one speaker to another. This is usually achieved through the use of large speech database as training data .

2. Nature of the utterance: Isolated word recognition impose on the speaker the need to insert artificial pause between successive utterances. Continuous speech recognition systems are able to cope with natural speech utterances in which words may be tied together and may at times be strongly affected by co articulation. Spontaneous speech recognition systems allow the possibility of pause and false starts in the utterance, the use of words not found in the lexicon, etc.

3. Vocabulary size: In general, increasing the size of the vocabulary decrease the recognition scores.

IV. Differences between speakers due to sex, age, accent and so on.

v. Language complexity: The task of continuous speech recognisers is simplified by limiting the number of possible utterances through the imposition of syntactic and semantic constraints.

VI. Environment conditions: The sites for real applications often present adverse conditions (such as noise, distorted signal, and transmission line variability) which can drastically degrade the system performance.

Chapter 4

5.3 Phonetics

Phonetics is the scientific study of the speech sounds of human language. It is concerned with how speech sounds can be categorized, how they are generated in the human vocal tract, why each sound is different to a listener, and how a listener is able to recognize them. Phonetics is categorized into two parts Articulatory and Acoustic phonetics.

5.4Articulatory Phonetics

Articulatory phonetics is the branch of phonetics that deals with the physical process of human speech production. The vocal cords, tongue, velum, lips are the movable parts of the vocal tract that work together to produce the sounds that make up speech. The teeth, alveolar ridge, hard palate play a passive role in speech production. All these parts of speech-producing anatomy are called articulators .

Speech Sounds

Speech sounds occur in two types: consonants and vowels. Speech consists of vowel sounds with intervening consonant sounds.

Consonants: Speech sounds pronounced with obstructions, in which the articulators come close to each other or touch, are called consonants.

The sound of a consonant can be described into two parameters
Place of articulation: The location in the vocal tract of the obstruction, or obstructions, called place of articulation.

(a) Bilabial:bringing both lips together e.g. _ प
(b) Labiodental:the lower lip touching the upper teeth e.g फ
(c) Dental:the tongue tip between the front teeth e.g. थ
(d) Alveolar:the tongue tip near or at the alveolar ridge e.g. ट
(e) Post-alveolar:the tongue tip just behind the alveolar ridge e.g. र
(f) Velar:the tongue body drawn backwards towards the velum e.g. क

Vowel: Speech sounds in which the air stream is un-obstructed are called vowels. Vowels are more prolonged than consonants. Vowels are sounds that resonate in the vocal tract, much like the air in a woodwind. In speaking, the tongue and lips perform a similar function, producing different vowels by altering the shape of the vocal tract so that the vibrating air produces sounds in which different frequencies are emphasized. In phonetic terms, each vowel has a number of properties that distinguish it from other vowels. These include the shape of the lips, which may be rounded,neutral or spread. Secondly, the front, the middle or the back of the tongue may be raised, giving different vowel qualities.The tongue (and the lower jaw) may be raised close to the roof of the mouth, or the tongue may be left low in the mouth with the jaw comparatively open. Alphabetically there are thirteen vowels found in Hindi language.

5.5 Acoustic Phonetics

Acoustic phonetics is the branch of phonetics, that deals with the study and description of the acoustical properties of individual speech sounds , and voice quality. It forms not only the immediate link between articulatory phonetics and speech perception, but is also important for applications in the fields of signal processing and speech technology.

In this phonetics we concentrate on the mechanism of speech production and the ways they may be represented , modeled and simulated.Sound is composed of waves of pressure variations that oscillate from positive to negative relative to surrounding medium, usually air. The number of air pressure oscillations in each second determines the pitch of the sound, whose physical correlate is frequency.The size of the pressure variations determines the loudness of the sound, whose physical correlate is intensity.

3 SPECIFIC FEATURES OF HINDI SOUNDS

Each language has a set of abstract linguistic units called phonemes. For example, English can be described by a set of about 42 phonemes whereas Hindi by about 50 phonemes. The sounds of Hindi speech can be conveniently divided into two broad categories of vowels and consonants. Hindi speech contains a set of about 35 consonants, of which about 29 consonants are of frequent usage. These can be conveniently classified according to the manner and place of production as shown in Table I

MoP/PoA	Bilabials	Dentals	Retroflex	Palatal	Velar	Glottal
UvUa	प	त	ट	च	क	
VoUa	ब	द	ड	ज	ग	
UvAs	फ	थ	ढ	छ	ख	
VoAs	भ	ध	ढ	झ	घ	
Fricative		स		श		ह
Vowel like	व	ल	़	य		
Nasal	म	न				

Articulatory Classification of Hindi Consonants
Table I

Abbreviations/Symbols used :

MoP : Manner of Production VoUa : Voiced Unaspirated
PoA : Place of Articulation UvAs : Unvoiced Aspirated
UvUa : Unvoiced Unaspirated VoAs : Voiced Aspirated

Chapter 4

Speech Feature Extraction
Introduction

The purpose of this module is to convert the speech waveform to some type of parametric representation (at a considerably lower information rate) for further analysis and processing. This is often referred as the *signal-processing front end*.

The speech signal is a slowly timed varying signal (it is called *quasi-stationary*). An example of speech signal is shown in Figure 2. When examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. However, over long periods of time (on the order of 1/5 seconds or more) the signal characteristic change to reflect the different speech sounds being spoken. Therefore, *short-time spectral analysis* is the most common way to characterize the speech signal.

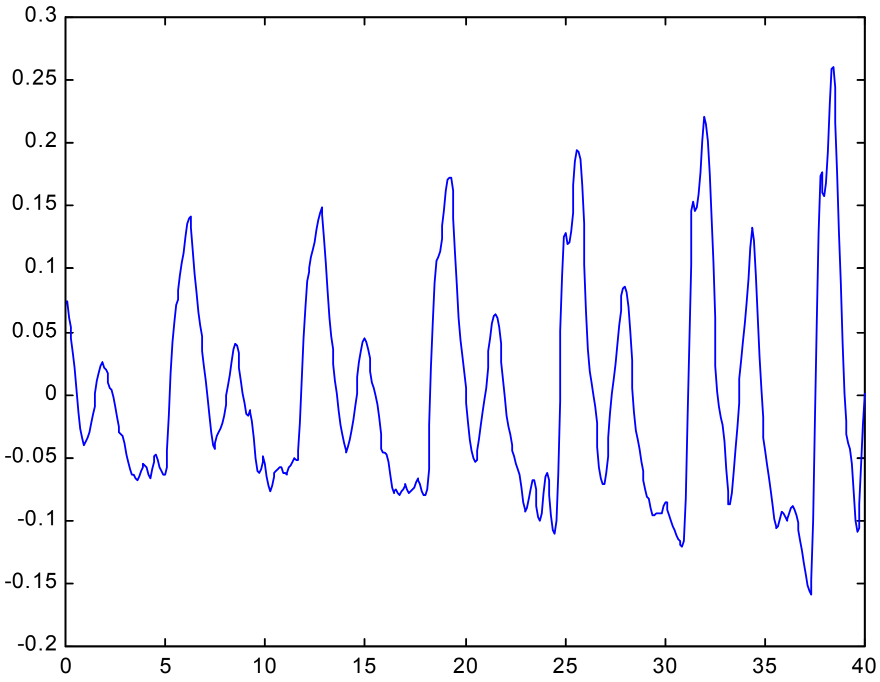


Figure 2. An example of speech signal

A wide range of possibilities exist for parametrically representing the speech signal for the speaker 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 these will be used in this project.

MFCC's are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the *mel-frequency* scale, which is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The process of computing MFCCs is described in more detail next.

Mel-frequency cepstrum coefficients processor

A block diagram of the structure of an MFCC processor is given in Figure 3. The speech input is typically recorded at a sampling rate above 10000 Hz. This sampling frequency was chosen to minimize the effects of *aliasing* in the analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. As been discussed previously, the main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFFC's are shown to be less susceptible to mentioned variations.

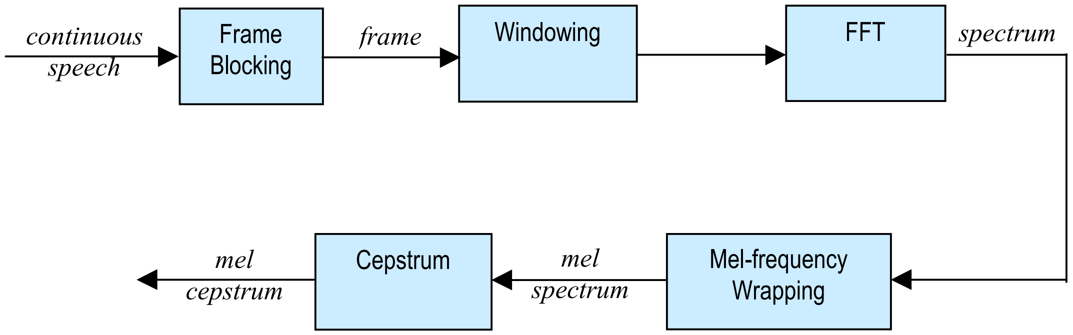


Figure 3. Block diagram of the MFCC processor

Frame Blocking

In this step the continuous speech signal is blocked into frames of N samples, 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. Similarly, the third frame begins $2M$ samples after the first frame (or M samples after the second frame) and overlaps it by N - $2M$ 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 msec windowing and facilitate the fast radix-2 FFT) and $M = 100$.

Windowing

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n), 0 \le n \le N-1$, where N is the number of samples in each frame, then the result of windowing is the signal

$$y_l(n) = x_l(n)w(n), \quad 0 \le n \le N-1$$

Typically the *Hamming* window is used, which has the form:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples $\{x_n\}$, as follow:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, \qquad n = 0,1,2,...,N-1$$

Note that we use j here to denote the imaginary unit, i.e. $j = \sqrt{-1}$. In general X_n's are complex numbers. The resulting sequence $\{X_n\}$ is interpreted as follow: the zero frequency corresponds to $n = 0$, positive frequencies $0 < f < F_s / 2$ correspond to values $1 \le n \le N/2 - 1$, while negative frequencies $-F_s / 2 < f < 0$ correspond to $N/2 + 1 \le n \le N - 1$. Here, F_s denotes the sampling frequency.

The result after this step is often referred to as *spectrum* or *periodogram*.

Mel-frequency Wrapping

As mentioned above, psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on a scale called the 'mel' scale. The *mel-frequency* scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels. Therefore we can use the following approximate formula to compute the mels for a given frequency f in Hz:

$$mel(f) = 2595 * \log_{10}(1 + f / 700)$$

One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel scale (see Figure 4). That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval. The modified spectrum of *S(ω)* thus consists of the output power of these filters when *S(ω)* is the input. The number of mel spectrum coefficients, K, is typically chosen as 20.

Note that this filter bank is applied in the frequency domain, therefore it simply amounts to taking those triangle-shape windows in the Figure 4 on the spectrum. A useful way of thinking about this mel-wrapping filter bank is to view each filter as an histogram bin (where bins have overlap) in the frequency domain.

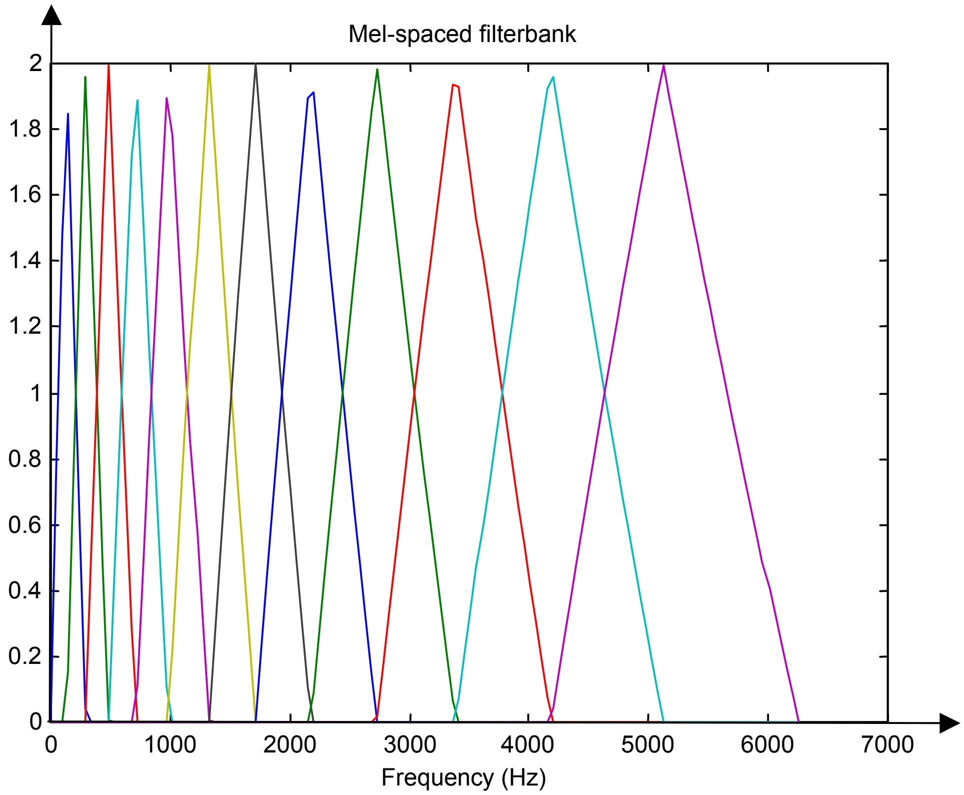


Figure 4. An example of mel-spaced filterbank

Cepstrum

In this final step, we convert the log mel spectrum back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the mel spectrum coefficients (and so their logarithm) are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are $\tilde{S}_k, k = 1,2,...,K$, we can calculate the MFCC's, $\tilde{c}_n$, as

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \qquad n = 1,2,...,K$$

Note that we exclude the first component, $\tilde{c}_0$, from the DCT since it represents the mean value of the input signal which carried little speaker specific information.

INTRODUCTION TO HTK TOOL KIT

Hidden Markov Model toolkit (HTK) was used. HTK is a toolkit for building Hidden Markov Models (HMMs). HMMs can be used to model any time series and the core of HTK is similarly general-purpose. However, HTK is primarily designed for building HMM-based speech processing tools, in particular recognisers**.**

The Hidden Markov Model is a Markov Chain in which the output symbols or probabilistic functions that describe them. To be specific, it uses the graph structure, which is the number of states and their connections, and the number of mixtures per state. The algorithm consists of a set of nodes that are chosen to represent a particular vocabulary. These nodes are ordered and connected from left to right, and recursive loops are allowed. Recognition is based on a transition matrix of changing from one node to another.

The HMM is referred to often as a parametric model because the state of the system at each time t is completely described by a finite set of parameters.

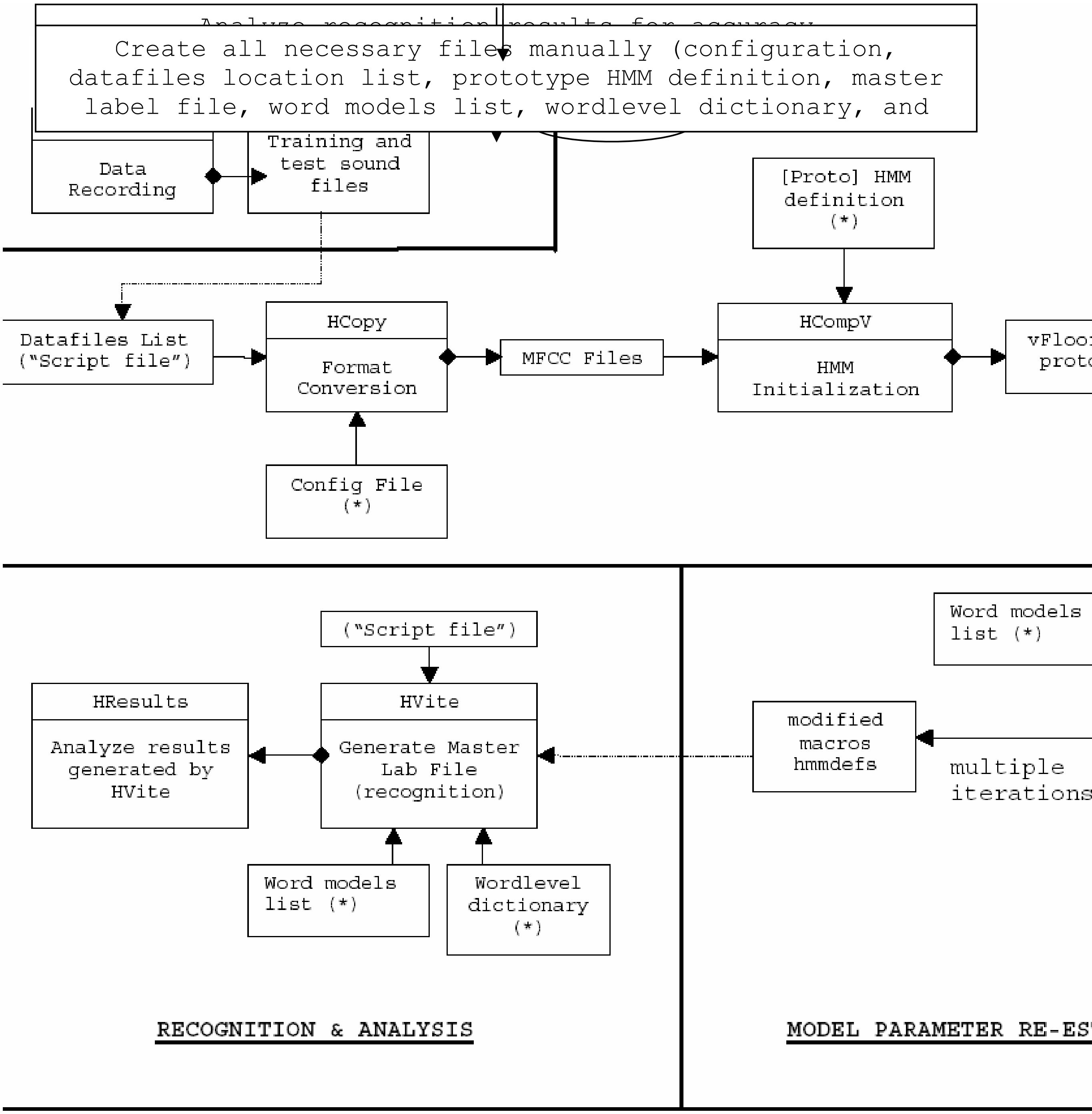
The training algorithm estimates the HMM parameters by taking a first good guess using the preprocessed speech data (features).. The HMM parameters are kept or stored as files and then retrieved by the training procedure.Model training is performed by estimating the HMM parameters, since estimation accuracy is roughly proportional to the number of training data. The HMM is well suited for a speaker-independent system because the speech used during training uses probabilities or generalizations and that makes it a good system to use for multiple speakers.

The various tools are summarized in the following table.

1) Create word net from specified grammar format HParse
2) Convert files into MFCC format HCopy
3) Initialize prototype model based on all speech data HCompV

4) Re-estimate parameters HRest
5) Recognize new speech data (in MFCC format) HVite

Figure 3.5: HTK Procedures & Commands

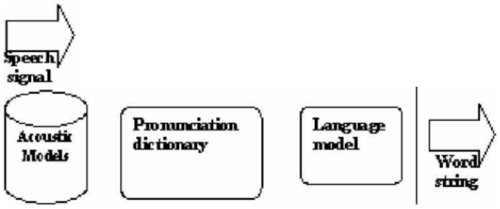

Analyze recognition results for accuracy
Create all necessary files manually (configuration, datafiles location list, prototype HMM definition, master label file, word models list, wordlevel dictionary, and
Data Recording
Training and test sound files
[Proto] HMM definition (*)
Datafiles List ("Script file")
HCopy
Format Conversion
MFCC Files
HCompV
HMM Initialization
vFloo
proto
Config File (*)
("Script file")
HResults
Analyze results generated by HVite
HVite
Generate Master Lab File (recognition)
Word models list (*)
Wordlevel dictionary (*)
Word models list (*)
modified macros hmmdefs
multiple iterations
RECOGNITION & ANALYSIS
MODEL PARAMETER RE-ES

Chapter 4

METHODOLOGY

In this chapter full description of how the Hindi language speech recognition system was developed. The goal of the project was to build a robust whole word recognizer. That means it should be able to generalise both from speaker specific properties and its training should be more than just instance based learning.

In ASR systems acoustic information is sampled as a signal suitable for processing by computers and fed into a recognition process**.**



Automatic Speech Recognition System

Speech recognition is a complicated task and state of the art recognition systems are very complex. I have developed HINDI speech recognition for railways. There are a big number of different approaches for the implementation[11] of an ASR but for this project the four major processing steps as suggested by HTK were considered namely data preparation, training, Recognition/testing and analysis.

For implementation purposes the following sub-processes were taken

1. Building the task grammar

11. Constructing a dictionary for the models

111. Recording the data.

IV. Creating transcription files for training data

v. Encoding the data (feature processing)

VI. (Re-) training the acoustic models

V11. Evaluating the recognisers against the test data

V111. Reporting recognition results

3.1 Data Preparation

The first stage of any recogniser development project is data preparation. Speech data is needed both for training and for testing. The training data is used during the development of the system. Test data provides the reference transcriptions against which the recogniser's performance can be measured . It follows from above that before the data can be recorded, a phone set must be defined, a dictionary must be constructed to cover both training and testing and a task grammar must be defined.

3.1.1 The Task Grammar

The task grammar defines constraints on what the recognizer can expect as input. As the system built provides a voice operated interface for railway system.. The grammar was defined in BN-form, as follows: $variable defines a phrase as anything between the subsequent = sign and the semicolon, where I stands for a logical or. Brackets have the usual grouping function and square brackets denote optionality.

The few lines of grammar was:

```
#
#Task grammar
# $word =BAD|AGLE|PASHIM|DIBBA|DWARA|DECEMBER;
(SENT-START [$word] SENT-END)
```

The above high-level representation of a task grammar is provided for user convenience. The HTK recogniser actually requires a word network to be defined using a low level notation called HTK Standard Lattice Format (SLF) in which each word instance and each word-toword transition is listed explicitly. This word network can be created automatically from the grammar above using the HParse tool, thus assuming that the file gram contains the above grammar, executing

HParse gram wdnet

Creates an equivalent word network in the file wdnet (appendix A) see the figure below

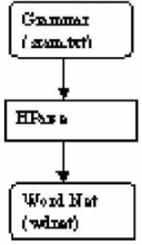


Figure 3.3: Process of creating a word lattice

3.1.2 A Pronunciation Dictionary

The dictionary provides an association between words used in the task grammar and the acoustic models which may be composed of sub word (phonetic, sysllabic etc,,) units. In order to train the HMM network, a large pronunciation dictionary is needed.

Since we are using whole-word models in this assignment, the dictionary has a simple structure.

A file called 'lexicon' was created. Few lines from lexicon are .:

agle	axgley
bad	baadh
birth	bdrth
chaubis	chawbiys
december	dihsmbr
dibba	dihbdbaa
dwara	dhvaaraa
hemgiri	hihmgihriy
hosakta	howsktxaa
kaunsi	kawnsiy

ke	key
panch	paanch
pashim	phihchm
sent-end	[] sil
sent-start	[] sil
sham	shaam

A file named wdlist.txt was created containing the words that make up the vocabulary. Few lines are as given below.

agle

bad

birth

chaubis

december

dibba

dwara

hemgiri

hosakta

kaunsi

ke

panch

pashim

sent-end

sent-start

sham

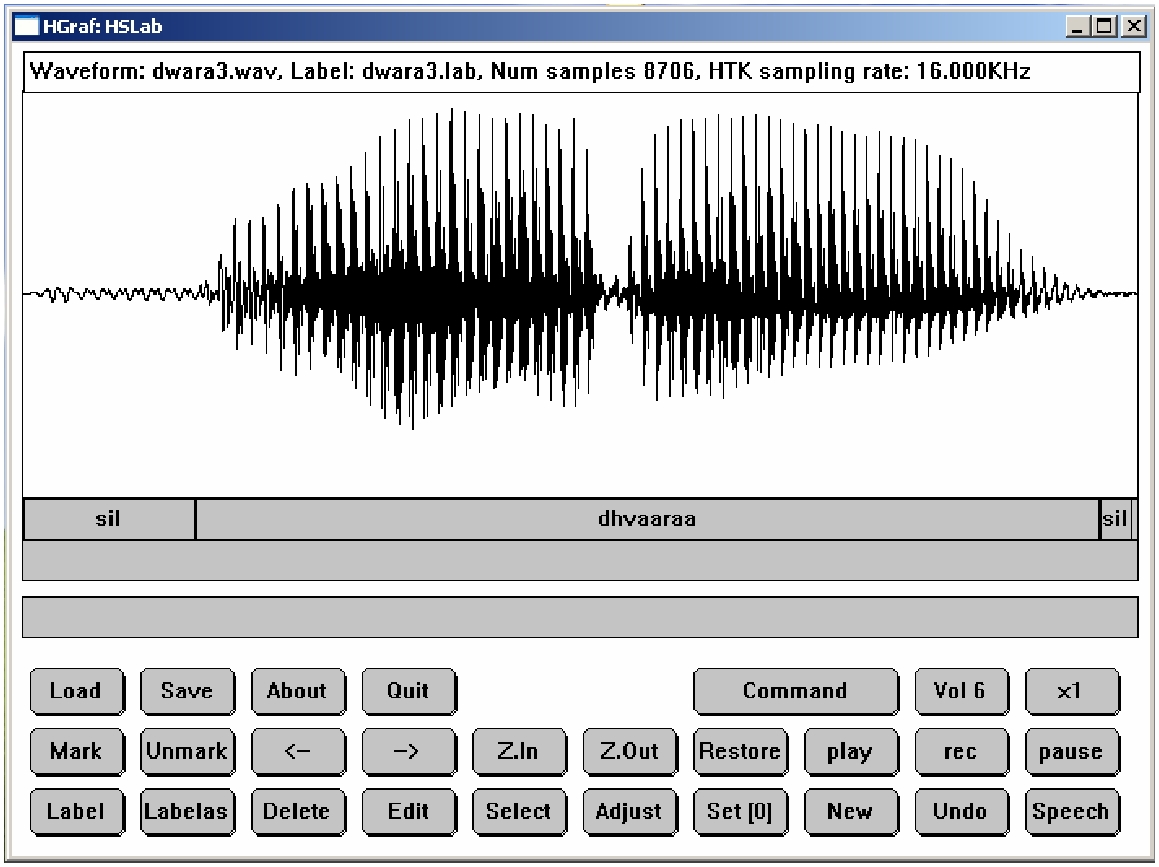
The dictionary was created finally by using

HDman as follows HDman -m -w wdlist.txt -n models1 -1 dlog dict lexicon

This will create a new dictionary called dict by searching the source dictionary(s) lexicon to find pronunciations for each word in wdlist.txt. Here, the wdlist.txt in question needs only to be a sorted list of the words appearing in the task grammar given above. The option -1 instructs HDMan to output a log file dlog which contains various statistics about the constructed dictionary. In particular, it indicates if there are words missing. HDMan can also output a list of the words used, here called modelsl. Once training and test data has been recorded, an HMM will be estimated for each of these words.

3.1.3 Recording

The toolkit learns to recognise the words through fitting the word transcriptions on the training set. These transcriptions are used for all realisations of the same sentence, even though there might be variation between speakers relative to the transcription. I have used(.wav) audio file format . Sampling rate is 16 KHz. The training corpus was labeled using the HTK tool HSLAB.


Waveform: dwara3.wav, Label: dwara3.lab, Num samples 8706, HTK sampling rate: 16.000KHz

3.1.5 Encoding the Data

The speech recognition tools cannot process directly on speech waveforms. These have to be represented in a more compact and efficient way. This step is called" acoustical analysis":

The signal is segmented in successive frames (whose length is chosen between 20ms and 40ms, typically), overlapping with each other. Each frame is multiplied by a windowing function (e.g. Hamming function).

A vector of acoustical coefficients (giving a compact representation of the spectral properties of the frame) is extracted from each windowed frame. Here Mel Frequency Cepstral Coefficients (MFCC) has been used.

In order to specify to HTK the nature of the audio data (format, sample rate, etc.) and feature extraction parameters (type of feature, window length, pre--emphasis, etc.), a configuration file (config. txt) was created as follows:

```
# Coding parameters

SOURCEFORMAT = WAVE

TARGETKIND = MFCC_0

WINDOWSIZE = 250000.0

TARGETRATE = 100000.0

NUMCEPS = 12

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 26

CEPLIFTER = 22
```

To run a HCopy a list of each source file and its corresponding output file was created. The first few lines of hcopy.scp are like below.

```
data/train/wav/panch1.wav       data/train/mfcc/panch1.mfc
data/train/wav/panch2.wav       data/train/mfcc/panch2.mfc
data/train/wav/panch3.wav       data/train/mfcc/panch3.mfc
data/train/wav/panch4.wav       data/train/mfcc/panch4.mfc
data/train/wav/panch5.wav       data/train/mfcc/panch5.mfc
data/train/wav/panch6.wav       data/train/mfcc/panch6.mfc
data/train/wav/panch7.wav       data/train/mfcc/panch7.mfc
data/train/wav/panch8.wav       data/train/mfcc/panch8.mfc
data/train/wav/panch9.wav       data/train/mfcc/panch9.mfc
data/train/wav/panch10.wav      data/train/mfcc/panch10.mfc
data/train/wav/panch11.wav      data/train/mfcc/panch11.mfc
data/train/wav/panch12.wav      data/train/mfcc/panch12.mfc
```

One line for each file in the training set. This file tells HTK to extract features from each audio file in the first column and save them to the corresponding feature file in the second column. The command used is:

```
HCopy -T 1 -C config.txt -S hcopy.scp
```

3.2 Parameter Estimation (Training)

Defining the structure and overall form of a set of HMMs is the first step towards building a recognizer. The second step is to estimate the parameters of the HMMs from examples of the data sequences that they are intended to model. This process of parameter estimation is usually called training. The topology for each of the hmm to be trained is built by writing a prototype definition. HTK allows HMMs to be built with any desired topology. HMM definitions can be stored externally as simple text files and hence it is possible to edit them with any convenient text editor. With the exception of the transition probabilities, all of the HMM parameters given in the prototype definition are ignored. The purpose of the prototype definition is only to specify the overall characteristics and topology of the HMM. The actual parameters will be computed later by the training tools. Sensible values for the transition probabilities must be given but the training process is very insensitive to these.

An acceptable and simple strategy for choosing these probabilities is to make all of the transitions out of any state equally likely. In principle the HMM should be tested on a large corpus containing wide range of word pronunciations.

3.2.1 Training Strategies

The tool HCompV is used.

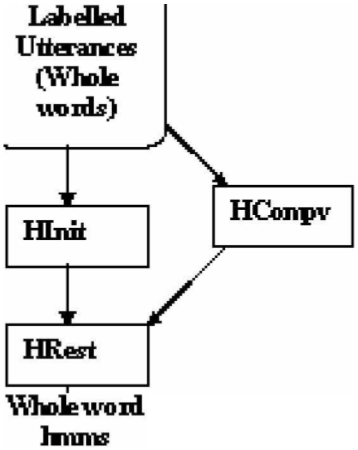


Figure 3.6: Training isolated whole word models

The following command line is used.

```
HCompV -C config.txt -f 0.01 -m -8 train.scp -M hmm0 proto.txt
```

The tool HCompv will scan a set of data of data files , compute the global mean and variance . The file train.scp contains all the training files. The above command will create a new version of proto in the directory hmm0.

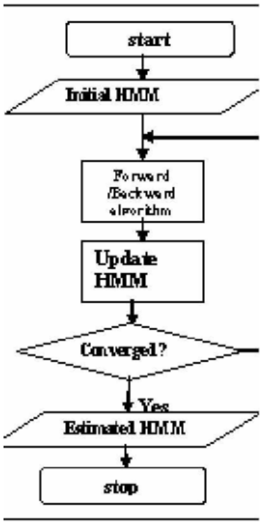
3.2.2 HMM Definition.

The first step in HMM training is to define a prototype model. The purpose of the prototype is to define a model topology on which all the other models can be based. In HTK a HMM is a description file and in this case it is

```
~o

 <VecSize> 13

  <MFCC_0>

~h "proto"

<BeginHMM>

 <NumStates> 5

<state> 2
   <mean> 13
    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> 13
    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
   <mean> 13
    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> 13
    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>  4
   <mean> 13
    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> 13
    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


<TransP>   5
    0.0 1.0 0.0 0.0 0.0
    0.0 0.6 0.4 0.0 0.0
    0.0 0.0 0.6 0.4 0.0
    0.0 0.0 0.0 0.7 0.3
    0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

3.2.3 HMM Training

The training described in the parameter estimation introduction can be summarized in a diagram form as below.



Initialisation

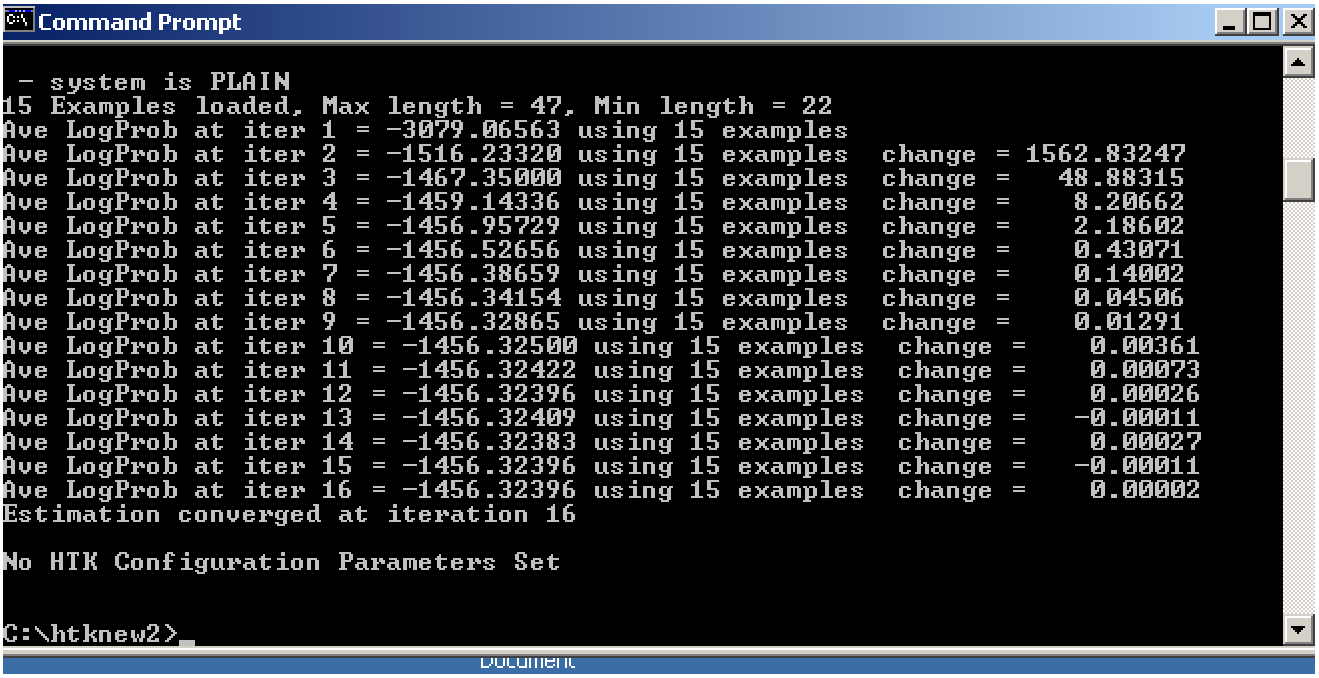
The HTK tool HComp V was used to initialize the models to the training data as follows.

hcompv -A -D -T 1 -S trainlistj18.txt -M model/hmm0flat -H model/proto/hmm_paanch -f 0.01 paanch

3.2.4 Training

The following command line was used to perform one re-estimation iteration with HTK tool HRest, estimating the optimal values for the HMM parameters (transition probabilities, plus mean and variance vectors of each observation function):

HREST -A -D -T 1 -S trainlistjul26.txt -M model/hmm1 -H macrojul35.txt -H model/hmm0flat/hmm_chawbiys.txt -l chawbiys -L data/train/lab/ chawbiys



```
 - system is PLAIN
15 Examples loaded, Max length = 47, Min length = 22
Ave LogProb at iter 1 = -3079.06563 using 15 examples
Ave LogProb at iter 2 = -1516.23320 using 15 examples  change = 1562.83247
Ave LogProb at iter 3 = -1467.35000 using 15 examples  change =   48.88315
Ave LogProb at iter 4 = -1459.14336 using 15 examples  change =    8.20662
Ave LogProb at iter 5 = -1456.95729 using 15 examples  change =    2.18602
Ave LogProb at iter 6 = -1456.52656 using 15 examples  change =    0.43071
Ave LogProb at iter 7 = -1456.38659 using 15 examples  change =    0.14002
Ave LogProb at iter 8 = -1456.34154 using 15 examples  change =    0.04506
Ave LogProb at iter 9 = -1456.32865 using 15 examples  change =    0.01291
Ave LogProb at iter 10 = -1456.32500 using 15 examples  change =    0.00361
Ave LogProb at iter 11 = -1456.32422 using 15 examples  change =    0.00073
Ave LogProb at iter 12 = -1456.32396 using 15 examples  change =    0.00026
Ave LogProb at iter 13 = -1456.32409 using 15 examples  change =   -0.00011
Ave LogProb at iter 14 = -1456.32383 using 15 examples  change =    0.00027
Ave LogProb at iter 15 = -1456.32396 using 15 examples  change =   -0.00011
Ave LogProb at iter 16 = -1456.32396 using 15 examples  change =    0.00002
Estimation converged at iteration 16

No HTK Configuration Parameters Set


C:\htknew2>_
```

3.3 Recognition

The recognizer is now complete and its performance can be evaluated. The recognition network and dictionary have already been constructed, and test data has been recorded. The following command line was used.

```
hvite -A -D -T 1 -H hmm_defs2.txt -i recosham16.mlf -w wdnetjul2345 dictjul23455
hlist2.txt   sham16.mfc
```

The input observation was then processed by a Viterbi algorithm, which matches it against the recogniser's Markov models using the HTK tool HVite: As follows

```
HVite -A -D -T 1 -H modelfhmm3/hmmdefs.txt -i recout.mlf -w wdnet dict hmmlist.txt -
S test.scp.
```

Where:

v. The built system will be very useful to computer manufactures and software developers as they will have a speech recognition engine to include Hindi language in their applications.

Template based approaches matching :-Unknown speech is compared against a set of pre-recorded words(templates) in order to find the best match. This has the advantage of using perfectly accurate word models. But it also has the disadvantage that pre-recorded templates are fixed, so variations in speech can only be modelled by using many templates per word, which eventually becomes impractical.

Dynamic time warping is such a typical approach . In this approach, the templates usually consists of representative sequences of features vectors for corresponding words. The basic idea here is to align the utterance to each of the template words and then select the word or word sequence that contains the best. For each utterance, the distance between the template and the observed feature vectors are computed using some distance measure and these local distances are accumulated along each possible alignment path. The lowest scoring path then identifies the optimal alignment for a word and the word template obtaining the lowest overall score depicts the recognised word or sequence of words.

2. Sub-word matching. The engine looks for sub-words - usually phonemes - and then performs further pattern recognition on those. This technique takes more processing

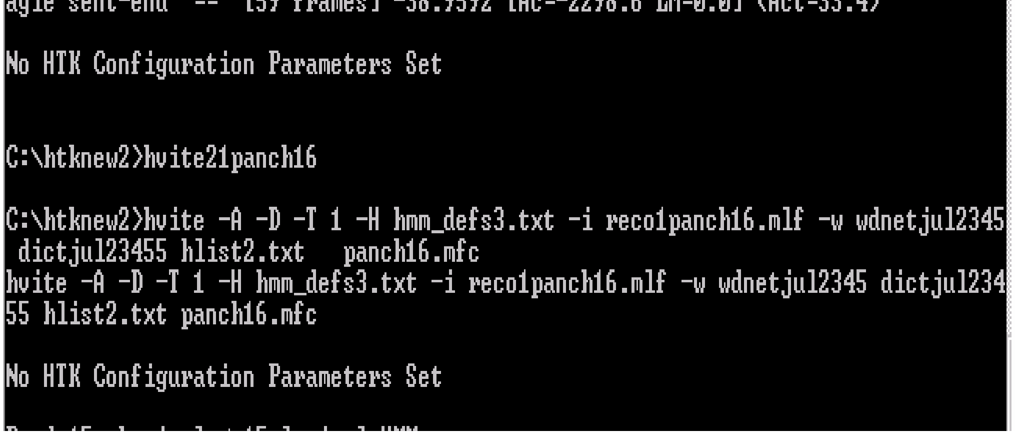
than whole-word matching, but it requires much less storage . In addition, the pronunciation of the word can be guessed from English text without requiring the user to speak the word beforehand.

To build any speech engine whether a speech recognition engine or speech sythensis engine you need a corpus. Corpora are any collections of text and/or speech, and are used as a basis of statistical processing of natural language **.** There are many kinds of corpora. For example, one of the largest and best-known corpora, the British National Corpus and Timit.

ASR has been proved to be a not easy task. According to the main challenge in the implementation of ASR on desktops is the current existence of mature and efficient alternatives, the keyboard and mouse. In the past years, speech researchers have found several difficulties that contrast with the optimism of the first speech technology pIOneers. Problems in designing ASR are due to the fact that it is related to so many other fields such as acoustics, signal processing, pattern recognition, phonetics, linguistics, psychology, neuroscience, and computer science. And all these problems can be described according to the tasks to be performed.

An input speech to be recognized is first transformed into a series of "acoustical vectors" (here MFCs) using the HTK tool HCopy, in the same way as what was done with the training data. The result was stored in a file known as test.scp (often called the acoustical observation).

hmmdefs.txt contains the definition of the HMMs. It is possible to repeat the -H option and list the different HMM definition files, in this case: -H modelfhmm3/hmm_0.txt -H modelfhmm3/hmm_1.txt etc .. but it is more convenient (especially when there are more than 3 models) to gather every definitions in a single file called a Master Macro File.



```
agle sent-end  --  [57 frames] -58.9592 [Ac=-2278.6 LM=0.0] (Act=33.4)

No HTK Configuration Parameters Set


C:\htknew2>hvite21panch16

C:\htknew2>hvite -A -D -T 1 -H hmm_defs3.txt -i reco1panch16.mlf -w wdnetjul2345
 dictjul23455 hlist2.txt   panch16.mfc
hvite -A -D -T 1 -H hmm_defs3.txt -i reco1panch16.mlf -w wdnetjul2345 dictjul234
55 hlist2.txt panch16.mfc

No HTK Configuration Parameters Set
```

For this project this file was obtained by copying each definition after the other in a single file, without repeating the header information (see Appendix E).

The output is stored in a file (recout.mlf) which contains the transcription of the input

recout.mlf is the output recognition transcription

file. W dnet is the task network.
dict is the task dictionary.

hmmlist.txt lists the names of the models to use . Each element IS separated by a new line character. Test.scp is the input data to be recognised. We will get in reco.mlf an output as.

```
#!MLF!#
"agle10.rec"
0 5800000 axgley -2198.147217
5800000 6500000 sil -299.778961
.
HResult tool evaluates recognition performance.
```

Chapter 4

4.1 What is New

1. New Improved Technique for Mel Frequency Cepstral Coefficients(MFCC) :

Standard deviation was calculated for mel cepstral coefficients and new improved coefficients were calculated as below.

Std = $\sqrt{\sum x^2/n}$

Where x is deviation from mean. Value of mel cepstral coefficients.

$$\text{New coefficients} = \frac{\text{old value} - \text{std}}{\text{Std}}$$

2. A labeling program was developed.

RESULTS

The recognition performance evaluation of an ASR system must be measured on a corpus of data different from the training corpus. A separate test corpus, with Hindi language was created as it was previously done with the training corpus. The following results were observed.

REGNITION SCORE TABLE

Data	Trained data	Test data
MFCC_0	**80% to 100%**	**30% to 100%**
New MFCC	**90 % to 100%**	**40% to 100%**

Analysis of the results:

1.Retroflexive sounds like ataragh needs more training samples than others for hmm initialization and training..
2.The word bad was recognized for birth.
3. The word agle has highest recognition score.
4.Panch has low recognition score
5.The word sham was recognised for others words most of the times.

4.2 Conclusion

The main purpose of this study is to perform speech recognition for isolated words in Hindi language. The railway speech corpus has been included for the study.

In order to meet this objective a limited word grammar was constructed, a dictionary created and data from railway corpus was taken and trained thereafter.

The system was tested using testing corpus data and training data and the system scored recognition rate with exististing mfcc and new improved mfcc.The recognition score was improved by 10% with new mfcc technique. This implies that the objective of creating a system that can perform speech recognition for Hindi language was achieved.

The Hindi language automatic speech recognition method accompanying this report can be used by any researcher desiring to join language processing research.

The project is however not all conclusive as it has catered for only a voice operated railway system. As much as it has created a basis for research, this project can be expanded to cater for more extensive language models and larger vocabularies.

4.3 Areas for Further Study

The developed system can be used by researchers interested in the field of Hindi language speech recognition. The findings of the study can be generalized to cater for large vocabularies and for continuous speech recognition

References

1. Fundamental of speech recognition by Rabiner

2. Isolated and connected word recognition theory and selected applications.

   IEEE transaction on communication. Vol. COM 29 No. 5 May 1981

3. A large vocabulary continuous speech recognition system for hindi IBM India

   Research lab, Chalapathy Neti, Nitendra Rajput, Ashish Verma

4. Speech and audio signal processing by Ban Gold and Nelson Morgan

5. Digital waveform processing and recognition by C.H. Chen PHD CRC

6. Hindi speech recognition using Neural networks. International conference on cognitive systems. Page 134-140, Dec 15th -17th 1999. Dev A, Agrawal SS

7. Hindi speech database, Samudra Vijay K., P.V.S. Rao and S.S. Agrawal, 6th international conference on spoken language processing (ICSLP-2000) Beijing China October 16-20, 2000

8. A review of large vocabulary continuous speech recognition. Steve Young, IEEE signal processing magazine, September 1996

9. Te HTK Book by Microsoft/ Cambridge University Engineering Department

10. Fundamental of computer algorithms by Ellis Harowitz, Sartaj Sahni

11. Speech recognition, theory and C++ implementation, Claudio Becchetti and Lucio Prina Ricotti

12. Digital speech processing by Rabiner

13. Comparison of software for transcription of speech data, NSA 2003-030 Samudra Vijay K.

14. Durational characteristic of hindi phoneme in continuous speech by Samudra Vijay K.
15. Statistical Method for Speech Recognition Frederick Jelinek
16. Speech and Speaker Recognition Manfred R Schroeder Gottingen
17. Electronic speech

List of words

1 vxys
2 vBkjg
3 cjFk
4 fnLEcj
5 ckn
6 fMCck
7 }kjk
8 “kke
9 dSa l y
10 gks l drk
11 ds
12 if pe
13 pkSch l
14 ikap
15 fgefxjh
16 dkSu lh
17 c|fdax