Speech Recognition Using Monophone and Triphone Based Continuous Density Hidden Markov Models

Sharada C. Sajjan[1], Vijaya C[2]

[1,2]*Department of Electronics and Communication Engineering,*
SDM College of Engineering and Technology,
Dharwad, Karnataka, India

***Abstract*—Speech Recognition is a process of transcribing speech to text. Phoneme based modeling is used where in each phoneme is represented by Continuous Density Hidden Markov Model. Mel Frequency Cepstral Coefficients (MFCC) are extracted from speech signal, delta and double-delta features representing the temporal rate of change of features are added which considerably improves the recognition accuracy. Each phoneme is represented by tristate Hidden Markov Model(HMM) with each state being represented by Continuous Density Gaussian model. As single mixture gaussian model do not represent the distribution of feature vectors in a better way, mixture splitting is performed successively in stages to eight mixture gaussian components. The multi-gaussian monophone models so generated do not capture all the variations of a phone with respect to its context, context dependent triphone models are build and the states are tied using decision tree based clustering. It is observed that recognition accuracy increases as the number of mixture components is increased and it works well for tied-state triphone based HMMs for large vocabulary. TIMIT Acoustic-Phonetic Continuous Speech Corpus is used for implementation. Recognition accuracy is also tested for our recorded speech.**

Keywords—MFCC, Hidden Markov Model(HMM), Hidden Markov Model Tool Kit(HTK), Monophones, Tied-State Triphones.

I. INTRODUCTION

Automatic Speech Recognition (ASR) has gradually evolved from limited vocabulary isolated word systems to very large vocabulary, speaker independent continuous systems. One of the possible approaches to speech recognition is to create statistical models for each word in the vocabulary and use pattern recognition approach to recognize speech. This is a reasonable approach for vocabularies up to several hundred words. When the vocabulary size is of the order of 1000 words or greater, the computational complexity and memory size increases which is not acceptable. Hence, most large vocabulary speech recognition systems have adopted the strategy of using a basic unit of speech called phoneme that is shorter than a word and for which the number of such units required to represent all spoken sounds is relatively small[1]. For this work, phoneme is chosen as basic unit of representing words. All the speech recognizers include an initial front end that converts speech signal into its compressed form called feature vectors. The most commonly used methods are Linear Predictive Coding(LPC) and MFCC. LPC analysis provides better representation of speech sounds as it closely matches the resonant structure of human vocal tract that produces the corresponding sound. This simplified all pole model is a natural representation of voiced sounds, but for nasals and fricative sounds, the acoustic theory requires both poles and zeros in the vocal tract transfer function. If the prediction order is high enough, all pole model provides a good representation of almost all the sounds of speech [2]. The disadvantage of LPC is that it operates on linear scale where as human perception is logarithmic in nature. MFCC analysis is based on 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 [3,4].

The next step is pattern processing which determines the category of each pattern by comparing with stored patterns. Speech being non stationary, stochastic approach like HMM provides most robust way of quantifying speech patterns. Speech is described by parameterized statistical models and recognition is performed by considering the most likely model which produces the given sequence of observations. This paper aims to develop continuous speech recognition using HMM by using Hidden Markov Model Tool Kit (HTK).

Hidden Markov Model Tool Kit (HTK) is a software toolkit for building speech recognition applications developed by Cambridge University Engineering Department (CUED). It consists of set of library modules written in ANSI C language and runs mainly on UNIX operating system[6]. The tool provides sophisticated facilities for speech analysis, HMM training, testing and result analysis. The software supports HMMs using both continuous density mixture gaussians and discrete distributions and can be used to build complex HMM systems. There are two major processing stages involved. Firstly HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. To determine the parameters of HMM, Baum-Welch algorithm is used. Secondly, unknown utterances are transcribed using HTK recognition tools by using Viterbi Decoding[6]. HTK 3.4.1 version[7] is used for simulation.

II. Feature Extraction

Speech being quasi-stationary, feature extraction is carried over short segments which gives compact representation of speech. The quality of good feature is that it gives maximum information about the class within much smaller dimension. These features are important in deciding the overall recognition performance. Among LPC and MFCC, MFCC analysis provides better representation of speech sounds [3].

The human ear resolves frequencies non-linearly across the audio spectrum and therefore designing a front-end to operate in a similar non-linear manner improves recognition performance [6]. 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 on Mel-frequency scale, defined by

$$\text{Mel(f)} = 2595\log_{10}\left(1+\frac{\text{f}}{700}\right) \quad (1)$$

To implement this filter bank, magnitude of the fourier transform of windowed speech is taken. Each FFT magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated. Thus each filter holds a weighted sum representing the spectral magnitude in that filter bank channel. MFCC are then computed from log filter bank amplitudes using Discrete Cosine Transform given by

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

where m_j is the output of j^{th} filter, N is the number of filterbank channels. Fig.1 shows the block diagram of feature extraction using MFCC.

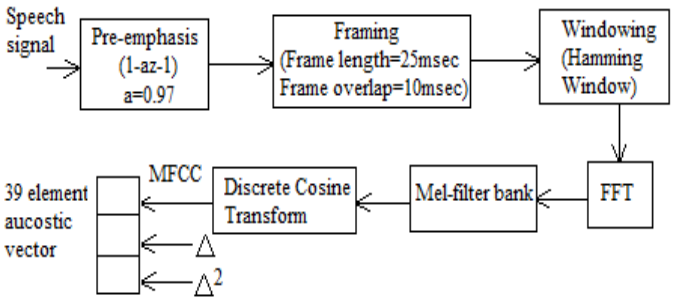

Fig. 1. MFCC feature extraction method

Delta and Acceleration coefficients have been added which reflects the dynamic changes within frames of speech. The output of feature extraction is a series of vectors consisting of MFCC coefficients.

III. Pattern Recognition

Pattern recognition classifies input pattern by comparing with stored patterns. HMM is the most suitable pattern classification method for nonstationary signals. The basic problem of speech recognition is to find the word sequence $W=W_1W_2\ldots W_m$ given the acoustic observation $O=O_1O_2\ldots O_n$. This can be evaluated using Bayes theorem given by

$$W = arg\max_{w}(P(W|O) = arg\max_{w}\left(\frac{P(W)P(O|W)}{P(O)}\right) \quad (3)$$

where P(W) is the priori probability of the word W that can be estimated from language model, P(O|W) is the observation likelihood called acoustic model and P(O) is the probability of the given acoustic observation. Since the maximization of (3) is done with variable O fixed, to find W it is enough to maximize the numerator alone. Fig.2 shows the architecture for continuous speech recognition.

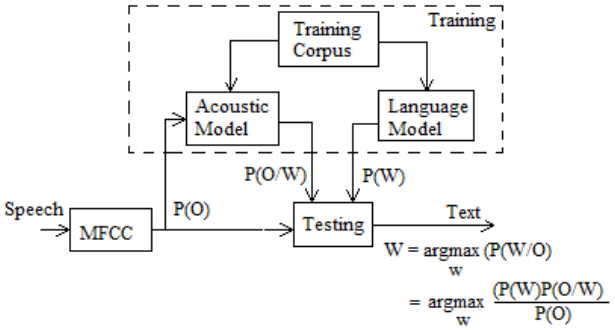

Fig. 2. System architecture for continuous speech recognition.

Language models are used to model regularities in natural language. The most popular methods are statistical n-gram models, which attempts to capture the syntactic and sematic constraints by estimating the probability of a word in a sentence given the preceding (n-1) words as given in (4). This probability strictly depends on the amount of the text corpus.

$$P(W) = P(w_1)P(w_2/w_1)\ldots\ldots P(w_N/\,w_1w_2\,\ldots w_{N-1}) \quad (4)$$

The bigram language model is used which evaluates the probability of occurrence of a word given the previous word.

Acoustic model is represented using P(O|W) which estimates the probability of acoustic observations given the word. HMM is used for acoustic modeling because HMM can normalize speech signals time-variation and characterize speech signal statistically thus helping to parameterize the conditional probabilities [8]. Thus given the acoustic model and language model, unknown word sequence W is obtained by maximizing the probability P(W)P(O|W).

An HMM consists of N number of states, where each state represents the position of human vocal tract apparatus when speaking. Each state j has an observation probability distribution $b_j(o_t)$ which determines the probability of generating observation o_t at time t and each pair of states i and j has an associated transition probability a_{ij}. Each state is represented by continuous density model in which each observation probability distribution is represented by mixture gaussian density [6]. For each state j, the probability $b_j(o_t)$ of generating observation o_t is given by

$$b_j(o_t) = \prod_{s=1}^{S}\left[\sum_{m=1}^{M_{js}} c_{jsm}\,N(o_{st};\mu_{jsm},\sum_{jsm})\right]^{\gamma_s} \quad (5)$$

Where M_{js} is the number of mixture components in state j for stream s, c_{jsm} is the weight of m'th component and $N(.:\mu, \sum)$ is a multivariate gaussian with mean μ and covariance matrix $\sum$, that is given by

$$N(o,\mu,\sum) = \frac{1}{\sqrt{(2\pi)^n|\sum|}} e^{-\frac{1}{2}(o-\mu)^T \Sigma^{-1}(O-\mu)} \quad (6)$$

Where n is the dimensionality of o. The exponent γ_s is a stream weight and its default value is one [6].

One HMM is build for each phoneme and continuous speech is recognized by joining the phonemes together to make any required vocabulary using a pronunciation dictionary.

Single gaussian HMM do not accurately characterize the speech data, hence multiple mixture gaussian components have been used which represent the distributions of feature vectors in a better way. As the occurrence of a phoneme depends on the previous and the next phoneme, context dependent triphone HMMs are build and states are tied using decision tree based clustering to have robust speech recognition.

IV. Implementation

ASR system is implemented using HTK 3.4.1 version. Speech data is needed both for training and testing. In the case of training data, the transcriptions (textual information) of the training utterances are used in conjunction with pronunciation dictionary to provide the initial phone level transcriptions. Training data with good phonetic balance and coverage is needed. Here TIMIT acoustic-phonetic database is used for implementation. ASR using HTK involves four basic steps as shown in fig.3.

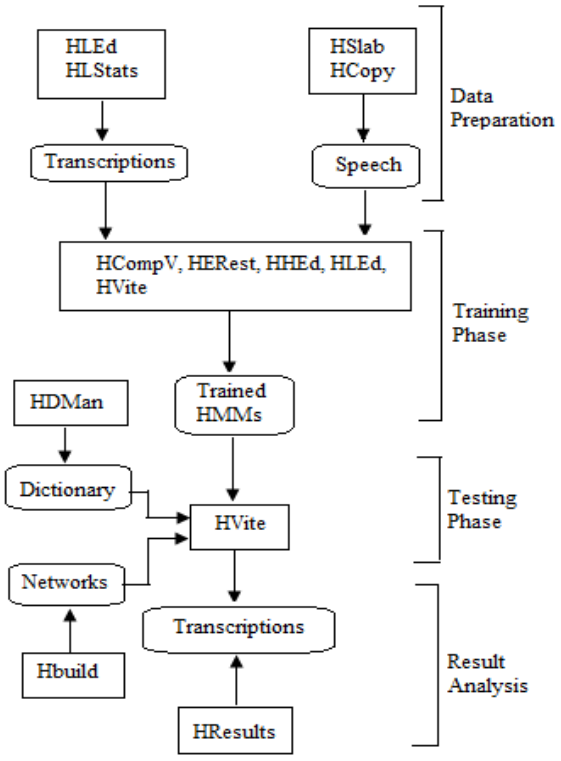

Fig. 3. HTK processing stages for ASR

A. Data Preparation

It is a collection of grammar, corresponding pronunciation dictionary, recording wave files, and coding the data. In grammar preparation, a network that describes the sequence of words to be recognized and a dictionary which describes the sequence of phonemes that constitute a word are created. Transcriptions of the training data are transformed into formats that are required for training. i.e., Wordlist, master label files, phoneme dictionary, phoneme list are created. The tool **HSlab** is used for recording. By making use of the phonetic dictionary, the tool **HLEd** creates phonetic transcriptions files, the tool **HLStats** generates bigram language model which displays statistics of words and word sequences. Using this bigram language model and word list, the tool **HBuild** constructs word level network containing list of nodes representing words and a list of arcs representing the transition between the words. Coding the data i.e., feature extraction is done using **HCopy** tool by setting parameters in the configuration file.

B. Training Phase

In training, an acoustic model is build for each phoneme. This acoustic modeling using HMM captures the distinctive properties of the speech which takes into account speaker variations, pronunciation variations, context dependent phonetic coarticulation variations. For this reason, acoustic training corpus has to be quite large to obtain robust acoustic model.

Creating Single Gaussian Monophone HMMs

Firstly initial set of single gaussian monophone HMMs are created by using prototype model and the set of observation vectors. Each HMM consists of five states with first and last state being the silence model. Each state is represented by a single gaussian having mean, variance and mixture weights. First a set of identical monophone HMMs in which every mean and variance are identical are created by using **HCompV** tool. This tool will scan a set of data files, compute the global mean and variance and sets all gaussians in a given HMM to have same mean and variance[6]. Once the initial set of models have been created, the tool **HERest** performs embedded re-training to build new HMMs where-in model values are updated to maximize the probability of observation sequence. This retraining is performed for several iterations. Fig.4 shows the monophone expansion of the word "cat" which consists of three phonemes with each phoneme represented by tri-state HMM with each state being represented by single gaussian having mean, variance, mixture weights which are calculated using observation vectors of training data.

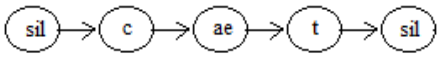

Fig. 4. Monophone Expansion of the word bit

Creating Multiple Mixture Gaussian Monophone HMMs

Single mixture gaussian component do not fit the distributions of feature vectors in a better way. Hence mixture splitting is performed successively in stages to multiple mixture components and re-training is performed by tool **HHEd**. This

tool chooses component with the largest mixture weights and making two mixtures where there was one before by perturbing the mean values of the split mixture components. This is done by state-by-state basis until all states are mixture split [6]. Phones are found to vary depending on the preceding and succeeding phones and this aspect needs to be captured within the acoustic models to improve performance. This can be achieved by building triphone models.

Creating Triphones from Monophones

Here context-dependent triphone HMMs are build by converting monophone transcriptions to triphone transcriptions referred to as word internal and a set of triphone models are created by copying monophones and re-estimating. The tool **HLEd** is used to generate triphone transcriptions and the tool **HHEd** is used to generate triphone models. Re-estimation of these models is then performed using **HERest.** Fig.5 shows the triphone expansion of the word "cat". The first and last are biphones since they are preceded and followed by silence respectively.

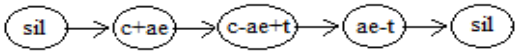

Fig. 5. *Triphone Expansion of the word bit*

Creating Tied-State Triphones

Tying means, one or more HMMs share the same set of parameters. In this model building process, similar acoustic states of triphones are tied in order to share data to ensure that all state distributions can be robustly estimated [6,9]. The two mechanisms which allow states to be clustered are data-driven and decision trees. Decision-tree based clustering is used, the advantage being it allows previously unseen triphones to be synthesized. This is performed using **HHEd** tool.

C. Testing Phase

The test utterances are coded using **HCopy** tool by setting the same configuration parameters used in training. The tool **HVite** takes as input, a network describing the allowable word sequences, a dictionary defining how each word is pronounced, a set of HMMs and coded test data. It operates by converting the word network to a phone network, and then attaching an appropriate HMM definition to each phone instance[6].

D. Result Analysis

The tool **HResults** gives sentence and word level statistics indicating the recognition accuracy.

$$\text{Percent Correct} = \frac{\text{N} - \text{D} - \text{S}}{\text{N}} \times 100\%$$

$$\text{Percent Accuracy} = \frac{\text{N} - \text{D} - \text{S} - \text{I}}{\text{N}} \times 100\%$$

$$\text{Word Error rate} = 100\% - \text{Percent Accuracy}$$

Where N is the total number of test utterances, D is the number of deletions, I is the number of insertions, S is the number of substitutions. Percent Accuracy is lower than percent correct since it takes into account of insertion errors while latter ignores.

V. Results and Discussion

A. Database

Acoustic Phonetic Continuous Speech TIMIT database has been used for the implementation. It consists of utterances of 630 speakers. There are a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The speech material has been subdivided into portions for training and testing. The test data has a core portion containing 24 speakers, 2 male and 1 female from each dialect region. There are total of 4620 training utterances and 1680 test utterances which are sampled at the rate of 16 kHz.

B. Coding

The speech signal is pre-emphasized using first order digital filter with pre-emphasis coefficient 0.97, blocked into frames of 25 msec duration at 10 msec frame rate, passing through hamming window to minimize discontinues at the edges, then represented by MFCC vector of length 39 consisting of 13 MFCC coefficients + 13 delta + 13 delta-delta coefficients. These MFCC values are evaluated by setting parameters in the configuration file. The output of feature extraction is series of frames of speech where in each frame is represented by MFCC vector.

C. Grammar preparation

Using training transcriptions, bigram network is created which defines all the words and word sequences. i.e., it puts all words in the vocabulary in a loop and therefore allows any word to follow any other word. The tool **HLStats** generates bigram language model(bg1) and the tool **HBuild** generates a word network(bgwdnet) with 6144 nodes representing the words and 37736448 arcs representing the transition between the words.

D. Monophone system

There are total of 78 phonemes used in TIMIT database, out of which 46 phonemes are of English language (American), 1 phoneme for silence(sil), 1phoneme for short pause(sp) and the remaining 30 are stressed phonemes. Continuous Density HMMs are build for each phoneme which uses 5 states with first and last state being non emitting states. Each state is represented by a gaussian model having observation means and observation covariances. Initial HMMs are created by setting all gaussians to have same mean and variance equal to global mean and variance of observations. These model values are then updated using embedded re-estimation formulae. The number of iterations used is four. Then mixture splitting is performed successively in stages from 1 to 2, 2 to 4 then from 4 to 8 mixture components. After each splitting HMMs are re-estimated twice. Thus for each of 78 phonemes, 78 HMMs are build.

Testing is performed using first three hundred test utterances of TIMIT test database using Viterbi decoding. The

tool **HResults** gives sentence and word correctness in percent as shown in fig.6 and fig.7. It is observed that as the number of mixture components is increased the percent correctness increases as multiple mixture gaussians fit the distributions of feature vectors in better way.

E. Tied-state Triphone system

Context dependent triphone models are created by copying the monophones and re-estimating. Similar acoustics of these states are tied together using decision tree clustering for robust recognition. For TIMIT training database consisting of 4620 training utterances having 6144 words, 9267 word internal triphones were created which are called logical HMMs. This huge amount of triphones increases the search process because the decoder determines which models should be used during search. Also unseen triphones should be included in the list of triphones occurred in acoustic training. To reduce the number of triphones and to accomplish the unseen triphones, similar states are tied using decision tree based clustering which generates upto 1815 physical HMMs. Recognition experiment performed with first 300 test utterances from TIMIT test database for different gaussian mixtures is shown in fig.6 and fig.7. It is observed that recognition accuracy is good for triphone system compared with monophone system. It is also observed that as the number of gaussian components is increased the recognition accuracy increases.

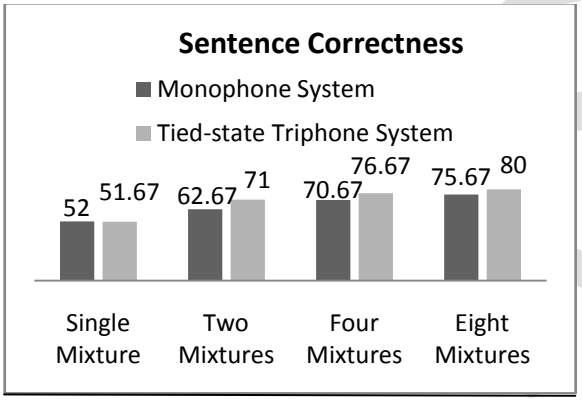

Fig. 6. Comparison of Sentence Correctness for Monophone System and Tied-state Triphone System for multiple mixture CDHMMs

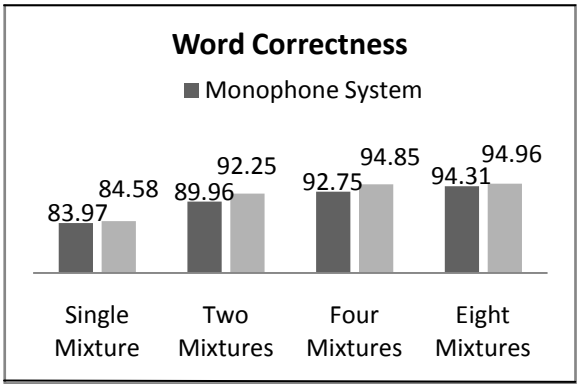

Fig. 7. Comparison of Word Correctness for Monophone System and Tied-state Triphone System for multiple mixture CDHMMs

F. Recognition accuracy for recorded database

ASR is also tested for our recorded database. The recordings are taken at a sampling rate of 16 kHz with 16 bits per sample value. Training database consists of 12 sentences of read speech spoken by different speakers. With 42 phonemes for our Indian English, separate phoneme dictionary is created. Test database consists of 50 sentences which are not used in training but having the same words used in training database, taken from speaker-independent speakers.

Monophone and Tied-state triphone HMMs are generated using training database and tested using test database. Table 1 shows the sentence and word correctness for monophone system and eight gaussian tied state triphone system. It is observed that accuracy for triphone system is better compared with monophone system as triphone HMMs capture the context variations.

Table 1. Recognition accuracy for recorded database

Recognition Accuracy	Eight Gaussian monophone system	Eight Gaussian tied-state triphone system
Sentence Correctness	80.00	84.6
Word Correctness	96.61	97.9

VI. Conclusion

Speech being concatenation of phonemes, each phoneme is represented by tri-state HMM, with each state being represented by single gaussian monophone model. As this model do not represent the distribution of feature vectors in a better way, mixture splitting is performed successively in stages up to eight mixture gaussian components. As these multi-gaussian monophone models donot capture all the variations of phone with respect to its context, triphone models are generated by cloning monophone systems which generates very large number of word internal triphones. To minimize the search process and to overcome the unseen triphones in test database, states are tied using decision tree based clustering which generates less number of physical HMMs. These are used as training database. Testing is performed using Viterbi search process and results are obtained which shows that tied-state triphone system gives better performance than monophone system and accuracy increases as the number of mixture components is increased. Future work is to implement recognition system for our regional language, Kannada.

Acknowledgement

The authors would like to thank for matching grants from VTU, Belgaum for the project sanctioned by Institution of Engineers, India. The authors would also like to thank TEQIP 1.2 grant, management of SDME society and Principal of SDM College of Engineering and Technology, Dharwad for providing all the support to carry out the research work.

REFERENCES

[1] Andrej Ljolje, Stephen E Levinson, "Development of Acoustic-Phonetic Hidden Markov Model for Continuous Speech Recognition, IEEE transactions on signal processing, vol. 39, No. 1, January 1991.

[2] L R Rabiner and Schafer, "Digital Processing of Speech Signals", Pearson Education, 1993.

[3] Sharada C. Sajjan, Vijaya C, "Comparison of DTW and HMM for Isolated Word Recognition", IEEEXplore, 2012.

[4] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi, "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques" Journal of computing, volume 2, issue 3, 2010.

[5] L R Rabiner and B H Juang, "Fundamentals of Speech Recognition", Prentice-Hall International, New Jersey, 1993.

[6] S. Young and others, "The HTK book(for HTK version 3.4)" Cambridge University Engineering Department, March 2009.

[7] HTK "Hidden Markov Model Toolkit", available at http://htk.eng.cam.ac.uk, 2012.

[8] R. Thangarajan, A M Natarajan, M Selvam, "Word and Triphone based Approaches in Continuous Speech Recognition for Tamil Language", WSEAS transactions on signal processing, Issue 3, Volume 4, March 2008.

[9] P C Woodland, J J Odell, V Valtchev, S J Young, "Large Vocabulary Continuous Speech Recognition using HTK", IEEE, 1994.

[10] L R Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE Vol. 77, Issue 2, pp. 257–286, 1989.

[11] Yunxin Zhao "A Speaker Independent Continuous Speech Recognition System using Continuous Mixture Gaussian Density HMM of Phoneme-Sized Units", IEEE transactions on speech and audio processing vol 1, No. 3, July 1993