Design and Implementation of Monophones and Triphones-Based Speech Recognition Systems for Voice Activated Telephony

Rupayan Das[1] and **Pradip K. Das**[2]

Submitted in April 2012; Accepted in January 2013

Abstract - Speech recognition is the ability of a machine or program to convert spoken words into its equivalent text form. Nowadays, most recognition systems use Hidden Markov Models for modeling the spoken utterances. In this paper we have implemented two speaker independent speech recognition systems which include all the words required for dialing a phone. The systems contain 42 words including digits from zero to nine and also include names of 20 persons. A total of 16,800 utterances have been used for training each system. The two systems are able to recognize continuous speech and it is implemented with the help of monophones and triphones using HTK. Experimental results show an accuracy of 74.11% for monophones based models and 93.77% for triphones based models.

Index Terms - HMM, HTK, Monophones, Triphones, Mel Frequency Cepstral Coefficient (MFCC).

1. INTRODUCTION

Pattern recognition is an important area of machine learning domain. The domain of pattern recognition is itself quite wide and encompasses several other interesting areas. The basic goal of a pattern recognition problem is to be enable a machine to identify as to which class, among a set of given classes, does a test pattern belongs. One interesting application of this area is presented in [1] for generation of traffic models in urban areas. [2] presents an interesting research on the problem of face recognition, which is now-a-days widely used as a measure to authenticate the users. [3] presents a very nice review of the statistical pattern recognition methods.

A subset of the pattern recognition domain is the area of speech recognition where spoken utterances are the patterns that are intended to be recognized. The process of speech recognition involves the communication between persons and machines where automata is generated to report the written equivalent of spoken words. From 1950's researchers were trying to make a device that can recognize human voice. In 1952, at Bell Laboratories a system for isolated digits recognition was built by Davis Biddulph and Balashek. The system heavily relied on the spectral resonances of the vowels of each digit. After that, lot of work on speech recognition has been done all over the world.

Some speech recognition systems give very good accuracy of more than 95% and are able to transcribe more than 150-160 words per minute. The improvement in the speech recognition system is increasing rapidly day by day. Nowadays, many hand-held devices like mobile phones, iPods, iPhones are trying to provide a good recognition system and research is still going on to improve the quality of the recognition accuracy. In the present era, mainly Hidden Markov Models (HMM) based speech recognition systems are used. HMM is a doubly embedded stochastic process with an underlying stochastic process that is not directly observable but can be observed only through another set of stochastic processes that produce the sequence of observations [4]. HMMs were first discussed in the second half of 1960 in a series of statistical papers by Leonard E. Baum and his colleagues [5]. In 1970 it has been first used as a tool for speech recognition by Baker [6] at CMU and by Jelinek and his colleagues at IBM [7]. Since then, due to its strong mathematical structure it gained its popularity day by day and started to be used in a wide range of applications, such as handwriting recognition [8], natural language domain and also for forecasting stock prices for interrelated markets [9], etc. HMM can also be used for speech recognition in other languages. In 2006 Gupta made an isolated word speech recognition for Hindi digits using continuous HMM [10]. Also in 2011 Kumar and Aggarwal made a Hindi recognition system using HTK which recognized 30 Hindi words [11]. In 2011 Hguyen presented a paper which describes a study of building a Vietnamese speech recognition system using HTK. The system gives the accuracy of 71.37% for speaker independent recognition before speaker adaptation and 75.96% after speaker adaptation [12]. HTK has also been used for speech recognition for other international languages such as Arabic language [13].

In this paper, a speaker independent recognition system is implemented with the help of Hidden Markov Model Toolkit which can be used to recognize continuous speech. The system includes all the commands required for dialing a phone. It consists of numbers from zero to nine and commands like "Call", "Dial", "Phone", "Flash", "Hangup", "Hash", "Star", "Redial" and "Hold". It also contains names of 20 persons. A total of 42 words have been used to make the system. The system is implemented using both monophones and triphones as base units. Experimental results show that the accuracy based on triphones models is much higher than the monophones based system.

2. HIDDEN MARKOV MODEL TOOLKIT (HTK)

HTK is a software toolkit for building and manipulating systems that use continuous density Hidden Markov models

*[1, 2] Department of Computer Science & Engineering,
Indian Institute of Technology (IIT), Guwahati, Assam, India
E-mail:[1] rupayan@iitg.ernet.in and [2]pkdas@iitg.ernet.in*

(HMMs) [14]. It is a collection of library modules written in C combining which a system can be designed. The first version of the HTK was developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED) in 1989 by Steve Young. The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be usedto build complex HMM systems [15].

2.1 HTK Implementation Structure

The different steps for building the HMMs using the toolkit are detailed below:

•Data Preparation: In this phase a database has been created by collecting data from 20 different speakers. Each speaker has 20 utterances of each word having a total of 16800 (20*42*20) utterances. The data is recorded using CSL workstation in a laboratory environment. A distance of approximately 5-10 cm is used between mouth of the speaker and microphone. Sounds are recorded at a sampling rate of 16000 Hz. After recording has been done, all the words are manually labeled and stored with a logical name.

•Feature extraction: As it is very complex to work with raw speech data, it is important to extract all relevant acoustic information in a compact form from raw speech. We use Mel frequency cepstral coefficients (MFCCs) [16] to extract feature vectors from the recorded raw data

•Model Training: In this phase, the first thing required is to define a prototype model which contains the information about the characteristics and the topology of the HMM. For our system, the topology used is 3-state left–right with no skips [17]. With the help of this proto file we generate the first HMM and then repeatedly re-estimate it to get the required optimal model.

3. IMPLEMENTATION DETAILS

First of all we make a grammar file which describes the words to be recognized. The grammar file for the telephone based system contains:

$digit = ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | ZERO;

$name = RAM | JOHN | AMIT | HEMANT | BIKASH | GOPAL | ARUN | SUMIT | JAMES | NITIN | MAYANK | DEBANJAN | ROHIT | ANIL | RAJA | STEVE | JHONSON | KRISHNA | NIL | PUNIT ;

$mode = ON | OFF;

(SENT-START (DIAL $digit $digit $digit $digit $digit $digit $digit $digit $digit $digit | (PHONE | CALL) $name | SPEAKER $mode | FLASH | HANGUP | HASH | STAR | REDIAL | HOLD) SENT-END)

From this grammar file, some sample commands that can be formed are listed in Table 1:

A total of 42 words have been selected to make the recognition system. The word CALL and PHONE can be used interchangeably. It is taken to give the user more flexibility while calling. After making the grammar it is saved in the *gram*file. The symbol $ denotes a string variable, the vertical bars denotes alternatives and the angle braces denotes one or more representations. After making the grammar file we need to make the word network for these words. This is done by executing the command **Hparse gram wdnet** which will take *gram* file as input and generate the word network file *wdnet* that contains each word-to-word transition.

Command	Command
DIAL 9985345631 (any 10 digit from zero to nine)	HASH
CALL RAM (any name chosen from $name in the grammar file)	STAR
PHONE RAM (any name chosen from $name in the grammar file)	REDIAL
FLASH	HOLD
HANGUP	

Table 1: Sample commands using the grammar file

The next step is to build the list of phonemes for each of the words in the vocabulary. This is done by using the command HDMan -m -w wlist -n monophones1 -l dlog dict names which will take as input names and wlist and generate the list of phonemes in the file monophones1.The wlist file contains the list of words and names file is same as wlist except that it also contains the phoneme sequences of the words. Table 6 contains all the words along with their corresponding phonemes. Now the silence sil is added to the list and saved in file monophone0. In addition SENT-END and SENT-START is augmented.

In order to train the system with the given words, the list of words to be spoken for training is generated. It is generated by using the command HSgen -l -n k wdnetdict > trainprompts which will use wdnet and dict files and generate the train prompts that contain a total of k training sentences. Next, the recording of all these sentences are to be done using the software HSLab provided by the toolkit.

Now for training the system it is required to replace the word in train.mlf file with its corresponding phonemes. This is done by executing the command HLEd -l '*' -d dict -i phones0.mlf mkphones0.led train.mlf which will take as input train.mlf file, dict file and mkphones0.led file and generate the corresponding phonemes in the file phones0.mlf. The train.mlf file contains the trainprompts sentences and mkphones0.led file contains commands used to replace the word with its corresponding phonemes.

The next step is to parameterize the raw speech waveforms into sequences of feature vectors. This is done by the command HCopy -T 1 -C cfg_mfc -S code_mfc.scp. The command will take code_mfc.scp and cfg_mfc file as input. The scp file contains the location of the .wav files and also the location of the .mfc files to be created. The configuration file cfg_mfc can be set as shown in Table 2 below:

Parameters	Value
TARGETKIND	MFCC_0_D_A
TARGETRATE	100000.0

Parameters	**Value**
SAVECOMPRESSED	T
SAVEWITHCRC	T
WINDOWSIZE	250000.0
USEHAMMING	T
PREEMCOEF	0.97
NUMCHANS	26
CEPLIFTER	22
NUMCEPS	12
ENORMALISE	F

Table 2: Contents of the *cfg_mfc* file

Now the monophones HMM is generated by using following steps:

For training the HMM, first of all a proto file is defined which defines the model topology. In our experiments the topology used is a 3-state left-right with no skips. The command HCompV -C config -f 0.01 -m -S tr_mfc_mono.scp -M hmm0 proto is executed to generate a new version of file proto and Vfloor in hmm0 directory. The config file contains the only line TARGETKIND = MFCC_0_D_A and tr_mfc_mono.scp file contain the locations of the MFC files. After that the model formed which is saved in the file proto is placed against each phoneme entry in hmmdefs file. Also copy the contents of vfloors to a file named macro.

Then it is required to re-estimate the flat start monophones models and this can be done by executing the command HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S tr_mfc_mono.scp -H hmm0\macros -H hmm0\hmmdefs -M hmm1 monophones0 which will generate hmmdefs and macros files in hmm1 directory. Executing the command two more times, the file hmmdefs and macros can be generated in hmm3 directory. The previous step generates a 3 state left-to-right HMM for each phone and also a HMM for the silence model sil. Now we need to create a 1 state short pause sp model by copying the contents of the sil model and placing it in the sp model. Since sp has its emitting state tied to the center state of the silence model, the centre step is retained and other states are deleted.

Now for making the model more robust, it is required to add an extra transition in the sil model which absorbs the various impulsive noises in the training data. This can be done by executing the command HHEd -H hmm4\macros -H hmm4\hmmdefs -M hmm5 sil.hed monophones1 where the input files are monophones1 and sil.hed. The sil.hed file contains data including:

```
AT 2 4 0.2 {sil.transP}
    AT 4 2 0.2 {sil.transP}
    AT 1 3 0.3 {sp.transP}
   TI silst
   {sil.state[3],sp.state[2]}
```

The AT command adds transitions to the given transition matrices and TI command creates a tied-state called slist. When we execute the command HHED, we get corresponding hmmdefs and macros files in the hmm5 directory. Finally, another two passes of HEREST are applied using the phone transcriptions with sp models between words. This results the models to be stored in hmm7 directory.

Since the dictionary contains multiple pronunciations of some words, so the phone models created so far can be used to realign the training data and create new transcriptions. This can be done by executing the command HVite -l '*' -o SWT -b silence -C config -a -H hmm7/macros -H hmm7/hmmdefs -i aligned.mlf -m -t 250.0 -y lab -I train.mlf -S train.scp dict monophones1 which uses the HMMs stored in hmm7 to transform the input word level transcription train.mlf to the new phone level transcription aligned.mlf using the pronunciations stored in the dictionary dict. When the aligned.mlf file is created, we execute another two passes of HERest which will store the required HMMs in hmm9 directory.

Now we are ready to run the recognizer for live input. For this, a configuration file config2 is needed which will convert the input data into its parameterization form. The config2 file contains the following parameters and their values:

Parameters	**Value**
SOURCERATE	625.0
SOURCEKIND	HAUDIO
SOURCEFORMAT	HTK
TARGETKIND	MFCC_0_D_A
TARGETRATE	100000.0
ENORMALISE	F
USESILDET	T
MEASURESIL	F
OUTSILWARN	T

Table 3: Contents of the config2 file

Now for recognizing the word, the command HVite -H hmm9/macros -H hmm9/hmmdefs -C config2 -w wdnet -p 0.0 -s 5.0 dict monophones1 is used which uses a token passing algorithm to perform viterbi-based speech recognition.The Viterbi algorithm finds the best state sequence for the observation sequence obtained from the previous steps. It takes wdnet, dict, monophones1and a set of HMMS as input. It converts the word network to a phone network and then attach the appropriate HMM definition to each phone instance. When we run the command, it first measures the speech and background silence level by prompting the user to speak an arbitrary sentence. After that it will repeatedly recognize the word and output into the terminal.

The triphones based HMMs are generated with the help of the following additional steps:

First the command HLEd -n triphones1 -l '*' -i wintri.ml fmktri.led aligned.mlf is executed which will convert the monophone transcriptions in aligned.mlf to an equivalent set of triphone transcriptions in wintri.mlf. Also a list of triphones is saved in triphones1 file. The mktri.led file is an edit script.

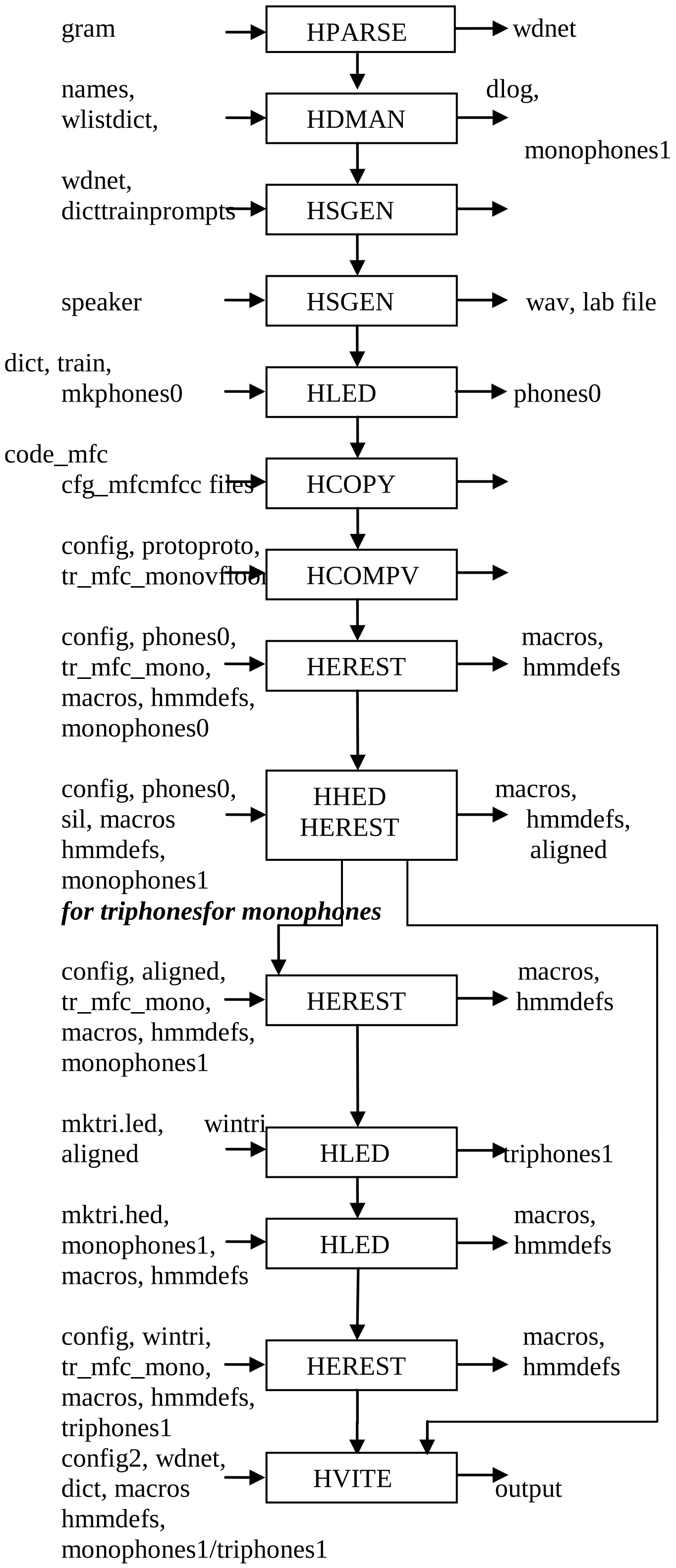


Figure 1: Block diagram representing the working of monophones and triphones-based recognition system.

which contains WB sp, WB sil and TC where WB commands define sp and sil as word boundary symbols. Now we have to make an edit script mktri.hed containing a clone command CL followed by TI commands to tie all the transition matrices in each triphone set. Now the cloning of models can be done by using the command HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1. Finally, another three passes of command HERest -B -C config -I wintri.mlf -t 250.0 150.0 1000.0 -s stats -S train.scp -H hmm11/macros -H hmm11/hmmdefs -M hmm12 triphones1 are applied to save the resultant models in hmm13 directory.
For live recognition, we can use the command:

```
HVite -H hmm13/macros -H hmm13/hmmdefs -C config2 -w
wdnet -p 0.0 -s 5.0 dict triphones1
```

The complete training and recognition process is explained with the help of a block diagram in Figure 3.

4. EXPERIMENTAL RESULTS

To find the accuracy of the system, 20 speakers have been selected to test the system. From these 20 speakers, 10 speakers are those whose voices have been already included while making the model and 10 new speakers are included to test the system. Each person speaks 20 utterances of each word resulting in a total of 400 (20*20) utterances per person. The system is tested for both monophones and triphones based models and the results are shown in Table 4 below:

Sl. No.	Words	Recognition accuracy (in percentage)	
		Monophones based HMMs	Triphones based HMMs
1	AMIT	77.25	94.25
2	ANIL	74.5	95.5
3	ARUN	71.5	92
4	BIKASH	76.25	93
5	CALL	80.25	97
6	DEBANJAN	77.75	95.5
7	DIAL	81.75	96.25
8	EIGHT	67.75	91.75
9	FIVE	73.75	93.5
10	FLASH	64.5	92.25
11	FOUR	78	93
12	GOPAL	76	90.5
13	HANGUP	80.5	94.75
14	HASH	63	92.75
15	HEMANT	73	95.25
16	HOLD	76.75	92.5
17	JAMES	71.5	96
18	JHONSON	75.25	93.25
19	JOHN	77.25	92.5

Sl. No.	Words	Recognition accuracy (in percentage)	
		Monophones based HMMs	Triphones based HMMs
20	KRISHNA	69.25	91
21	MAYANK	72.25	95.5
22	NIL	69	92
23	NINE	80.5	94
24	NITIN	69.5	95.75
25	OFF	74.5	93
26	ON	75.5	94.25
27	ONE	81.5	96.75
28	PHONE	83	97.75
29	PUNIT	63.75	90.5
30	RAJA	76.25	95.75
31	RAM	69.25	94
32	REDIAL	77.5	93
33	ROHIT	73.25	91
34	SEVEN	78.25	97.25
35	SIX	83.5	98
36	SPEAKER	75.25	92
37	STAR	71	93.25
38	STEVE	72	94.25
39	SUMIT	65.75	90.75
40	THREE	76.75	94.25
41	TWO	62.75	90.25
42	ZERO	78	93.25

Table 4: Word recognition performance using Monophones and Triphones HMMs.

Table 5 below shows the confusion matrix of the mostly mis-recognized words and also for the word "six" which gives good recognition result.

Words	Words misrecognized as (percentage of misrecognition)				
TWO	ONE (10.5)	THREE (4.5)	NINE (3.25)	ZERO (7)	FOUR (2.25)
GOPAL	ROHIT (3.25)	ANIL (4.5)	JAMES (1.75)	JOHN (2.25)	BIKASH (2.75)
PUNIT	NIL (3.25)	GOPAL (4.75)	HEMANT (2.75)	ROHIT (6.75)	AMIT (9.25)
SUMIT	AMIT (9.75)	PUNIT (4.5)	JOHN (2)	HEMANT (1)	NITIN (7.75)
KRISHNA	GOPAL (3.25)	NITIN (5)	JHONSON (4.25)	JOHN (4)	BIKASH (5.25)
ROHIT	JAMES (7.5)	PUNIT (2)	JAMES (1.75)	HEMANT (5.25)	AMIT (1.25)
EIGHT	NINE (8.25)	TWO (8.5)	FIVE (6.5)	THREE (2.75)	-
FLASH	HASH (15.25)	HOLD (8.5)	STAR (4)	-	-
HASH	FLASH (16.5)	HOLD (8.5)	HANGUP (4.75)	-	-
SIX	ONE (5.5)	SEVEN (9)	-	-	-

Table 5: Words that are confused with other words.

5. ANALYSIS AND DISCUSSIONS

From the above Table 4 it has been observed that for monophones based system, the word **hash** and **flash** gives low recognition score. This may be due to the fact that the pronunciation of both the words is very similar to each other. Also the words **two, Rohit, Punit** and **Sumit** gives poor results. From this we can infer that the words containing the dental sound /t/ can confuse the system. Sometimes the pruning of the words during labeling is not done properly and some part close to the boundary of the words gets removed. As a result of this, the models formed by that data is not properly trained and it gives low recognition score. So while cutting the word for labeling it is necessary to leave some silence region before and after each word in the data preparation stage.

The two Figures 2 and 3 shown below depicts the spectrogram of worst recognized word 'TWO' and the best recognized word 'SIX' found in the course of the experiments. It is clear from the plots that though the formants are well marked and steady for "two", there is less variations that can be captured and modeled by our system. On the other hand the spectrogram plot for "six" clearly shows numerous acoustic changes in the formant structure during the utterance. We can infer that this property of modeling the transitions is well captured by the triphones based HMMs.Though some earlier experiments show poor result for the word "six", in our experiment "six" gives very good result. This may be due to the good recording quality or may be the spotting and pruning of word is proper for "six".

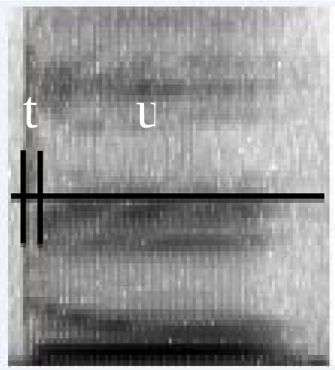


Figure 2. Spectrogram of "Two".

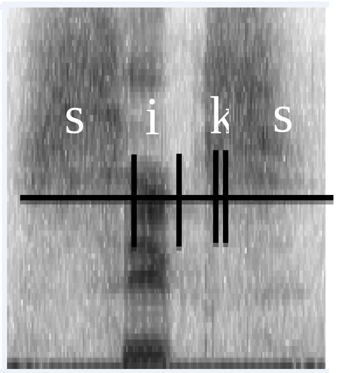


Figure 3. Spectrogram of "Six".

For triphones based recognition system it is observed that all the words give reasonably good result as compared to monophones based system. The monophones based recognition system gives accuracy of 74.11% while triphones based recognition system gives 93.77%. So, from the experiment we can clearly say that the triphones model gives much better results than the monophones-based model.

6. CONCLUSIONS AND FUTURE WORK

In this paper two telephone-based recognition systems are developed with the help of monophones and tri-phones using the Hidden Markov Toolkit (HTK). The system is able to recognize the words spoken by speakers inside and outside the set for both continuous and isolated words. The whole experiment has been carried out in a normal room environment. The system gives good accuracy of 74.11% for monophones and 93.77% for triphones based models. The triphones based models perform far better than the monophones based models. Work is now underway to semi-automate the generation of the training models to deploy a speech recognition system at a short notice. The authors are also planning to update the computation of the feature vectors in *hcopy* to include new acoustic-phonetic features within the toolkit.

REFERENCES

[1] Shivendra Goel, J. B. Singh, Ashok Kumar Sinha, "Traffic Generation Model For Delhi Urban Area Using Artificial Neural Network", BIJIT - BVICAM's International Journal of Information Technology, Vol. 2, No. 2, 2010.

[2] R. K. Agrawal, Ashish Chaudhary, "Modified Incremental Linear Discriminant Analysis for Face Recognition", BIJIT - BVICAM's International Journal of Information Technology, Vol. 1, No. 1, 2009.

[3] Anil K. Jain, Robert P. W. Duin, Jianchang Mao, "Statistical Pattern Recognition: A Review",IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1,2000, pp. 4-37.

[4] Lawrence Rabiner, Biing-Hwang Juang, "Fundamentals of Speech Recognition",PTR Prentice Hall, 1993, ISBN 0130151572, 9780130151575.

[5] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes", Inequalities, Vol. 3, 1972, pp. 1-8.

[6] J**.** K. Baker, "The Dragon System - An Overview", IEEE Transactions on Acoustic Speech and Signal Processing, Vol. ASSP-23**,** No. 1, Feb. 1975, pp. 24-29.

[7] F. Jelinek, "Continuous Speech Recognition by Statistical Methods", Proceedings of IEEE,Vol. 64**,** April 1976, pp. 532-536.

[8] Vinciarelli A and Luettin J., "Off-line Cursive Script Recognition based on Continuous Density HMMs", Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000, pp. 493-498.

[9] Md. Rafiul Hassan and Baikunth Nath, "Stock Market Forecasting using Hidden Markov Model: A New Approach", IEEE Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA), 2005, pp. 192-196.

[10] Gupta, R., "Speech Recognition for Hindi", M. Tech. Project Report, Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Mumbai, India, 2006.

[11] Kuldeep Kumar, R. K. Aggarwal, "Hindi Speech Recognition System using HTK", International Journal of Computing and Business Research, ISSN (Online) : 2229-6166, Volume 2, Issue 2, May 2011. http://www.researchmanuscripts.com/PapersVol2N2/IJCBRVOL2N2P3.pdf

[12] Nguyen Hong Quang, Trinh Van Loan, Le The Dat, "Automatic Speech Recognition for Vietnamese using HTK System", IEEE International Conference on Computing and Communication Technologies Research, Innovation and Vision for the Future (RIVF), Hanoi, Nov. 2010, pp. 1-4.

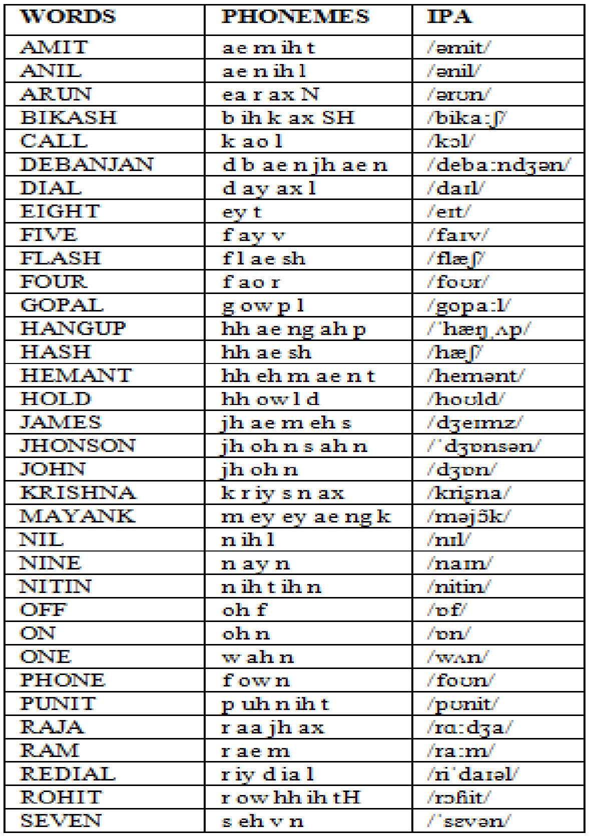
[13] Bassam A. Q., Al. Qatab, Raja N. Ainon, "Arabic Speech Recognition using Hidden Markov Model Toolkit (HTK)", Information Technology International Symposium (ITSim), Kuala Lumpur, 2010, pp. 557-562. http://noble.gs.washington.edu/papers/htk-sequence.pdf

[14] P C Woodland, J J Odell, V Valtchev, S J Young, "Large Vocabulary Continuous Speech Recognition Using HTK", Proceedings of ICASSP 94 IEEE International Conference on Acoustics Speech and Signal Processing, Adelaide, 1994, pp. II/125-II/128

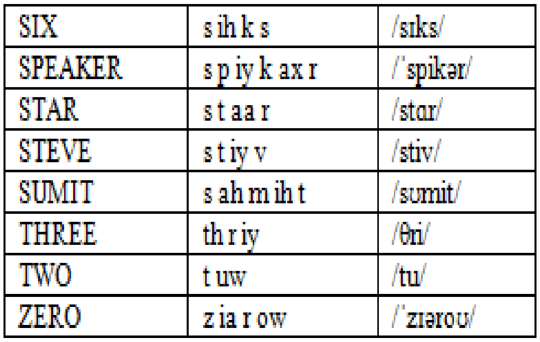
[15] HTK Official website on History of HTK page [Online] Available: http://www.htk.eng.cam.ac.uk

[16] A. N. Mishra, Mahesh Chandra, Astik Biswas, S. N. Sharan, "Robust Features for Connected Hindi Digits Recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 4, No. 2, June 2011, pp. 79-90.

[17] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. C. Woodland. The HTK Book Version 3.4.1, 2009.



WORDS	PHONEMES	IPA
AMIT	ae m ih t	/əmit/
ANIL	ae n ih l	/ənil/
ARUN	ea r ax N	/ərʊn/
BIKASH	b ih k ax SH	/bika:ʃ/
CALL	k ao l	/kɔl/
DEBANJAN	d b ae n jh ae n	/deba:ndʒən/
DIAL	d ay ax l	/daɪl/
EIGHT	ey t	/eɪt/
FIVE	f ay v	/faɪv/
FLASH	f l ae sh	/flæʃ/
FOUR	f ao r	/foʊr/
GOPAL	g ow p l	/gopa:l/
HANGUP	hh ae ng ah p	/ˈhæŋ ˌʌp/
HASH	hh ae sh	/hæʃ/
HEMANT	hh eh m ae n t	/hemənt/
HOLD	hh ow l d	/hoʊld/
JAMES	jh ae m eh s	/dʒeɪmz/
JHONSON	jh oh n s ah n	/ˈdʒɒnsən/
JOHN	jh oh n	/dʒɒn/
KRISHNA	k r iy s n ax	/krişna/
MAYANK	m ey ey ae ng k	/məjɔ̃k/
NIL	n ih l	/nɪl/
NINE	n ay n	/naɪn/
NITIN	n ih t ih n	/nitin/
OFF	oh f	/ɒf/
ON	oh n	/ɒn/
ONE	w ah n	/wʌn/
PHONE	f ow n	/foʊn/
PUNIT	p uh n ih t	/pʊnit/
RAJA	r aa jh ax	/rɑ:dʒa/
RAM	r ae m	/ra:m/
REDIAL	r iy d ia l	/riˈdaɪəl/
ROHIT	r ow hh ih tH	/rɔɦit/
SEVEN	s eh v n	/ˈsɛvən/



SIX	s ih k s	/sɪks/
SPEAKER	s p iy k ax r	/ˈspikər/
STAR	s t aa r	/stɑr/
STEVE	s t iy v	/stiv/
SUMIT	s ah m ih t	/sumit/
THREE	th r iy	/θri/
TWO	t uw	/tu/
ZERO	z ia r ow	/ˈzɪəroʊ/

Table 6: The phonetic breakup and IPA representation of all the words.