International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Distributed Speech Recognition HMM Modelling

Gauri A. Deshpande[1], Pallavi S. Deshpande[2]

[1]PG Student (Signal Processing), E&TC, SKNCOE, Pune, India

[2]Assistant Professor, E&TC, SKNCOE, Pune, India

Abstract: ***Speech Recognition performed over a network is referred to as Distributed speech recognition (DSR). It is a technique that enables access to services and communication systems without the need to type or use a keypad. The primary objective of speech recognition is to enable all of us to have easy access to the full range of computer services and communication systems, without the need for all of us to be able to type, or to be near a keyboard. It makes use of client server architecture to enable the distributed nature. The use of HMM models for modeling acoustic parameters of speech delivers the price/performance levels which are acceptable, practicable and affordable. As just one example of a spectrum of possible new applications, we will be able to dictate our meeting notes directly into our enhanced cellular handset immediately after a meeting, and the draft text will already be in our personal computer, ready for editing, by the time we return to our office (or hotel room, or home). The performance of speech recognition systems receiving speech that has been transmitted over mobile channels can be significantly degraded when compared to using an unmodified signal. The degradations are as a result of both the low bit rate speech coding and channel transmission errors. A Distributed Speech Recognition (DSR) system overcomes these problems by eliminating the speech channel and instead using an error protected data channel to send a parameterized representation of the speech, which is suitable for recognition. We are using Julius engine for core speech processing functionalities. Julius engine also supports client server model. A deep understanding of all the modules along with baum-welch re-estimation and viterbi algorithm is achieved.***

Keywords: Distributed Speech Processing; Speech Recognition; Client-Server model; HMM, Baum Welch Re-estimation, Viterbi Decoder;

1. Introduction

Communication is the most crucial part of human life. Amongst all the communication mechanisms humans can have, voice communication is the primary method. With speech communication, several advantages like faster and more effective. The speech produced and perceived by people is a set of periodic variations of pressure propagating through the air. These air vibrations are fueled by the human lungs. The vocal tract subsystem of the human auditory system is responsible for the shaping and the production of the actual sound. The human vocal tract consists of the pharynx, mouth, and nose cavities. The air generated by the lungs goes to the human glottis. This generated air is responsible for the production of the vowels and voiced sounds and also generates a pulse train for the vocal tract. A noise generated by the human glottis results in consonants or unvoiced sounds. Speech recognition refers to the extraction of verbal information from the speech utterance. In other words, a speech recognition system takes the speech utterance as the input and produces a text output that corresponds to the given speech. Initially the variations in the air pressure are converted into an analog signal by the use of a microphone or a telephone headset.

Then the analog signal is converted into a digital signal by the use of the analog to digital converter. An anti-aliasing low-pass filter is placed directly before the digital sampled speech for the rejection of any would-be high-frequency aliasing components, channel noise, and/or inter-channel interference. Ideally, the low-pass anti-aliasing filter would have a cut-off frequency at half of the Nyquist frequency, which corresponds to half of the sampling frequency. To put it into simple terms: The computer learns how words - or more correctly the sounds that make up those words - sound.

A speech model for continuous speech recognition consists of three distinct parts [4]:

- Language Model
- Acoustic Model
- Phonetic Model

2. Literature Survey

A. Spoken Query System

The applications which can mimic normal human interactions are likely to be preferred by potential on-line shoppers and persons looking for information over the WWW. It is also expected that the use of voice-based systems will increase the universe of persons willing to engage in e-commerce, e-learning etc. Application such as, various commercial programs sold by IBM (VIAVOICE) and Kurzweil (DRAGON) [2] permit some user control of the interface (Opening and closing files) and searching but they do not present a flexible solution that can be used by a number of users across multiple cultures and without time consuming voice training.

B. Efficiency Improvement with DSR

Another issue represented by the lack of voice based systems is efficiency. Many companies are now offering technical support over the Internet [2] and some even offer live operator assistance for such queries. While this is very advantageous, it is also extremely costly and inefficient, because a person must be employed to handle such queries. This presents a practical limit those results in long wait times for responses or high labor overheads.

C. Model Used

A typical speech recognizer consists of two distinct components [6]-

- A feature extractor (At client side)

- A pattern recognizer (At server side)- It makes use of acoustic models, language models and other domain specific knowledge.

When client–server architecture is adopted for speech recognition, it is essential for the client to compress the speech before transmission to the server in order to conserve bandwidth and power.

D. Different Recognition Algorithms Used

Natural language processing is concerned with the parsing, understanding and indexing of transcribed utterances and larger linguistics units [4]. Because spontaneous speech contains many surface phenomena such as disfluencies, hesitations, repairs and restarts. Discourse markers such as 'well' and other elements which cannot be handled by the typical speech recognizer, it is the problem and the source of the large gap that separates speech recognition and natural language processing technologies. Except for silence between utterances, another problem is the absence of any marked punctuation available for segmenting the speech input into meaningful units such as utterances. For optimal NLP performance, these types of phenomena should be annotated at its input.

In speaker dependent speech recognition system, interface is trained with the user's voice which takes a lot of time and is thus very undesirable from the perspective of a WWW environment. A user may interact only a few times with a particular website. Furthermore, speaker dependent systems usually require a large user dictionary (one for each unique user) which reduces the speed of recognition. This makes it much harder to implement a real time dialog interface with satisfactory response capability. i.e. something that mirrors normal conversation on the order of 3-5 seconds is probably ideal.

In a typical language understanding system, there is typically a parser that precedes the semantic unit. Although the parser can build a hierarchical structure that spans a single sentence, parser are seldom used to build up the hierarchical structure of utterances or text that spans multiple sentences.

The syntactic marking that guides parsing inside a sentence is either a weak or absent in a typical discourse. So for a dialog based system that expects to have smooth conversational features, the emphasis of the semantic decoder is not only on building deeper meaning structures for the shallow analyses constructed by the parser, but also on integrating the meanings of the multiple sentences that constitute the dialog.

Until now there are two new research paths taken in deep semantic understanding of language: informational and intentional. In the informational approach, the focus is on the meaning that comes from the semantic relationships between the utterance level propositions (e.g. effect cause, conditions) whereas with the intentional approach, the focus is on recognizing the intentions of the speaker (e.g. inform, request, propose).

3. Phases

a) Training & Testing

The training of the speech recognition system is a three step process. The training of the first and second layers is done separately. First, the Feature Extraction system is responsible for the extraction of the features from a collection of speech utterances. Second, the first layer training process generates a set of templates, each of which has a set of defining parameters. Finally, the second layer training process extracts the transition information from the phonemes and their length statistics from the speech database. After the training data has been generated, the feature extraction system generates the feature vector set for a test utterance. Then the first and second layer recognition systems together classify each of the feature vectors to one of the corresponding templates. Finally the system maps each of the templates to its corresponding phonemes and the second layer recognition system outputs the sequence of phonemes that was recognized from the speech signal.

b) Recognition Layers

The speech recognition system consists of two layers of recognition. The first recognition layer is responsible for analyzing a set of acoustic feature vectors and determining the likelihood of the acoustic feature vector set being matched to a particular template. The second recognition layer is responsible for determining the most likely sequence of the templates, given the set of acoustic feature vectors that was extracted from the test speech utterance.

The implementation of these two recognition layers forms the two interconnected subsystems of the phoneme recognition system. The subsystem implementing the second layer uses the subsystem of the first layer for determining the best sequence of the set of templates for the given speech utterance. Each of these two subsystems has its own set of algorithms for the corresponding training and recognition phases.

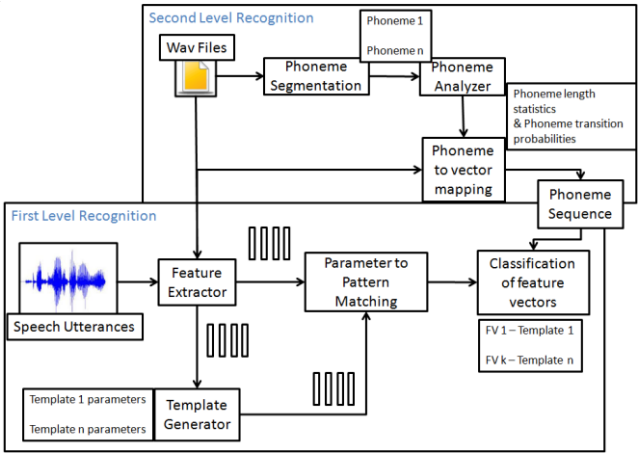

Figure 1: Layered Recognition

c) HMM Model

The training and the recognition in the two layers of the automatic speech recognition system is based on two HMM models. Those are the acoustic model and the phoneme model. The acoustic model of speech models each individual phoneme based on the feature vectors gathered from the speech training database. The phoneme model, on the other

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

$$x_n(t) = w(t)s(nS_F + t) \quad for \quad t = 0, 1, \ldots, L-1 \quad n = 0, 1, \ldots, N-1$$

S(t) is the sampled signal, n is the index, Sf 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 is 0.46.L > Sf. 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 ¹j introduced by the keyword <Mean>and a diagonal variance vector §j 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. Allstate 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 Ã 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.

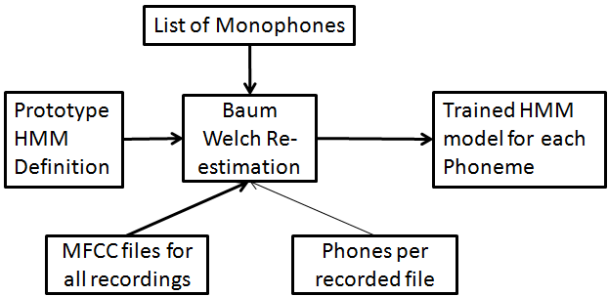

Figure 2: Baum Welch Algorithm

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.

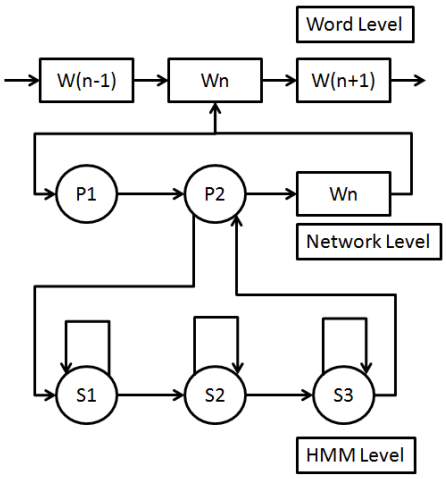

Figure 3: HMM Layers

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 the17 th 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