

hand, enables the recognition system to look at the phoneme sequence as a whole. Thus, the first and second layers of the phoneme recognition system are based on the acoustic model and the phoneme model respectively. The HMM parameters in the acoustic model represent the distinguishing characteristics of each phoneme, while the phoneme model parameters represent the phoneme transitions and weights. We use the HMM to model the speech because the production process of the English phonemes in speech is assumed to be a discrete time-homogeneous Markov process [7].

4. Implementation

The implementation phase starts with capturing the speech signal from Microphone and storing it into digitized format. In every phase some or the algorithm is used to change the format of data being processed. Below are the algorithms used to implement auto query answering mechanism based on DSR:

a) VAD Algorithm

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

b) MFCC Algorithm

The Mel-Frequency Cepstral Coefficients (MFCC) of a speech signal are commonly used for obtaining good recognition results in speech and speaker recognition tasks. The MFCC are used extensively in the speech/speaker recognition literature for two reasons.

Firstly, MFCC have a low number of dimensions, which effectively avoids the curse of dimensionality for the recognizers. Secondly, MFCC closely relate to the biology of the filtering performed in the human ear. For these reasons we have included the MFCC extraction in our feature vector extraction system. There is a total of eight steps involved in obtaining the MFCC feature vector for a frame of speech. The steps in obtaining MFCC are follows:

Windowing:

The analog signal is sampled and windowed with below equation

$$x_n(t) = w(t)s(nS_f + t) \quad \text{for } t = 0, 1, \dots, L-1 \quad n = 0, 1, \dots, N-1$$

$S(t)$ is the sampled signal, n is the index, S_f is frame step size, L is the length of the window and N is the number of consecutive frames in the speech utterances.

Window function used is hamming window. $\alpha = 0.54$ and $\beta = 0.46$. $L > S_f$. Fast fourier transform is performed with vector length 256. The signal

- The speech is first pre-emphasized with a pre-emphasis filter to spectrally flatten the signal.
- Then the pre-emphasized speech is separated into short segments called frame. A frame can be seen as the result of the speech waveform multiplies a rectangular pulse

whose width is equal to the frame length. This will introduce significant high frequency noise at the beginning and end points of the frame because of the sudden changes from zero to signal and from signal to zero. To reduce this edge effect, an 80-points non-overlapping Hamming window is applied to each frame.

- After the FFT block, the spectrum of each frame is filtered by a set of filters, and the power of each band is calculated. To obtain a good frequency resolution, a 128-point FFT is used. Because of the symmetry property of FFT, we only need to calculate the first 64 coefficients. The filter bank consists of 33 triangular shaped band-pass filters, which are centred on equally spaced frequencies in the Mel domain between 0Hz and 4 kHz.
- We can calculate the Mel-Frequency Cepstrum from the output power of the filter bank. The final feature vector for each frame in the speech signal used in the phoneme recognition system (the output of the Feature Extraction system in Figure 2.1) is a 25 dimensional feature vector comprised of the concatenation of the 12-D MFCC coefficients obtained in step 7, the 12-D delta coefficients (2.12) and the 1-D delta log-power coefficient (2.15) (both obtained in step 8). It is important to note the probability distributions of the speech signal before and after the MFCC transformation (step 7). If the speech signal is modeled to have a Gaussian distribution, then the Cepstral features are Rayleigh distributed, the Mel-Warped Power Cepstral features (step 4) will be Chi-Square distributed and the log Cepstrum and the MFCC features can be accurately modeled as a mixture of multivariate Gaussian distributions [13]. This gives the advantage for modeling the MFCC domain signal as a Gaussian Mixture Model (GMM).

c) Phoneme Segmentation

Phoneme segmentation is nothing but extracting phonemes and training monophone HMMs for each of the phoneme. For extracting phonemes, we provide a text file containing the details about what has been spoken while recording the wav file during training phase. This file carries the file name, and corresponding sentence spoken.

Sample1: What is a process

Considering the dictionary which carries phonetic sequence of every word used in the vocabulary, a new file is generated which carries the phonetic sequence present in every file.

Sample1.lab: silhw a t sp I s sppr o cess /sil

Where sil represents, the start and end of sentence and sp represents short pause between words. For extracting the acoustic models of every phoneme, the starting point is a set of identical monophone HMMs in which every mean and variance is identical. These are then retrained, short-pause models are added and the silence model is extended slightly. The monophones are then retrained.

The first step in HMM training is to define a prototype model. The parameters of this model are not important; its purpose is to define the model topology. For phone-based systems, a good topology to use is 3-state left-right with no skips such as in **Error! Reference source not found.** where each ellipsed vector is of length 39. This number, 39, is computed from the length of the parameterized static vector

(MFCC 0 = 13) plus the delta coefficients (+13) plus the acceleration coefficients (+13). The data files are scanned to compute the global mean and variance and set all of the Gaussians in a given HMM to have the same mean and variance.

Hence, a new version of proto is created in the directory `hmm0` in which the zero means and unit variances below have been replaced by the global speech means and variances. Note that the prototype HMM defines the parameter kind as MFCC 0 D A (Note: 'zero' not 'oh'). This means that delta and acceleration coefficients are to be computed and appended to the static MFCC coefficients computed and stored during the coding process. To ensure that these are computed during loading, the configuration file configuration should be modified to change the target kind, i.e. the configuration file entry for TARGETKIND should be changed to

```
TARGETKIND = MFCC_0_D_A
```

Given this new prototype model stored in the directory `hmm0`, a Master Macro File (MMF) called `hmmdefs` containing a copy for each of the required monophone HMMs is constructed by manually copying the prototype and relabeling it for each required monophone (including "sil"). The format of an MMF is similar to that of an MLF and it serves a similar purpose in that it avoids having a large number of individual HMM definition files.

It is a continuous density HMM with 5 states in total, 3 of which are emitting. The first symbol in the file `~h` indicates that the following string is the name of a macro of type `h` which means that it is a HMM definition. The HMM definition itself is bracketed by the symbols `<BeginHMM>` and `<EndHMM>`. The first line of the definition proper specifies the global features of the HMM. In any system consisting of many HMMs, these features will be the same for all of them. In this case, the global definitions indicate that the observation vectors have 4 components (`<VecSize>4`) and that they denote MFCC coefficients (`<MFCC>`). The next line specifies the number of states in the HMM. There then follows a definition for each emitting state `j`, each of which has a single mean vector `lj` introduced by the keyword `<Mean>` and a diagonal variance vector `sj` introduced by the keyword `<Variance>`. The definition ends with the transition matrix `{aij}` introduced by the keyword `<TransP>`. There is no definition for the number of input data streams or for the number of mixture components per output distribution.

Hence, in both cases, a default of 1 is assumed.

The allowable transitions between states should be indicated by putting non-zero values in the corresponding elements of the transition matrix and zeros elsewhere. The rows of the transition matrix must sum to one except for the final row which should be all zero. Each state definition should show the required number of streams and mixture components in each stream. All mean values can be zero but diagonal variances should be positive and covariance matrices should have positive diagonal elements. All state definitions can be identical

d) HMM Training

HMM training is performed for building sub-word systems in which the basic units are the individual sounds of the language called phones. One HMM is constructed for each such phone and continuous speech is recognized by joining the phones together to make any required vocabulary using a pronunciation dictionary. It starts by uniformly segmenting the data and associating each successive segment with successive states. Of course, this only makes sense if the HMM is left-right. If the HMM is ergodic, then the uniform segmentation can be disabled and some other approach taken.

If any HMM state has multiple mixture components, then the training vectors are associated with the mixture component with the highest likelihood. The number of vectors associated with each component within a state can then be used to estimate the mixture weights. In the uniform segmentation stage, a K-means clustering algorithm is used to cluster the vectors within each state. The probability of an observation being associated any given Gaussian mixture component is determined. This occupation probability is computed from the forward and backward probabilities.

Baum-Welch training is similar to the Viterbi training described in the previous section except that the hard boundary implied by the \hat{A} function is replaced by a soft boundary function L which represents the probability of an observation being associated any given Gaussian mixture component. This occupation probability is computed from the forward and backward probabilities.

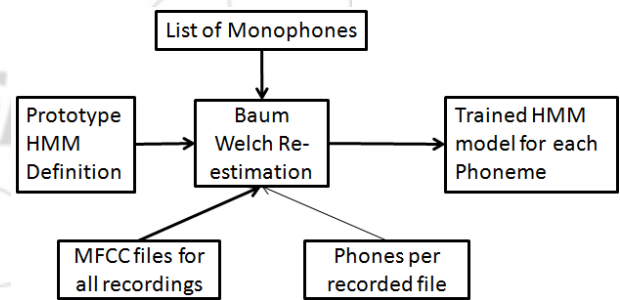


Figure 2: Baum Welch Algorithm

e) Viterbi Decoder

Decoding is controlled by a recognition network compiled from a word-level network, a dictionary and a set of HMMs. The recognition network consists of a set of nodes connected by arcs. Each node is either a HMM model instance or a word-end. Each model node is itself a network consisting of states connected by arcs. Thus, once fully compiled, a recognition network ultimately consists of HMM states connected by transitions. However, it can be viewed at three different levels: word, model and state.

For an unknown input utterance with T frames, every path from the start node to the exit node of the network which passes through exactly T emitting HMM states is a potential recognition hypothesis. Each of these paths has a log probability which is computed by summing the log probability of each individual transition in the path and the log probability of each emitting state generating the

corresponding observation. Within-HMM transitions are determined from the HMM parameters, between-model transitions are constant and word-end transitions are determined by the language model likelihoods attached to the word level networks.

The job of the decoder is to find those paths through the network which have the highest log probability. These paths are found using a Token Passing algorithm. A token represents a partial path through the network extending from time 0 through to time t . At time 0, a token is placed in every possible start node.

Each time step, tokens are propagated along connecting transitions stopping whenever they reach an emitting HMM state. When there are multiple exits from a node, the token is copied so that all possible paths are explored in parallel. As the token passes across transitions and through nodes, its log probability is incremented by the corresponding transition and emission probabilities. A network node can hold at most N tokens. Hence, at the end of each time step, all but the N best tokens in any node are discarded.

As each token passes through the network it must maintain a history recording its route. The amount of detail in this history depends on the required recognition output. Normally, only word sequences are wanted and hence, only transitions out of word-end nodes need be recorded. However, for some purposes, it is useful to know the actual model sequence and the time of each model to model transition. Sometimes a description of each path down to the state level is required. All of this information, whatever level of detail is required, can conveniently be represented using a lattice structure.

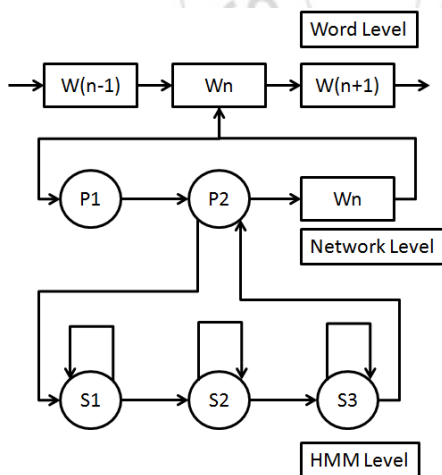


Figure 3: HMM Layers

5. Conclusion

The system adopts a distributed architecture in which the speech recognizer and the knowledge-based IR system are located in different servers. The spoken queries are processed using the DSR technology. With this technology, there is a great help provided to the society by enabling the vision impaired people from successfully using the internet services to the fullest.

Future work will consider using n -gram language models in the DSR server. It also needs implementation of VAD and MFCC algorithms. Along with IR methodology to provide real time correct responses for the queries raised by user. Post this implementation; the system can be designed for N number of clients interacting with the DSR and IR servers. There can be a prototype defined for the network over which all clients communicate with DSR server and IR server.

References

- [1] WeiQi Zhang, Archit. Dev. Lab., Intel Corp., China ; Liang He, Yen-Lu Chow, RongZhen Yang, "The study on distributed speech recognition system", Vol 3, June 2000
- [2] Junhui Zhao, Xiang Xie, Jingming Kuang, "The Performance Evaluation of Distributed Speech Recognition for Chinese Digits", *Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA'03)*, IEEE Computer Society, March 2003
- [3] Wenjing Han ; Inst. for Human -Machine Commun ., Tech. Univ. Munchen, München, Germany ; Zixing Zhang ; Jun Deng ; Wollmer, M, "Towards distributed recognition of emotion from speech", *IEEE Conference Publication*, Vol 3, June 2013
- [4] Ian Bennett, Palo Alto, CA (US), "Method For Processing Speech Data For a Distributed Recognition System" *United State Patent, US 7672841*, March 2010.
- [5] Alexander Sorin, Haifa (IL), "Restoration of High-Order Mel Frequency Cepstral Coefficients", *United States Patent, US 20090144058*, June 2009
- [6] Naveen Srinivasamurthy, Antonio Ortega, Shrikanth Narayanan, "Efficient scalable encoding for distributed speech recognition", *United States Department of Electrical Engineering-Systems, Signal and Image Processing Institute, Integrated Media Systems Center*, November 2005.
- [7] Tom Brøndsted1, Henrik Legind Larsen2, Lars Bo Larsen1, Børge Lindberg1, Daniel Ortiz-Arroyo2, Zheng-Hua Tan1, Haitian Xu1, "Mobile Information Access with Spoken Query Answering", *Department of Communication Technology 2 Software Intelligence and Security Research Center (SIS-RC)*, Esbjerg Aalborg University, Denmark, December 2012.
- [8] J A Gonzalez, A Lopez-Lopez, J Munoz-Arteaga, M Montez, Y Gomez, "Natural Language Dialogue System for Information Retrieval", *Institute National Astrophysics*, May 2010.
- [9] M. H. Moattar and M. M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm", 17th European signal processing conference, August 2009