International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 9 - No. 4, July 2015 - www.ijais.org

Speech Recognition System for North-East Indian Accent

Moirangthem Tiken Singh
Institute of Engineering and Technology
Dibrugarh University

Abdur Razzaq Fayjie
Institute of Engineering and Technology
Dibrugarh University

Biswajeet Kachari
Institute of Engineering and Technology
Dibrugarh University

ABSTRACT

Speech recognition is the process of converting an acoustic waveform into the text containing the similar information conveyed by the speaker. This paper presents a speech recognition system for English digits in Indian (especially North Eastern) accent. Hidden Markov Model Tool kit (HTK-3.4.1) is chosen to implement the Hidden Markov Model as classifier with several set of Hidden Markov Model mixture. Mel Frequency Cepstral Coefficients are used as speech features. Experiments were performed for data collected in natural noise environment. The performance is evaluated using recognition rate. Hidden Markov Model state numbers and number of mixtures are investigated and possible directions for future research work are suggested.

General Terms:

North-East Indian Ascent, Automatic Speech Recognition

Keywords

Regional accent, HMM, HMM mixture, MFCC

1. INTRODUCTION

Speech is the natural and vocalized form of human communication. For a human-machine interaction, ASR (Automatic Speech Recognition) is considered as an alternative to hardware interfaces like keyboard, mouse etc. Speech recognition is a field of computer studies, aims to design computer system that can recognize human voice. Automatic speech recognition came into existence in speech recognition field after TTS (text-to-speech) supporting interactive voice response (IVR) systems. ASR takes an utterance of speech signal as input, captured by a microphone, a telephone etc. and convert it into a text sequence as close as possible to spoken data[1]. ASR was first introduced during 1950s. The first attempt (during the 1950s) to develop techniques for speech recognition, which were based on the direct conversion of speech signal into a sequence of phoneme-like units, failed. The first positive results of spoken word recognition came into existence in the 1970s, when general pattern matching techniques were introduced[2]. ASR has attracted much attention over the last three decades and has witnessed dramatic improvement in the last decade. Today it has different areas of application like dictation, program controlling, automatic telephone call, weather report information system, travel information systems etc. But its implementation is difficult due to the different speaking styles of human beings (i.e. the accents). Therefore the main aim of ASR today is to transform an input voice signal to its corresponding text output independent of speaker or device. In the world, many research groups such as HCI group at Stanford University, the Human-Computer Interaction group at Microsoft Research Asia (MSRA HCI), The Human-Computer Interaction (HCI) group at Department of Computer Science, University of Toronto are working in the field of speech recognition and interaction[3]. Most of the work is about English, but due to the difference in British, American or Australian accents with North-Eastern accents of India, it fails here. Feeling the importance of speech recognition in regional accent, especially for the rural areas of North-East India, this paper aims to design a speech digit recognizer about English digit (with regional accent) based on Hidden Markov Model using Hidden-Markov Tool Kit under Linux environment. This paper also aims for a comparative study based on speaker dependent and speaker independent speech recognition system.

2. LITERATURE REVIEW

Yanli Zheng[6] and his team worked on speech recognition for Chinese language with different dialects including Mandarin, Wu (spoken by Shanghainese), Yue (spoken by Cantonese), Min (spoken by Taiwanese). They combined MLLR (Maximum Likelihood Linear Regression) with the MAP (Maximum A Posteriori), optimized and adapted individually to each test speakers. Based on this optimal MAP/MLLR combination, they developed new approaches to detecting and utilizing degree of accent in accented ASR. A series of new algorithms is proposed: phoneme-based automatic accent detection, formant-augmented acoustic features for accented speech, and accent-based model selection during acoustic model decoding. Different groups of researchers have been working on accent based speech recognition using Hidden Markov Model also. Elitza Ivanova, Sara Kazemi[7] and their group worked on American and Chinese spoken English at San Diego state university using HMM and HTK. Konstantin Markov and Satoshi Nakamura[8] used HMM with Bayesian Network for accent based speech recognition. In India, A. N. Mishra[9] and his team worked on speaker independent connected digits with Revised perceptual linear prediction, Bark frequency cepstral coefficients and Mel frequency cepstral coefficients to describe the robust features for digit recognizing both in noisy and clean environment based upon Hidden Markov Model and using Hidden Markov Tool Kit for MFCC. All other features are extracted using Matlab and saved in HTK format. The result shows features extraction with MFCC having an accuracy rate of 98% for clean data which is 1% lesser than MF-PLP. A.N. Mishra along with Astik Biswas and Mahesh Chandra[4] again produced a comparative study on isolated digit recognition in Hindi implementing HMM and using MFCC for features extraction. They performed experiments using both HTK and Matlab, where HTK gave an accuracy of 99-100% for clean data that is 5-6% better than Matlab. In noisy environment, HTK excels with an accuracy rate of 89-94% compared to Matlab. Ganesh S. Pawar and Sunil S. Morade[2] designed a digit recognition system for isolated English digits with a huge database of 50 speakers using HMM as classifier and MFCC as features ex-

traction algorithm. HTK is used in training and testing purposes. The system came up with 95% of accuracy in recognizing digits for speaker dependent behaviour. Maruti Limkar[10] proposed an approach to speech recognition for isolated English digit using MFCC and DTW(Dynamic time wrapping) algorithm which gave a result with accuracy rate 90.5%. This paper provides a comparative study on the speaker dependent and speaker independent speech recognition by the designed digit recognition system using Hidden Markov Tool Kit based on Hidden Markov Tool. HTK is used for its better accuracy of speech recognition compared to Matlab. Also HTK is open-source and easily available while Matlab is a commercial product. MFCC is chosen for features extraction as it is the most widely used and one of the effective features extraction algorithms. HMM is preferred as classifier over DTW algorithm as the former is easier and brings more accuracy in recognition. The accuracy of both the speaker dependent and speaker independent is compared and it is also compared to some of the previously known results[4].

3. STATISTICAL FRAMEWORK OF ASR

Automatic speech recognition is performed in two major steps-features extraction and training-testing data. The recognition performance heavily depends upon the features extraction of input signal. Automatic speech recognition mainly comprises of

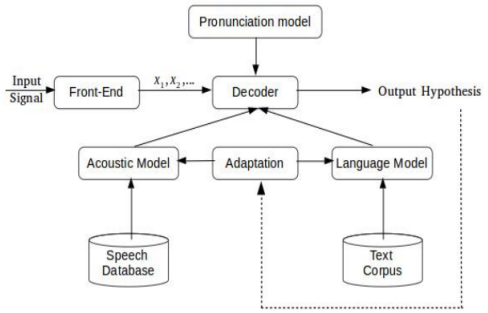

Fig. 1. Block diagram of ASR

five parts Acoustic analysis for features extraction, Acoustic model, Language model, Pronunciation dictionary and decoder for recognition. Speech captured by a microphone at the front end is passed through acoustic analysis where sound is converted into digital form with a series of feature vectors and then forwarded to the speech decoder. Decoder with speech database in acoustic model and pronunciation dictionary produce outputs in text format by Language model. This whole process of speech recognition is done in four steps- features extraction using MFCC or LPC or other algorithms (using MFCC), acoustic scoring with Gaussian Mixture Model (GMM's), Modelling of sequence is done with static or dynamic model like Hidden Markov Model, Dynamic Time wrapping etc. (Using static HMM), generating the competitive hypothesis (based on pronunciation dictionary, acoustic model and language model) using the score of database and selecting the best as final output[11]. Figure 1 represents the statistical framework for ASR.

4. INDIAN ENGLISH PHONOLOGY

Indian accents for English vary from state to state. From the southern part of India to the northern part, accents vary accordingly and significantly. Some Indians speak with an accent closer to British English while most of them lean toward a more 'Vernacular' native-tinted accent for their English speech.

4.1 Vowels

Many Indian English speaker especially in north-eastern part of India, do not make a clear distinction between /ɒ/. and /ɔː/ (eg. cot and caught). Most of north-eastern Indian English speaker pronounced /ə/ as /ɛ/ (eg. chiken), /æ/ as /a/ (eg. fan), /oo/ is pronounced as /u/ (eg. school) etc. In India, the speech distribution varies from American, British to Australian English but it has a complete split in Cultivated Indian English and Standard Indian English.

4.2 Consonants

Pronunciation of consonants in India vary between rhotic and non-rhotic, with pronunciation leaning towards native phonology being generally rhotic and others being non-rhotic, imitative of British pronunciation. Speakers from north-eastern part of India generally differentiate between /v/ and /w/ where a group of speakers pronounce it as /ʋ/ for the both consonants (eg. wet and vet becomes homophones for such speakers). The voiceless plosive /p/,/t/,/k/ are not aspirated in Indian English whereas in other English accents (American, British etc.), they are aspirated in word initial or stressed syllables. Thus 'Pin' is pronounced as [Pɪn] in Indian English but [P^hɪn] in other dialects. Some speakers in these part use /z/ or /dʒ/ and some use/ʃ/ for the consonant 's' like in the word 'Treasure' (trɛzəːr or trɛʃər). In the north, English accent lack the dental fricatives (/θ/ and /ð/). Usually the aspirated voiceless dental plosive [t^h] is substituted for /θ/ and possibly aspirated version [d^h] is substituted for /ð/. Thus 'Thin' would be pronounced as [t^hɪn] instead of /θɪn/. Most Indian languages lack the voice alveolar fricative /z/. A significant portion of Indians thus, even though their native languages do have its nearest equivalent: the unvoiced /s/, often use the postalveolar /dʒ/. This make the word 'Zero' sound as [dʒiːro]. Many speaker again use /f/ and /p^h/ interchangeably. When retaining /ŋ in the final position/, many Indian speakers add the [g] sound after it when it occurs in the middle of the word. Hence ringing that is /riŋiŋ/ changed to /riŋgiŋ/. Syllabic /l/, /m/, /n/ are usually replaced by the VC clusters [əl], [əm] and [ən]. (e.g., metre that is /miːtər/ is pronounced as /miːtər/). Unlike the other English accent, Indian speaker uses clear /l/ in all instances.
British phonology for English is considered as one of the correct pronunciation over American or Australian English phonology. British pronunciation for digits 0 to 9 is given in the Table 1[1]

Table 1. Phonology of English digits.

digit	phonology	digit	phonology
0	/ˈzɪərəʊ/	5	/fʌɪv/
1	/wʌn/	6	/sɪks/
2	/tuː/	7	/sɛv(ə)n/
3	/θriː/	8	/eɪt/
4	/f ɔː/	9	/nʌɪn/

The above pronunciation varies from speaker to speaker in north-eastern part of India. In general, /r/ is also pronounced in four by the speakers here. It also becomes very difficult to differentiate one from nine (/wʌn/ and /nʌɪn/) for these speakers. A group of speakers pronounce /t/ in eight as /θ/ . These are some basic differences in Indian English phonology, due to which most speech recognition system fails here.

5. DIGIT SPEECH RECOGNITION SYSTEM

The recognizer for digit speech recognition is implemented using Hidden Markov Tool Kit version 3.4.1 under Ubuntu 14.04

[1]based on Oxford dictionary on `http://www.oxforddictionaries.com/definition/english`.

LTS Linux environment. The system is trained with different permutations of dataset and tested accordingly. Wavesurfer (version 1.8.8) is used to record voice samples both for male and female speakers.

5.1 Procedure

The design of the speech digit recognizer mainly involves four steps: database preparation, features extraction, Training and Testing. The recognition performance basically depends on the performance of the feature extraction block. Thus choice of features and its extraction from the speech signal should be such that it gives high recognition performance with reasonable amount of computation[9]. In this research, Mel-Frequency Cepstral Coefficient is used for the features extraction phase. Training module leads to the generation of Hidden Markov Model from where recognition module starts in Testing phase. The testing phase can acquire accuracy of recognition of speech digits and their recognition level. Figure 2 describes the steps involved in the design of digit speech recognition. Database preparation involves the

Fig. 2. Steps involved in designing of digit recognizer

recording of voice data by different speakers. MFCC algorithm is applied to extract features from the data recorded and the system tries to bring out equivalent data meaning from it. System is trained with a selected dataset and testing involves the process of acquiring results from the system.

6. DATABASE

The speech data is recorded with 4 speakers (3 male speakers and a female speaker) for training and testing the acoustic model in natural noise environment. The recording is done with respective noise in a room with a door and 5 (3+2) windows fixed in two walls. The respective noise refers to internal background noises and silence in the room. All speakers are of an age group from 18 to 24 years. The utterances are recorded using 2-channel with sample frequency of 48 KHz and bit rate of 16 bits per sample. Chipping is used in segmentation of data sample manually to get the desired 10 digits length and after removing unwanted information, the recorded sample is stored accordingly.

6.1 WaveSurfer

WaveSurfer[2] (version 1.8.8) is an open source for sound visualization and manipulation. It can be used by both novice and advanced users. It has a simple and logical user interface that provides functionality in an intuitive way that makes the tool easy to use and easy to understand. It is generally used as a stand-alone tool for a wide range of tasks in speech research and education. Typical applications are speech and sound analysis, sound annotation and transcription. It can be extended with custom plug-ins or by embedding it to other applications. It is a Tcl/Tk application that uses Snak sound toolkit. It is written in C for performance reasons. WaveSurfer can be extended in several ways. For example, using a combination of C or C++ code and Tcl. Wavesurfer comes with the following features-
Customizable: users can customize configuration as their own.
Extensible: new functionality can be added through plugin architecture.
Embedded: it can be used as a widget in custom applications.

[2]The too is available at `http://www.speech.kth.se/wavesurfer/`

Transcription file formats: reads and writes HTK (and MLF), TIMIT, ESPS or Waves+ and Phondat. It also support for encodings and Unicode.
Multi-platform: it can be run in Linux, Windows and OSX.

7. FEATURES EXTRACTION

Features extraction is the extraction of specific features from input speech, where features carry the characteristics of the speech which are different from one speaker to another. To understand features properly , it is need to focus on the following points- a) What do the features represent? - Features represent features vectors. b) What the features vectors represent? - Features vectors represent formants that carry the identity of the speech. c) What are formants? - To identify the formants, it is needed to know about the speech production in the human body. In human body, the three main cavities of speech production system are nasal, oral and pharyngeal forming the acoustic filter. The form and shape of the vocal and nasal tracts change continuously with time, creating an acoustic filter with time varying frequency response. As air from the lungs travels through the tracts, the frequency spectrum is formed by the frequency selectivity of these tracts. The resonance frequencies of the vocal tract tube are called formant frequencies or simply formants, which depend on the shape and dimension of the vocal tract. The speech by which the cords open and close is unique for each individual and define the feature and personality of the particular voice. Figure 3 illustrates the formants enveloped by the red smooth curve.

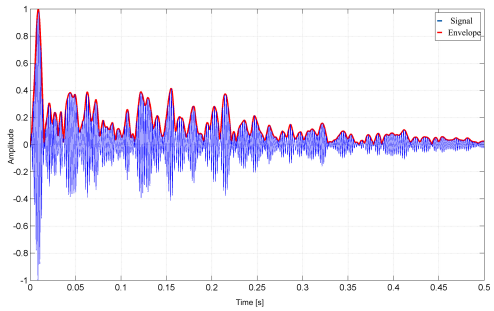

Fig. 3. The envelope connecting the formants

Figure 4 describes about the envelopes connecting the formants in features extraction. The Spectrum is obtained during the analysis of a wave file with digits 1, 0, 2 in WaveSurfer.

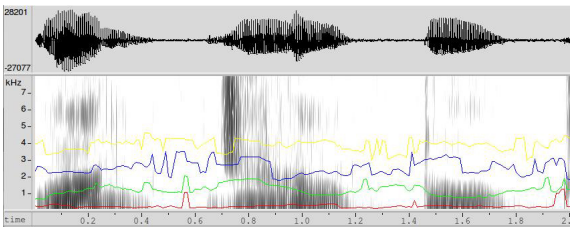

Fig. 4. Formants for digit 1, 0, 2 at four different frequency level

The formants are calculated at frequency level of 1 KHz, 2 KHz, 3 KHz and 4 KHz respectively and the obtained envelopes are represented by the four primary color.

7.1 Characteristics of formants

(1) Easily measurable.

(2) Vary among the speakers but consistent with each speaker.

(3) Hardly change over time or hardly effected by speakers health.

(4) Not effected by background noise nor depend on specific transmission medium.

(5) Occur naturally and frequently in speech.

7.2 MFCC Algorithm

Algorithms that can be applied to extract features from speech wave forms are mainly- linear predictive coefficients (LPC), linear predictive cepstral coefficients (LPCC), mel-frequency cepstral coefficients (MFCC) and human factor cepstral coefficients (HFCC). Here mel-frequency cepstral coefficients (MFCC) is used for features extraction, for the following reasons-

(1) It is the most common algorithm used for speech recognition.

(2) It shows high accuracy results.

(3) It can be regarded as standard features in speaker as well as in speech recognition. This coefficients are best for discriminating speakers.

Figure 5 describes the features extraction by using Mel Frequency Cepstrum Coefficient algorithm.

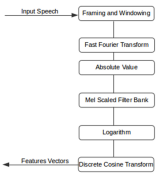

Fig. 5. Mfcc algorithm flowchart

7.3 Preprocessing

Speech signals are generally preprocessed before extracting the features to enhance the accuracy and efficiency of extraction. Speech signal preproceesing generally covers digital filtering and speech signal detection. Filtering includes pre-emphasis filter that means the removal of surrounding noises associated with the speech.

7.4 Pre-emphasis

The digitized speech waveform suffers from additive noise. It also has a high dynamic range. To reduce the noise and the range to make the frequencies sharper, pre-emphasis is applied. The input speech signal s[n] is passed through a high pass filter (FIR, first order) with an equation-

$$s[n] = s[n] - a.s[n-1] \quad (1)$$

where $s[n]$ is the output signal and $0.9a1$. Pre-emphasis can be implemented with fixed coefficient filter or as an adaptive one, where the coefficient a is adjusted with time according to autocorrelation values of the speech. This step boost the amount of energy in high frequencies. The drop of energy across frequencies (known as spectral tilt) is caused by glottal pulse. It results in higher formants available to acoustic model.

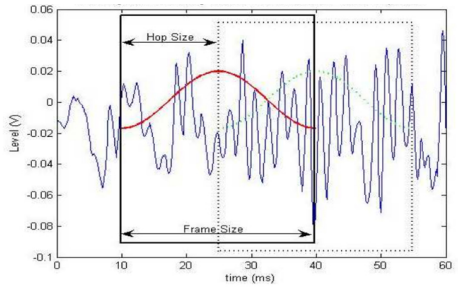

Fig. 6. Framing and Windowing

7.5 Framing and windowing

To analyze a speech signal more accurately, some frames are implemented with a short time range instead of analyzing the whole signal at once. This time range can vary from user to user but in general it is taken as 10-30 ms. Again an overlapping is applied to frames with a definite Hop size. In general, hop size is equal to half of the frame size. This is done because when it applies a hamming window, then some information may not be covered at beginning and the end of each frame. Overlapping results in regain such information into the extracted features. Figure 6 describes the framing and windowing with 30 ms frame size and 15 ms hop size.

7.6 Windowing

Windowing is performed to avoid unnatural discontinuity in speech segment and distortion in underlying spectrum. In windowing each data frame is multiply with a windowing function to keep continuity of the last point of the first frame and the starting point of the next. If the signal in a frame is denoted by$s[n]$, $n = 0, 1, 2....n-1$; then signal after windowing is $s[n]w[n]$, where $w[n]$ is the windowing function. The multiplication of signal waveform with windowing function has two effects-

(1) It preserves the amplitude in both the ends of a frame by preventing the abrupt change in the endpoints.

(2) It produce convolution for Fourier Transform of window function and speech spectrum.

There are several types of windows such as Hann window, Cosine window, Triangular window, Gauss window, Kaiser and Blackman window, Welch window, Hamming window etc. Humming window is widely used in speech recognition system. Humming window is used due to the fact that mfcc will be used which involves the frequency domain. It decreases the possibility of high frequency components in each frame due to such abrupt slicing of the signal. Hamming window W_H is defined as-

$$W_H(n) = 0.54 - 0.46cos(2n\pi/N-1) \quad (2)$$

7.7 Fast Fourier Transform

To convert the signal from time domain to frequency domain, fast Fourier transform is applied. The basis of performing fast Fourier transform is to convert the convolution of the glottal pulse and vocal tract impulse response in the time domain into multiplication in the frequency domain.

7.8 Mel Scaled Filter Bank

The speech signal consists of tones of frequencies. For each tone, an actual frequency f measured in H_z, a subjective pitch is measured on a particular scale known as Mel scale. The mel frequency scale is a linear frequency spacing below $1000H_z$ and logarithmic spacing above $1000H_z$. Following equation gives the mels for a particular frequency f-

$$2595 \times log_{10}(1 + f \div 700) \quad (3)$$

Mel Frequency Analysis which is based on human perception experiments. Human ear hears tones with frequencies lower than 1 KHz in linear scale instead of logarithmic scale for frequencies higher than 1 KHz. The information carried by lower frequencies is more important compared to information associated with higher frequencies. Mel scale is performed to emphasize on the low frequency components. Mel filter banks are non-uniformly distributed in frequency regions; more filter banks on low frequency region and less number of filter banks on high frequency regions. After having the spectrum (FFT for the windowed signal) mel filter banks are applied, the signal processed in such a way like that of human ear response.

$$S(l) = \sum_{k=0}^{N/2} S(k)M(k) \tag{4}$$

Where $S(l)$ is the Mel spectrum,
$S(k)$ is the Original spectrum,
$M(k)$ is the Mel filter bank,
$N/2$ is the Half of the FFT size,
$L = 0, 1, ..., (L-1)$, Where L is the total number of mel filter banks.

7.9 Cepstrum

In this finale step, the log mel spectrum has to be converted again to time domain. The result is called Mel Frequency Cepstral Coefficients (MFCCs). The cepstal representation of speech spectrum provides a good representation of spectral properties (i.e. Features) of the signal for the given frame analysis. As Mel Frequency Cepstral Coefficients are real numbers, and so their logarithms; it can be converted to time domain by Discrete Cosine Transform (DCT). Since the speech signal represented as a convolution between slowly varying vocal tract impulse response (filter) and quickly varying glottal pulse (source), so, the speech spectrum consists of the spectral envelop(low frequency) and the spectral details(high frequency). It is known that the logarithm has the effect of changing multiplication into addition. Therefore it can simply convert the multiplication of the magnitude of the Fourier Transform into addition. Then, by taking the inverse FFT or DCT of the logarithm of the magnitude spectrum, the glottal pulse and the impulse response can be separated.

7.10 MFCC Implementation

After the discussion about mel-frequency cepstrum coefficients, it uses the following computation steps of MFCC that include-
Framing: The signal is segmented in successive frames with a frame size= 25ms overlapping with each other.
Windowing: Each frame is multiplied by a windowing function (using Hamming function).
Extracting: A vector of acoustical coefficients (giving a compact representation of the spectral properties of the frame) is extracted from each windowed frame. 12 mfc coefficients are used for each frame. MFCC is practically implemented through a configuration file (*.conf*). It is a text file that specifies the configuration parameters for the Hidden Markov Tool Kit(HTK). The parameters are such as framing size(25 ms), number of MFC coefficients (12), source kind (waveform) with source format (wav), target kind (MFCC) with target format (HTK) etc. **Hcopy** tool of HTK is used to extract mfc coefficients from their corresponding wave (*.wav*) files connecting the configuration.

8. HIDDEN MARKOV MODEL

Hidden Markov Model is very powerful and mathematical tool for modeling time series that has significantly improved the performance of current speech recognition systems. But the problem of completely fluent, speaker independent speech recognizer is still far from being solved. It provides efficient algorithm for state parameter estimation that automatically performs dynamic time warping of locally stretched signals. It is a doubly stochastic process, generated by two interrelated mechanisms- an underlying Markov chain having a finite number of states and a set of random functions, one of which associated with each state. Markov chain is deterministically an observable event. The most likely word with largest probability is produced as a result of the given speech waveform. Hidden Markov Model is the extension of Markov chain, where the internal states are hidden and any state produces observable symbols. At discrete time instances, one process is assumed to be in some state representing the temporal variability and an observation is generated by another process corresponding to the current state representing the spectral variability. These two stochastic processes are flexible enough to design a practical ASR system. But for ASR, only observed sequence of events are known and both the underlying processes are unobservable. Hence it is called 'Hidden' Markov Model[2][12].

8.1 Mathematical Formulation of HMM

Let us consider a first order N-state homogeneous Markov chain illustrated for $N = 3$ in Figure 7. The system can be described as being in one of the N (=3 here) distinct states $1, 2, 3, ..., N$ at any discrete time instant t. The Markov chain is then described by a state transition probability matrix $A[a_{ij}]$, where

$$a_{ij} = P(q_t = j | q_{t-i} = i), 1 \leq i, j \leq N \tag{5}$$

with the following axiomatic constrains:

$$a_{ij} \geq 0 \tag{6}$$

and

$$\sum_{i=1}^{N} a_{ij} = 1, \forall i \tag{7}$$

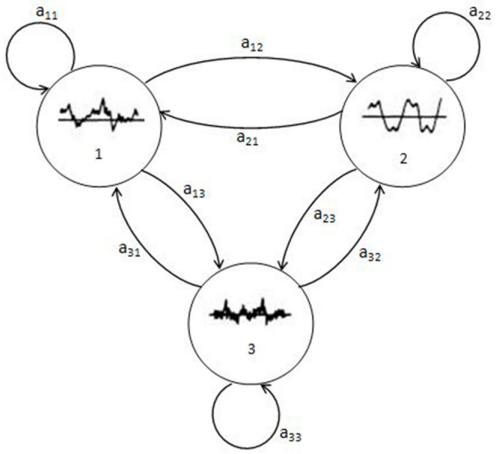

Fig. 7. A First-Order 3-State Markov Chain With Associated Processes

Due to the homogeneity of the Markov chain, transition probabilities do not depend on time. Assume that $t = 0$ the state of the system q_0 is specified by an initial state probability $\pi_i = P(q_0 = i)$. Then, for any state sequence $q = (q_0, q_1, q_2, ...q_\gamma)$, the probability of q being generated by the Markov chain is

$$P(q|A, \pi) = \pi_{q_0} a_{q_0 q_1} a_{q_1 q_2} ... a_{q_{\gamma-1} q_\gamma} \tag{8}$$

Suppose that the state q cannot be readily observed now. Instead, observe each observation O_t, say a cepstral vector as mentioned previously as being produced with the system in state $q_t, q_t \in 1, 2, 3, ..., N$. It assumes that the production of O_t in each possible state $i (i = 1, 2, 3, ..., N)$ is stochastic and is characterized by a set of observation probability measures $B = b_i(O_t)_{i=1}^N$ where

$$b_i(O_i) = P(O_t|q_t = i) \tag{9}$$

If the state sequence q that led to the observation sequence $O = (O_1, O_2, ..., O_\gamma)$ is known, the probability of O being generated by the system is assumed to be

$$P(O|q, B) = b_{q_1}(O_1)b_{q_2}(O_2)...b_{q_\gamma}(O_\gamma) \tag{10}$$

The joint probability of O and q being produced by the system is simply the product of (8) and (10), written as

$$P(O, q|\pi, A, B) = \pi_{q_0}\Pi_{t=1}^T a_{q_{t-1}q_t} bq_t(O_t) \tag{11}$$

It then follows that the stochastic process, represented by the observation sequence 0, is characterized by

$$P(O|\pi, A, B) = \sum_q P(O, q|\pi, A, B) = \sum_q \pi_{q_0}\Pi_{t=1}^T a_{q_{t-1}q_t} b_{q_t}(O_t) \tag{12}$$

which describes the probability of O being produced by the system without assuming the knowledge of the state sequence in which it was generated. Thus, the triple $i = (n, A, B)$ in (12) defines an HMM[13].

9. HTK

Hidden Markov Model Toolkit commonly known as 'HTK' was first developed at the 'Speech Vision and Robotics Group' of the Cambridge University Engineering Department (CUED) in 1989 by Steve Young. HTK consisting a set of C library modules and tools that has been used for speech recognition research (using continuous density HMMs) over the last ten years.

9.1 HTK Software Architecture

HTK architecture defines its functionalities those are built into the library modules. These modules ensure that every tool interfaces to the outside world in exactly the same way. HTK also provides a central resource of commonly used functions. Figure 8 illustrates the software structure of a typical HTK tool and shows its input/output interfaces.

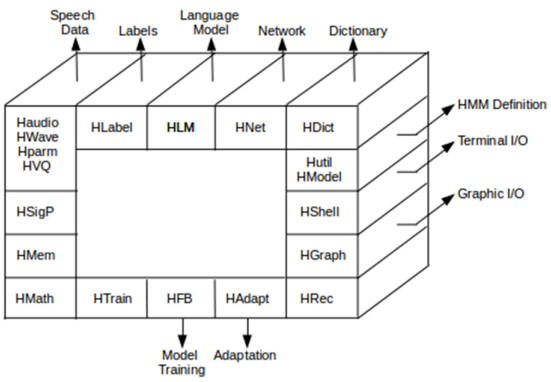

Fig. 8. HTK software architecture

User input/output and interaction with the operating system is controlled by the library module HShell and all memory management is controlled by HMem. Math support is provided by HMath and the signal processing operations needed for speech analysis are in HSigP. Each of the file types required by HTK has a dedicated interface module. HLabel provides the interface for label files, HLM for language model files, HNet for networks and lattices, HDict for dictionaries, HVQ for VQ codebooks and HModel for HMM definitions.All speech input and output at the waveform level is via HWave and at the parameterised level via HParm. As well as providing a consistent interface, HWave and HLabel support multiple file formats allowing data to be imported from other systems. Direct audio input is supported by HAudio and simple interactive graphics is provided by HGraf. HUtil provides a number of utility routines for manipulating HMMs while HTrain and HFB contain support for the various HTK training tools. HAdapt provides support for the various HTK adaptation tools. Finally, HRec contains the main recognition processing functions.

9.2 HTK Tool Kit

HTK[3] version 3.4.1 is used in the discussed digit recognizer. HTK tools HCopy is used in data preparation, HCompV, HERest snd HHEd in training, HVite in testing, and HResults in analysing the system. Figure 8 describes the HTK processes with the respective tools.

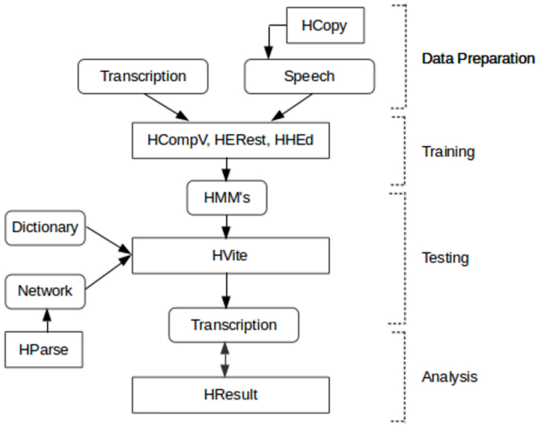

Fig. 9. HTK processes

9.2.1 Data Preparation Tool. **HCopy** is used to copy source files to an output file. Although HCopy can copy the whole file, but by setting appropriate configuration variables in a configuration file (i.e. .conf file), segments of files are extracted and converted to a parametric form as they are read-in.

9.2.2 Training Tool. For the training of the system, a prototype must be defined. For this, a HMM model prototype is created. The purpose of the prototype definition is to specify the overall characteristics and topology of HMM. Figure 10 shows the training process of the discussed system by HTK. The tool HCompV is used as no bootstrap data is available for our speech recognition system, so-called a flat start. In this case all of the phone models are initialised to be identical and have state means and variances equal to the global speech mean and variance.
The tool HERest is used to perform embedded training using

[3]The tool kit is available at `http://htk.eng.cam.ac.uk/`

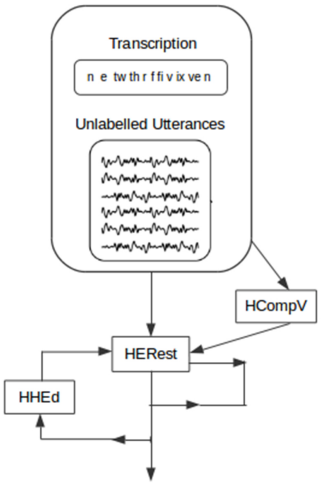

Fig. 10. HTK training process

the entire training set, after the creation of initial set of models. HERest performs a single Baum-Welch re-estimation of the whole set of HMM phone models simultaneously. For each training utterance, the corresponding phone models are concatenated and then the forward-backward algorithm is used to accumulate the statistics of state occupation, means, variances, etc. for each HMM in the sequence. The accumulated statistics are used to compute re-estimates of the HMM parameters when all training data has been processed. Hence, HERest is the core HTK training tool.
The tool HHEd is a HMM definition editor which will clone models into context-dependent sets, apply a variety of parameter tyings and increment the number of mixture components in specified distributions. It is used consistent to the philosophy of system construction in HTK that is HMMs should be refined incrementally. Thus, a typical progression is to start with a simple set of single Gaussian context-independent phone models and then iteratively refine them by expanding them to include context-dependency and use multiple mixture component Gaussian distributions.
The tool HVite is used to adapt HMMs to better model the characteristics of particular speakers using a small amount of training or adaptation data. The end result of which is a speaker adapted system with improved performance.

9.2.3 Recognition Tool. The HTK tool HVite uses the token passing algorithm to perform Viterbi-based speech recognition. HVite takes as input a network describing the allowable word sequences (network file), a dictionary defining how each word is pronounced (digit dictionary file) and a set of HMMs. It operates by converting the word network to a phone network and then attaching the appropriate HMM definition to each phone instance. Recognition can then be performed on either a list of stored speech files or on direct audio input.
The tool HParse is used to convert a higher level grammar notation (based on the Extended Backus Naur Form (EBNF) used in compiler specification) that can be applied as an alternative to a word network, into the equivalent word network.

9.2.4 Analysis Tool. HTK tool HResult is used to evaluate the performance of the recognizer, once it has been built. This is usually done by using it to transcribe some pre-recorded test sentences and match the recogniser output with the correct reference transcriptions. HResults uses dynamic programming to align the two transcriptions and produce results with word level recognition, Sentence level recognition and confusion matrix. Thus it calculates for the accuracy of the recognizing speech given by the speaker.

10. SYSTEM PERFORMANCE ANALYSIS

The analysis phase starts with the completion of processing test data by the recognizer. In this phase, HResult tool (of HTK) compares the HVite transcription output with original one and measure the performance for the system in various statistics. It matches each of the recognized and reference sequence by performing an optimal string match using dynamic programming. For calculation of optimal string match, reference score mechanism is used. Identical match is scored by 0, a label insertion by 7, a label deletion by 7 and a substitution by 10. The optimal string is calculated with lowest possible score, i.e. under worst case scenario. From optimal string score, substitution error (S), deletion error (D) and insertion error (I) can be calculated[4].
The percentage correct is then

$$\%C = \frac{N - D - S}{N} \times 100\% \tag{13}$$

and the percentage accuracy is

$$\%A = \frac{N - D - S - I}{N} \times 100\% \tag{14}$$

where, $\%C$ is the percentage correctness in word recognition, and $\%A$ is percentage accuracy in the recognition. HResult also produces a confusion matrix for the words those being confused with other words.
For performance analysis,the system is evaluated with different combination of data recorded in two ways- one where same speaker is involved with training and testing and the other where system is tested with different speaker from the training speaker. This paper evaluates a speaker dependent system by 120+20 training-testing samples combined for different speakers. The system performance is also measured with a female speaker to analyse the impact of female voice on the system. System is tested in a class room environment. All the data used is recorded in open room condition under the influence of natural noise. The average performance of the system lies between range of 85-95% word recognition for speaker dependent behaviour. But it falls to a level of 45% when comes to a speaker independent system.

10.1 Test 1

Here, the system is trained with only a particular speaker- it is trained with 40 samples (i.e. 400 words) and testing is done with 5 samples (i.e. 50 words) for the given trained speaker. Now, keeping the trained system same, the testing samples are increased by 5 for the respective speaker and the system is tested again. Former test gives 98% word recognition where for the latter gives 90% recognition of words. Table 2 gives the specification used for the system.

Table 2. Test-1 speaker specification

Speaker No.	Gender	Age	Training words	Testing words
1	male	22	40	5
2	male	22	40	10

The result obtained for the system is represented graphically in Figure 11.

[4]Available at `http://www.ee.columbia.edu/In/LabROSA/doc/HTKBook21`

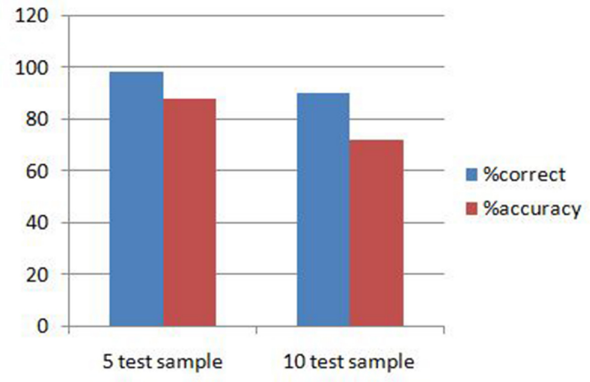

Fig. 11. Result for Test 1

10.2 Test 2

For the test, system is evaluated with a male and a female speaker. The training and testing is done using the same speaker. 80 samples per speaker is used for training and 20 samples per speakers is used for testing. The speaker and data specification used for Test 2 is given in Table 3.

Table 3. System specification for Test-2

Speaker No.	Gender	Age	Training words	Testing words
1	female	19	800	200
2	male	22	800	200

The system performance comes up with 83.6% and 95.4% for the female and male speaker respectively. The performance of the system in terms of word recognition. It shows 90% word recognition in average. Figure 12 represents the performance of the system in terms of word recognition and accuracy for each speaker.

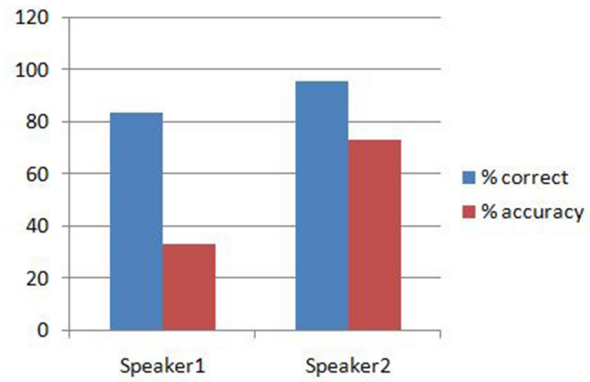

Fig. 12. Results for test 2

10.3 Test 3

The same system is again evaluated with 80 samples of data by a speaker for training in natural noise environment where testing is done with 20 samples of data by a different speaker. The system specifications for the training and testing is given in Table 4. System with Test 3 when evaluated for performance, the recognition value falls by 42-50%. It is due to the shortcoming of MFCC algorithm when implemented with speaker independent environment. The result is shown in Figure 13.

Table 4. System specification for Test 3

Speaker No.	Gender	Age	Training words	Testing words
1	male	22	800	
2	male	21		200

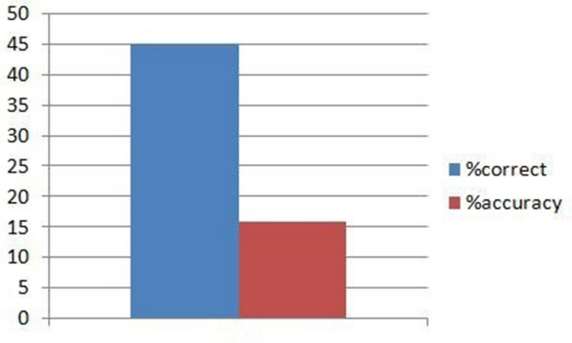

Fig. 13. Result for test 3

10.4 Test 4

For the test, it considers a mixed set of data in order to test the feasibility of the recognition system in case of mixed speaker database and tested within the system with 20 samples. This paper obtains a healthy recognition rate on a both male and female speaker based environment. The system specifications for Test 4 is given in the Table 5.

Table 5. System specification for test 4

Speaker No.	Gender	Age	Training words	Testing words
1	male	22	500	
2	female	19	500	
3	male	21	200	
1	male	22		100
2	male	19		100

The results obtained are shown in Figure 14.

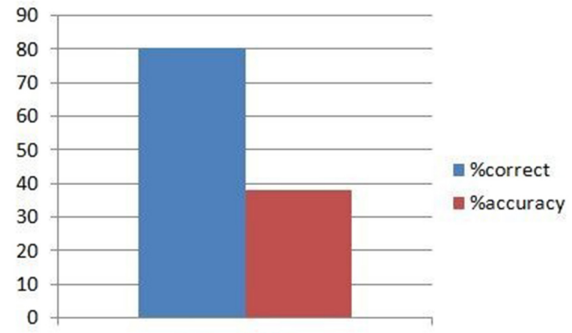

Fig. 14. Result for test 4

10.5 Comparison

The average result of the recognition obtained is compared to the average recognition result obtained by A.N. Mishra[1] and his team along with Mohit Dua[4] and his team. The comparison result is shown in Table 6.

Table 6. System specification for test 4

	Our work	Mohit Dua & Team	A.N. Mishra & Team
Recognition	80%	94.8%	99.2%
Environment	Natural Noise	Clean	Clean
Initialization	global parameter	initialization	initialization

All the systems are implemented using MFCC algorithm, HMM as acoustic model and HTK tool kit.

11. CONCLUSION AND FUTURE DIRECTIONS

To be concluded, an efficient and abstract system for the people of north-eastern part is need of the hour. The work implemented in this paper is a step towards the development of such systems. The work may be further extended to large vocabulary, continuous speech or in regional languages. As it is seen, the system is sensitive to spoken methods and changing scenario by making the accuracy a challenging area. Hence new algorithms and methods may be further added with the previous one for better accuracy too.

12. REFERENCES

[1] Mohit Dua et al., "Punjabi Automatic Speech Recognition Using HTK," in IJCSI International Journal of Computer Science Issues, IJCSI press, Mauritius, Vol. 1, Issue 4, No. 1, Jul. 2012.

[2] Ganesh S. Pawar, Sunil S. Morade, "Isolated English Language Digit Recognition Using Hidden Markov Model Toolkit," in International Journal of Advanced Research in Computer Science and Software Engineering, Jaunpur-222001, Uttar Pradesh, India, Vol. 4, Issue 6, June 2014.

[3] Vu Duc Lung et al., "Speech Recognition in Human-Computer Interactive Control" in Journal of Automation and Control Engineering, 2448 Desire Avenue, Rowland Heights, CA 91748, Vol. 1, No. 3, Sep. 2013.

[4] A.N. Mishra et al., "Isolated Hindi Digits Recognition: A Comparative Study" in International Journal of Electronics and Communication Engineering, India, Vol. 3, No. 1, 2010, pp. 229-238.

[5] P. Vijai Bhaskar et al.,"HTK Based Telugu Speech Recognition" in International Journal of Advanced Research in Computer Science and Software Engineering, Jaunpur-222001, Uttar Pradesh, India, Vol. 2, Issue 12, Dec. 2012.

[6] Yanli Zheng et al., "Accent Detection and Speech Recognition for Shanghai-Accented Mandarin" in Proc. Interspeech, 2005.

[7] Elitza Ivanova et al., "Recognizing American and Chinese Spoken English Using Supervised Learning"

[8] Konstantin Markov and Satoshi Nakamura, "Acoustic Modeling of Accented English Speech for Large-Vocabulary Speech Recognition" in International Speech Communication Association, ITRW on Speech Recognition and Intrinsic Variation(SRIV), Toulouse, France, May 2006.

[9] A. N. Mishra et al., "Robust Features for Connected Hindi Digits Recognition" in International Journal of Signal Processing, Image Processing and Pattern Recognition,Vol. 4, No. 2, June 2011.

[10] MarutiLimkara et al., "Isolated Digit Recognition Using MFCC AND DTW" in International Journal on Advanced Electrical and Electronics Engineering, Uttar Pradesh, India, Vol. 1, Issue 1, 2012.

[11] M.A.Anusuya and S.K.Katti, "Speech Recognition by Machine: A Review in International Journal of Computer Science and Information Security, vol. 6, no. 3, pp. 181-205, 2009.

[12] Preeti Saini et al., "Hindi Automatic Speech Recognition Using HTK" in International Journal of Engineering Trends and Technology, Vol. 4, Issue 6, June 2013.

[13] B. H. Juang and L. R. Rabiner "Hidden Markov Models for Speech Recognition" in Technometrics, American Statistical Association,732 North Washington Street, Alexandria, Vol. 33, No. 3, Aug. 1991, pp. 251-272.