ISSN: 2277 – 9043
International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)
Volume 3, Issue 1, January 2014

Automatic Speech Recognition: A Survey

Rajan Mehla, Mamta, R.K.Aggarwal

***Abstract*— This paper explains the concept of Automatic speech recognition (ASR) from the view point of pattern recognition. An ASR system can be broadly classified into two parts: front end and the back end which are responsible for feature extraction and acoustic modelling respectively. The presented paper elaborates and compares all popular feature extraction and acoustic modelling techniques along with the challenges and advancements in the field of ASR.**

***Index Terms* – automatic speech recognition, acoustic models, back-end, feature extractors, front-end**

I. INTRODUCTION

Speech recognition is the ability to listen spoken words and identify various sounds present in it, and recognize them as words of some known language. In computer domain the speech recognition may be defined as the ability of computer system to accept spoken words in audio format like wav and then generate its content in text format.

Automatic speech recognition (ASR) simulates human listening, it transforms speech into text. The problem of automatic speech recognition (ASR) is to program a computer to take digitized speech samples and print the words that a human would recognize when listening to the same sound.

ASR is one of the problems in the field of pattern recognition (PR). ASR has grown roughly in proportion to other areas of pattern recognition because of the desire to invent a machine capable of making complex decisions, and, practically, one that could function as swiftly as humans. In order to achieve this machine should understand speech. As in any PR task, ASR seeks to understand patterns in an input speech. During the processing of audio signals such as speech, there is a wide range of interesting patterns needs to be distilled from the speech signals, same as the task of pattern recognition.

To get knowledge of how speech recognition problems can be approached today, a review of some research highlights and challenges will be presented

Manuscript received Jan , 2014.

***Rajan Mehla**, Computer Engg. Dept., National Institute of Technology, Kurukshetra, India, 9996778289*

***Mamta**, Computer Engg. Dept., National Institute of Technology, Kurukshetra, India, 9996778287*

***R.K.Aggarwal**, Computer Engg. Dept., National Institute of Technology, Kurukshetra, India, 9416570828*

A. Brief history

The earliest attempts to devise systems for automatic speech recognition by machine were made in the 1950's, when researchers tried to exploit the fundamental ideas of acoustic-phonetics. In 1952, at bell Laboratories, Davis, Biddulph, and Balashek built a system for isolated digit recognition for a single speaker [1]. In 1959 another attempt was made by Forgie and Forgie, at MIT Lincoln Laboratories. Ten vowels embedded in a /b/-vowel-/t/ format were recognized in a speaker independent manner [2]. In 1970's the field of speech recognition achieved a number of significant milestones. First, the area of isolated word recognition became a viable and usable technology based on fundamental studies by Velichko and Zagorukyo in Russia [3], Sakoe and Chiba in Japan [4] and Itakura in the united states [5]. The Russian studies helped advance the use of pattern recognition ideas in speech recognition; the Japanese research showed how dynamic programming methods could be successfully applied; and Itakura's research showed the ideas of Linear Predictive Coding (LPC). At AT&T Bell Labs, began a series of experiments aimed at making speech recognition systems that were truly speaker independent [6]. They used a wide range of Clustering algorithms to determine number of distinct patterns required to represent all variations of different words across wide user population. In 1980's a shift in template-based approaches to statistical modeling methods occur, especially in the Hidden Markov Model (HMM) approach [7].

B. Challenges

The biggest challenge for ASR is to handle variability in speech. Since each person has a different vocal tract (VT), controlled by a unique brain, therefore a large range of variability exists in speech signals. In addition, it is impossible for humans to reproduce the same exact action twice; even when attempting to repeat a word uniformly, slight variations occur. These changes are readily observed in digital representations of speech signals.

In "speaker-dependent" (SD) automatic speech recognition system, speech variations are typically less vast (vs. "speaker-independent" (SI) cases, where an ASR system makes no assumption of who is talking). However, even when speech is limited to one cooperative speaker, significant variations are often evident owing to environmental (e.g., different communication channels) and speaking conditions (e.g., words in different contexts). When we generalize the ASR task to be SI, as in most services for the general public, we face the much larger range of variability that arises from different people, with their varied VTs and diverse styles of speaking.

Another major challenge for ASR is to overcome the "mismatch" problem, where very often a system is faced with testing speech that is a poor match for the speech the recognizer was trained on. For ASR, a set of speakers typically reads chosen texts, and models are developed from this speech. ASR accuracy is usually proportional to the empirical similarity between training and testing data. For example, we may get high accuracy if an ASR model is properly developed for a single speaker repeating a word many times in a quiet environment, then testing the system with new versions of that same word from that speaker in the same environment. However, if we then test on a different speaker, with a different microphone, or add some background noise, we usually get reduced (and often much lower) accuracy. This is called the mismatch problem. The challenge for ASR designers is to amass sufficient data and employ a good training algorithm.

The most difficult variability that ASR must handle is due to background, channel noise, and other external distortions [8]. Basic spectral subtraction techniques can help with additive noise, while some cepstral methods (which convert multiplication in the spectral domain to cepstral addition) suppress convolutional noise. Many methods that are used to enhance noisy speech for human listening can be used as preprocessors for ASR. In noisy cases, one should focus on the high-amplitude parts of the input signal spectrum because these are most relevant for speech perception, and are relatively less corrupted by noise [9].

Cepstral mean subtraction (CMS), like RASTA processing [10], eliminates very slowly varying signal aspects (presumed to be mostly from channel distortion). The mean value for each parameter over time (typically for periods exceeding 250 ms) is subtracted from each frame's parameter, thus minimizing environmental and intra-speaker effects. Channel noise is often assumed to be constant over an utterance, but portable telephones suffer fading channel effects, which require more frequent estimations [11].

II. Classification of ASR System

Speech recognition problem can be divided into following 3 major categories as shown in the Fig. 1.

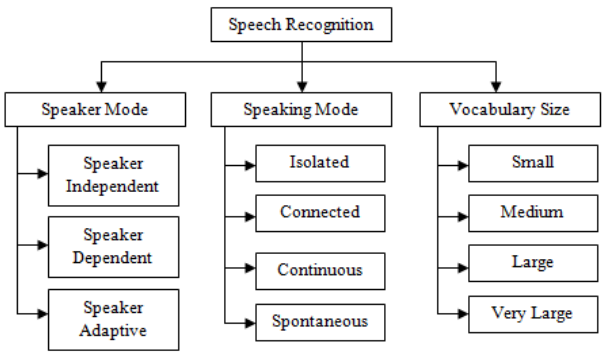

Fig. 1: Speech Recognition Classification

A. Classification based upon Speaker Mode

On the basis of speaker mode, speech recognition systems can be classified as speaker dependent, speaker independent and speaker adaptive systems. Speaker dependence describes the degree to which a speech recognition system requires knowledge of the speaker's individual voice characteristics to successfully recognize the speech.

Speaker dependent speech recognition system: Speech recognition systems that require a user to train the system for his/her voice are known as speaker dependent systems. These systems are usually easier to develop, cheaper to buy and more accurate. But these systems are not as flexible as speaker adaptive or speaker independent systems.

Speaker independent speech recognition system: Speech recognition systems that do not require a user to train the system are known as speaker independent systems. A speaker independent system is developed to operate for any speaker. These systems are the most difficult to develop and have less accuracy but more expense than speaker dependent systems.

*Speaker adaptive speech recognition system***:** A speaker adaptive system is developed to adapt its operation to the characteristics of new speakers. It lies somewhere between speaker dependent and speaker independent systems.

B. Classification based upon Speaking Mode

The speech recognition systems can be categorized into several different classes such as isolated words, connected words, continuous speech and spontaneous speech. A brief of each is given as below [12]:

Isolated word speech recognition system: Isolated word recognizers (IWR) accept single word or single utterance at a time. It usually requires each utterance to have quiet (lack of an audio signal) on both sides of the sample window. This is the simplest form of recognition to perform because end points are easier to find and the pronunciation of a word doesn't affect others. It is important to note that even though IWR has a very limited recognition power; it has many applications in the real world.

Connected words speech recognition system: Connected word systems (or more correctly 'connected utterances') are similar to isolated words, but allows separate utterances to be 'run-together' with a minimal pause between them.

Continuous speech recognition system: A continuous speech system operates on speech in which words are connected together i.e. not separated by pauses. It allows users to speak almost naturally, while the computer determines the content. Recognizers with continuous speech capabilities are somewhat difficult to create because it is difficult to find the start and end points of words.

Spontaneous speech recognition system: A spontaneous speech recognition system has the ability to handle a variety

of natural speech features such as words being run together, "ums" and "ahs", and even slight stutters.

C. Classification based upon Vocabulary

ASR can also be classified based upon the size of the vocabulary as small size vocabulary, medium size vocabulary, large vocabulary and very large vocabulary speech system. The size of the vocabulary of a speech system affects the complexity, processing requirements and the accuracy of the system. A small size vocabulary consists of up to ten words. Vocabulary of hundreds of words is used by the medium vocabulary speech system. A large vocabulary has thousands of words whereas very large vocabulary contains tens of thousands of words.

III. System Architecture

The basic model of ASR system is divided into two ends [13], front-end and back-end as shown in Fig. 2.

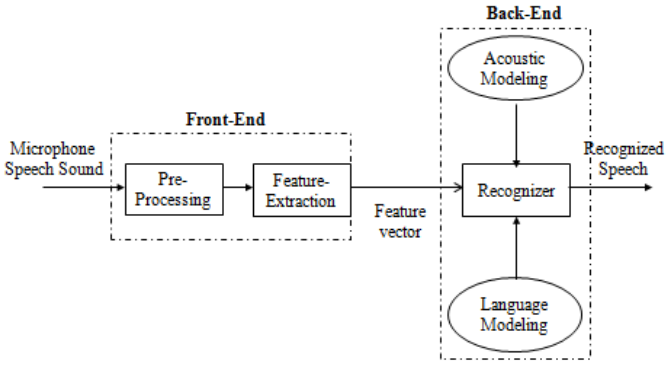

Fig. 2: Architecture of ASR System [13]

The sound waves captured by a microphone are fed to the front-end module. In this module the input speech is first pre-processed and then converted into series of feature vectors (observation vectors) which are then forwarded to the back-end. At back-end, recognizer/decoder module comes up with the results as plain text with the help of acoustic and language models.

A. Front-End of ASR

Input speech signal can be captured with the help of microphone. While capturing the audio speech signal, noise from external environment causes the captured signal to be noisy and faulty. To remove such noise, it is very essential to apply noise filtering process. The main purpose of using this step is to capture noise-free signal i.e. a speech signal with as high signal-to-noise ratio as possible [14]. The various sources of noise are air conditioning system, fans, fluorescent lamp, type writers, footsteps, opening and closing doors, electrical humming etc. The effect of noise can be minimized by the following methods:

- Prevention (applied before capturing the audio)
  - Close speaking microphone.
  - Speaker to microphone distance should be 10-12 inch.
  - Close speaking microphones may also exhibit somewhat poorer response characteristics.
- Filtering (applied after capturing the audio)
  - High pass filter the signal with cut-off frequency ~ 80-120 Hz (discussed below)

Front-end of an ASR mainly covers pre-processing and feature extraction phases. These are explained as below:

Pre-processing

Speech-signal is an analog waveform which cannot be directly processed by digital systems. Hence, it needs to be pre-processed. Pre-processing or signal processing includes mainly 2 steps: the sampling process and spectral analysis process [15], as shown in the Fig. 3.

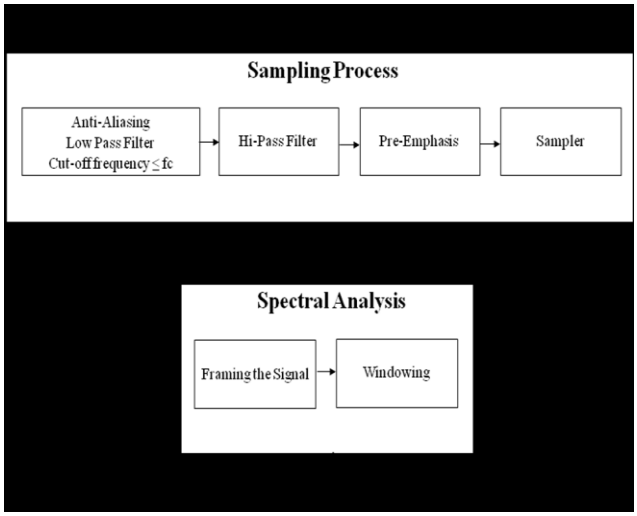

Fig. 3: Pre-processing of Speech Signal

Feature Extraction

After the pre-processing of speech signal, the output of spectral analysis i.e. windowed data is sent for discrete Fourier transform (DFT) and spectral estimation. This comes under feature extraction. The goal of feature extraction is to find a set of properties of an utterance that have acoustic correlation to the speech signal, that is, parameters that can somehow, be computed or estimated through processing of the signal waveform. Such properties are termed as features. It typically includes the process of converting the signal to a digital form (i.e. signal conditioning), measuring some important characteristic of the signal such as energy or frequency response (i.e. signal measurement), augmenting these measurements with some perceptually-meaningful derived measurements (i.e. signal parameterization), and statistically conditioning these numbers to form observation vectors. The steps covered in this phase are preprocessing, filter bank analysis and cepstrum generation as given below:

- An analog to digital conversion of the speech signals at discrete time intervals using a sampling rate of 8 kHz to 16 kHz.

- Fixed length window based frame segmentation by applying a Hamming window of 20 to 30 ms size at every 10–20 ms interval.
- High pass filtering based pre-emphasis to spectrally flatten the signal by boosting it approximately 20 dB per decade.
- FFT based transformation to convert the time domain presentation of speech signal into spectral domain.
- Filter bank analysis based on human auditory system.
- Cepstrum generation to decorrelate the spectral feature.

$$C(n) = DCT\left(\log\left(\left|FFT\left(s(n)\right)\right|\right)\right) \qquad (1)$$

- In order to capture the dynamic nature of the speech signal, cepstral vectors are augmented with "delta" parameters computed by taking the first and second differences (or derivatives) of the features in successive frames. It includes some information from a larger time span by measuring the change in feature values.

$$d_i = \frac{\sum_{n=1}^{N} n(c_{n+i} - c_{n-i})}{2\sum_{n=1}^{N} n^2} \qquad (2)$$

Where:

- d_i is a delta coefficient for frame i computed in terms of corresponding basic coefficients c_{n+i} and c_{n-i}
- N are the total input frames of data

The same equation is used to compute the acceleration coefficients by replacing the basic coefficients with delta coefficients.

The feature extraction process [16] is expected to discard irrelevant information of the task while keeping the useful one as depicted in Fig. 4. The following properties are required for a good feature extractor:

- Compact features to enable real time analysis
- Minimize the loss of discriminant information

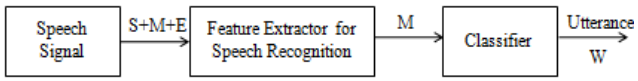

Fig. 4: Feature Extractor

A feature extractor (FE) for ASR performs a specific task. As shown in Fig. 4, the speech signal contains the characteristic information of the speaker (S) and environment (E) in addition to signal message (M). A feature extractor for speech recognition needs to maximally discard the speaker and environmental information and only allow the signal information to filter from the speech signal. The ability of FE for speech recognition improves depending upon how well S and E are filtered out.

A common division of the feature extraction approaches is into production-based and perception-based methods. Linear predictive coding (LPC), Linear Predictive Cepstral Coefficient (LPCC) is an example from the first group while Mel-frequency cepstral coefficients (MFCC) and Perceptual Linear Prediction (PLP) belong to the perception-based approaches family. Broadly, the feature extraction techniques are classified as temporal analysis and spectral analysis techniques.

- In temporal analysis the speech waveform itself is used for analysis.
- In spectral analysis spectral representation of speech signal is used for analysis.

There are several techniques to process and extract features from speech signal (frame) as given below:

- Linear predictive cepstral coefficients (LPCC)
- Mel frequency cepstral coefficients (MFCC)
- Perceptual linear prediction (PLP)
- PLP derived from Mel filter bank (MF-PLP)
- Relative spectral- perceptually based linear prediction (RASTA-PLP)
- Temporal patterns (TRAP) and TANDEM

Linear Predictive Cepstral Coefficients

LPCC is the extension of linear predictive coding (LPC). LPC is a production based method. In this, the speech sampled at time 'n' can be presented as a linear combination of 'p' previous samples or we can say it tries to predict the current output as a linear combination of previous outputs

$$s[n] \approx \sum_{k=1}^{p} a[k]s[n-k] \qquad (3)$$

Where, s[n] is the nth sample of signal, a[k] is the predictor coefficients; p is the no. of past samples also known as order of the predictor [17]. The total squared prediction error is:

$$E = \sum_{n} (s[n] - \sum_{k=1}^{p} a[k]s[n-k])^2 \qquad (4)$$

The objective of the linear predictive analysis is to determine the coefficients a[k] for each speech frame such that E is minimized. The basic idea behind linear predictive coding is that a speech sample can be approximated as a linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite internal) between the actual samples and the linearly predicted ones, a unique set of predictor coefficients can be determined [17]. Linear-predictive coding can be readily shown to be closely related to the basic model of speech production in which the speech signal is modeled as the output of a linear, time varying system excited by either quasi periodic pulses (for voiced sounds) or random noise (for unvoiced sounds) [18,19].

Mel-Frequency Cepstral Coefficients

In MFCC, a speech spectrum passes through a filter bank of Mel-spaced triangular filters as shown in Fig. 5, and the filter output energies are log-compressed and transformed to the cepstral domain by DCT. The spacing of filter bank follows the Mel frequency scale, which was introduced by

Davis and Mermelstein [20] in 1980 inspired by human auditory system. The Mel scale is defined as follows:

$$f_{mel} = 2595 log_{10}\left(1 + \frac{freq}{700}\right) \qquad (5)$$

f_{mel} is the subjective pitch in mels corresponding to a frequency in hertz.

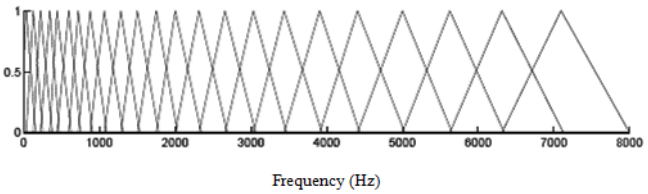

Fig. 5: Mel scaled filter bank for 0-8 kHz frequency range

The Mel-scale is a mapping from a linear to a nonlinear frequency scale based on human auditory perception. It increases significantly the performance of ASR in comparison with the linear scale. This scale usually covers the 156-6844 Hz frequency range and is logarithmic above 1 kHz and linear below this frequency. MFCC are the coefficients derived from cepstral representation of audio signals. It approximates the human auditory system more closely than the linearly-spaced frequency bands used in normal cepstrums. MFCC is ideal for speech recognition because it takes human perception sensitivity with respect to frequencies during feature extraction process.

To implement this filter bank, each window of speech data is processed using short time Fourier transforms and then magnitude is computed. The magnitude coefficients are then binned by correlating them with each triangular filter. Thus these filters are applied to the log of the magnitude spectrum of the signal, which is estimated on a short-time basis. Normally first 13 coefficients are enough for the representation of the signal. This cepstral as such, along with their first and second order derivatives are used as features for recognition.

As it can be seen, endpoints of each filter are defined by the central frequencies of adjacent filters. Bandwidths of the filters are determined by the spacing of filter central frequencies which depend on the sampling rate and the number of filters. That is, if the number of filters increases, the number of MFCC increases and the bandwidth of each filter decreases.

Perceptual Linear Prediction

In PLP the speech spectrum is modified by a set of transformations that are based on models of the human auditory system. It incorporates critical band spectral resolution into its spectrum estimate by remapping the frequency axis to the Bark scale and integrating the energy in the critical bands to produce a critical band spectrum approximation.

At conversational speech levels, human hearing is more sensitive to the middle frequency range of the audible spectrum. PLP incorporates the effect of this phenomenon by multiplying the critical band spectrum by an equal loudness curve that suppresses both the low and high frequency regions relative to the midrange from 400 to 1200 Hz. There is a nonlinear relationship between the intensity of sound and the perceived loudness. PLP approximates the power law of hearing by using a cube root amplitude compression of the loudness equalized critical band spectrum estimate [21]. It is a technique for speech analysis that uses three psycho-acoustic concepts to estimate the auditory spectrum:

- Critical-band spectral analysis
- The equal loudness curve
- The intensity power law

Perceptual linear prediction, is similar to LPC analysis, is based on the short-term spectrum of speech. In contrast to pure linear predictive analysis of speech, perceptual linear prediction (PLP) modifies the short-term spectrum of the speech by several psychophysically based transformations. The perceptual linear prediction method modifies basic LPC to closely model the psycho-acoustics of hearing using a critical-band power spectrum with a logarithmic amplitude compression. The spectrum is multiplied by a mathematical curve modeling the ear's behavior in judging the loudness as a function of frequency. The output is then raised to the power 0.33 to simulate the power law of hearing. To obtain the auditory spectrum, 17 band pass filters equally spaced on the Bark scale along a frequency range of 0-5 KHz are used. Their center frequency is defined by:

$$z = 6 \log\left(\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right) \qquad (6)$$

Where f is the frequency in Hz and z covers the range of 0-5 KHz by the 17 band pass filters (i.e. $0 \leq z \leq 17$ Bark).

PLP Derived from Mel Filter Bank

In this method, the MFCC and PLP techniques are merged into one algorithm. The first steps until generating the output of the Mel scale triangular filter bank are taken from the MFCC algorithm [22]. The only difference here is that the filter bank is applied to the power spectrum instead of the magnitude spectrum. The last steps generating the cepstrum coefficients are taken from the PLP algorithm. The 20 filter bank outputs are modified by the intensity loudness law. The 16 cepstrum coefficients are calculated from the output of the intensity loudness law via the all-poles approximation [23]. Finally, cepstral mean normalization is applied.

Relative Spectral- Perceptually Based Linear Prediction

In real world applications, ASR systems often encounter situations in which a mismatch between training and testing conditions exists (e.g. noise, transmission channel or the intra- or inter-speaker variations). In such cases, there is a dramatic degradation of the recognizer accuracy. The relative spectral technique (RASTA) [24] is one of the pioneering techniques developed in this context. RASTA basically consists in a band-pass filtering applied in the

log-sub band domain, which keeps the modulation frequencies in the range between 1 and 12 Hz. The low-pass filtering helps to smooth some of the fast frame-to-frame spectral changes appearing in the spectrum due to short-term analysis. The high-pass filtering was initially designed for minimizing the influence of convolution noise (such as distortions due to microphones or fixed-telephone channels). This effect can be viewed as that of a linear system, producing a non-desired component which is additive in the log filter-bank energies domain. As the spectrum of this kind of noise varies in a different way than the speech spectrum, it can be removed efficiently by means of the RASTA technique. In fact, Hermansky and Morgan (1994) [24] showed that the reduction of this irrelevant information in the parametric representation of speech signals significantly improves the performance of the recognition system.

Temporal Patterns (TRAP) and TANDEM

TRAP processing offers much larger space for research, but often do not compute the performance of the conventional features [25]. Therefore they are generally used in combination with the standard features to derive the complimentary information of both technologies. Standard features use short term frames (about 20 to 35 ms) which is not sufficient to capture the significant discriminant information about the current phoneme. Another issue is that phonemes are not completely separated in time, but they overlap due to fluent transition of speech production organs from one configuration to other (i.e. co-articulation). This concept motivates to create features or models based on long temporal span of few hundred milliseconds.

The TRAP vectors are obtained by segmenting the speech signals into 25 ms frames having 15 ms overlap (or 10 ms shift). The spectrum of speech segment is computed by the fast Fourier transforms and filtered using the Bark scaled trapezoidal filters (i.e. a bank of critical band filters). After that log-critical band spectrum is obtained by taking the logarithm of the output of the filters. The processing so far is same as for standard features and corresponds to the first half of the processing. Then each point of the log-critical band spectra (output of one critical band filter) is appended by 25 frames of time context on both sides. Such 51 point TRAP vector covers half second (~50×10=500) of original speech signal. The mean and variance normalization can be applied to such temporal vector. This vector forms an input to as band conditioned ANN classifier, which classifies the TRAP vector to speech class by producing posterior probabilities of subword (phonemes) classes as output. Such classifier is applied in each critical band. Since these probabilities are estimated from each band, all band conditioned class probabilities are concatenated to a single vector. This vector is further processed by a negative logarithm to obtain better distribution of the parameters.

The TRAP refers to a particular way the linguistic information is extracted from the speech data. The TANDEM refers to a way of converting the frequency-localized evidence to features for the HMM-based ASR system. The TANDEM part of the technique derives a vector of posterior probabilities of sub-word speech events for every speech analysis frame from the evidence presented to its input.

B. *Back-End of ASR*

Back-end of ASR consists of two modules, the acoustic model and the language model. For the classification purpose acoustic modeling is used, and one of the widely used technique is the statistical framework hidden Markov Model (HMM). It is a doubly stochastic process, generated by two interrelated mechanisms, an underlying Markov chain having a finite number of states, and each of those states is being associated with a specific probability distribution to compute the likelihood of acoustic features. State emission probabilities can be modeled via discrete probability distributions [26], semi-continuous probability distributions [27], or continuous probability distributions [28]. Commonly used models for continuous probability distributions are mixture distributions composed of a weighted sum of Gaussian or Laplacian probability density functions.

Prior to the 1980s, ASR only used acoustic information to evaluate text hypotheses. It was then noted that incorporating knowledge about the text being spoken (exploiting textual redundancies) would significantly raise ASR accuracy. ASR may output a sequence of symbols representing phonemes, with associated likelihoods, often in the form of a lattice. A language model (LM) is applied to this lattice which may take the form of a traditional grammar, i.e., syntactic rules through which sentences are parsed into component words and phrases. However, natural language (e.g., unrestricted English) can be very complex. Typically, N-gram models estimate the likelihood of each word, given the context of the preceding N−1 words, e.g., bigram models use statistics of word pairs and trigrams models use word triplets. Unigrams are simply prior likelihoods for each word, independent of context. These probabilities are obtained through analysis of much text, and capture both syntactic and semantic redundancies in text.

As vocabulary (V words) increases for practical ASR, the size of a LM (VN) grows exponentially with V. Large lexicons lead to seriously under-trained LMs, inadequate appropriate texts for training, increased memory needs, and lack of enough computation power to search all textual possibilities. As a result, most ASR systems have employed only unigram, bigram and trigram statistics. Back-off methods fall back on lower-order statistics when higher-order N-grams do not occur in the training texts [29]. Grammar constraints are often imposed on LMs, and LMs may be refined in terms of parts-of-speech classes. LMs can be designed for specific tasks. In this section, we mainly focus on various acoustic modeling techniques.

Acoustic Modeling

Acoustic models are used to link the observed features of the speech signals with the expected phonetics of the hypothesis sentence [30]. Two most widely used approaches in acoustic modeling are:

- Hidden Markov Model (HMM)
- Artificial Neural Network (ANN)
- Gaussian Mixture Model (GMM)
- Support Vector Machine (SVM)

Hidden Markov Model

Hidden Markov models (HMMs) are stochastic models widely used in speech recognition during recent years. The number of states in classical HMMs is usually predefined and fixed during training, and may be quite different from the real number of hidden states of the signal source. The underlying assumption of the HMM is that the speech can be well characterized as a parametric random process and that the parameters of the stochastic process can be determined in a precise, well-defined manner. HMMs are the natural extension to the markov chain that produces output observation symbols in any given state [31] [32]. Therefore, the observation is a probabilistic function of the state. For a given observation sequence, the state sequence is not observable and therefore hidden. This is why the word hidden is placed before Markov models. Formally, a Hidden markov model is defined as λ (S, M, A, B, π) where

- **S**: Set of states S=S_1, S_2…..… S_n
- **M**: Number of distinct observation symbols per states
  - Individual symbols are denoted by V= {v_1, v_2……v_k}.
- **A**: a_{ij} : State transition probability
  - Each a_{ij} represents the probability of transitioning from state S_i to S_j.
  - $a_{ij} = P(T_{t+1} = S_j \,|T_t = S_i)$
- **B**: $b_j(k)$: Emission probability or observation symbol probability distribution
  - $b_j\,(k) = P(v_k \; at \; t|T_t = S_t)$
- **Π:** Initial state distribution: the probability that S_i is a start state

Given the observation sequence O = o_1, o_2……o_T and an HMM model λ = (A, B, π), we compute the probability of O given the model i.e. $(O|\lambda)$ as depicted by Fig. 6. Unfortunately, HMM suffers from some major limitations too. One major limitation of conventional HMM is that it does not provide an adequate representation of the temporal structure of speech [33]. Secondly, HMM relies on first order Markov assumption, following which the duration of each stationary segment captured by single state is inadequately modeled [33]. Finally, because of conditional independence assumption, all observation frames are dependent only on the state that generated them, and not on neighboring observation frames, which makes it hard to handle non-stationary strongly correlated frames [33].

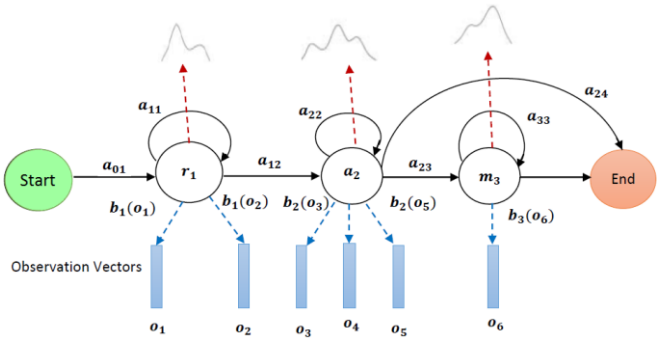

Fig. 6: Block diagram of hidden Markov model for word "ram" [33]

Three Basic Problems for HMMs

The three basic problems of interest that must be solved for the HMM model to be useful in real-world applications are [34]:

Problem 1: Evaluation problem

Given the observation sequence O={o_1,o_2…….,o_T} and the model λ =(A,B,π), how is the probability of the observation sequence given the model, computed? That is, how is (O| λ) computed efficiently? It is solved by forward backward algorithm.

Problem 2: Decoding problem

Given the observation sequence O={o_1,o_2…….,o_T} and the model λ =(A,B,π) how is a corresponding state sequence, q={q_1,q_2…….,q_T}, chosen to be optimal in some sense (i.e. best "explains" the observations). This problem can be solved by Viterbi algorithm [35].

Problem 3: Learning problem

How are the probability measures, λ= (A, B, π) adjusted to maximize (O| λ)? This problem of HMM is solved using Baum-Welch algorithm [36], which is a special case of expectation-maximization (EM) algorithm [37] and also known as forward-backward algorithm.

Artificial Neural Network

The Artificial neural network (ANN) is an information processing system, inspired by the working of biological nervous systems, i.e. brain [38]. ANN consists of large number of highly interconnected processing elements called neurons working together to solve specific problems.

Speech recognition modeling by ANNs does not require a prior knowledge of the speech process. Neural networks, such as the multilayer feed-forward networks (MLPs) or the recurrent neural networks (RNN) can be trained to associate unknown input data to learned words. To consider the temporal relationships of speech signal, time delay neural network (TDNN) and recurrent neural networks (RNN) have been proposed [39]. TDNN is a neural network approach which addresses both temporal relationships between acoustic events and invariance under translation, in time of speech. Superior speech recognition results can be achieved using TDNN approach. RNNs provide a very elegant way of dealing with (time) sequential data that embodies

correlations between neighboring data points that are close in the sequence.

Gaussian Mixture Model

Gaussian Mixture Models (GMMs) [40] are among the most statistically mature methods for clustering. A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum *A Posteriori* (MAP) estimation from a well-trained prior model.

A Gaussian mixture model is a weighted sum of M component Gaussian densities as given by the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \sum_i) \qquad (7)$$

where x is a D-dimensional continuous-valued data vector (i.e. measurement or features), w_i, i = 1,2,…..,M are the mixture weights, and $g(x|\mu_i,\sum_i)$, i = 1,2,…..,M are the component Gaussian densities with mean vector μ_i and covariance matrix $\sum_i$.

The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation (8),

$$\lambda = \{w_i, \mu_i, \sum_i\}, \quad i = 1 \dots \dots \dots M \qquad (8)$$

It is important to note that because the component Gaussian are acting together to model the overall feature density, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modeling the correlations between feature vector elements. The effect of using a set of M full covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians.

A GMM can also be viewed as a single-state HMM with a Gaussian mixture observation density, or an ergodic Gaussian observation HMM with fixed, equal transition probabilities. Assuming independent feature vectors, the observation density of feature vectors drawn from these hidden acoustic classes is a Gaussian mixture [40].

Support Vector Machine

SVM is one of the powerful tools for pattern recognition. SVMs use linear and nonlinear separating hyper-planes for data classification. Since SVMs can only classify fixed length data vectors, this method cannot be readily applied to task involving variable length data classification. The variable length data has to be transformed to fixed length vectors before SVMs can be used. It is a generalized linear classifier with maximum-margin fitting functions. This fitting function provides regularization which helps the classifier generalized better. The classifier tends to ignore many of the features.

The SVMs are effective classifiers with several outstanding characteristics [41], namely: their solution is that with maximum margin; they are capable to deal with samples of a very higher dimensionality; and their convergence to the minimum of the associated cost function is guaranteed. A Support Vector Machine (SVM) performs classification by constructing an *N*-dimensional hyper plane that optimally separates the data into two categories.

These characteristics have made SVMs very popular and successful. In the parlance of SVM literature, a predictor variable is called an *attribute*, and a transformed attribute that is used to define the hyper plane is called a *feature*. The task of choosing the most suitable representation is known as *feature selection*. A set of features that describes one case (i.e., a row of predictor values) is called a *vector*. So the goal of SVM modeling is to find the optimal hyper plane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyper plane are the *support vectors*.

Given a set of separable data, the goal is to find the optimal decision function. It can be easily seen that there is an infinite number of optimal solutions for this problem, in the sense that they can separate the training samples with zero errors. Function is used to generalize for unseen samples; the additional criterion is used to find the best solution among those with zero errors. If the probability densities of the classes, we could apply the maximum a posteriori (MAP) criterion to find the optimal solution. In most practical cases this information is not available, so it adopts other simpler criteria: among those functions without training errors, it will choose that with the maximum margin, being this margin the distance between the closest sample and the decision boundary defined by that function. Of course, optimality in the sense of maximum margin does not imply necessarily optimality in the sense of minimizing the number of errors in test, but it is a simple criterion that yields to solutions which, in practice, turn out to be the best ones for many problems [42].

REFERENCES

[1] K.H. Davis, R. Biddulph and S. Balashek, "Automatic Recognition of Spoken Digits," *J. Acoustic Society of America*, vol. 24, no. 6, pp. 637-642, 1952.

[2] J.W. Forgie and C.D. Forgie, "Results Obtained From a Vowel Recognition Computer Program," *J. Acoustic Society of America*, vol. 31, no. 11, pp. 1480-1489, 1959

[3] V.M. Velichko and N.G. Zagorukyo, "Automatic Recognition of 200 Words," *Int. J. Man-Machine Studies*, pp. 2-223, 1970.

[4] H. Sakoe and S. Chiba, "Dynamic Programming Optimization for Spoken Word Recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-26, no. 1, pp.43-49, 1978.

[5] F. Itakura, "Minimum Prediction Residual Applied to Speech Recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP -23 no. 1, pp. 67-72,1975.

[6] L.R. Rabiner, S.E. Levinson, A.E. Rosenberg and J.G. Wilpon, "Speaker Independent Recognition of Isolated Words using Clustering

Techniques," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-27, pp. 336-349, 1979.
[7] L.R.Rabiner and B.H. Juang, "Fundamentals of Speech Recognition," Prentice Hall, ISBN 0-13-015157-2, 1993.
[8] C.H. Lee, "On stochastic feature and model compensation approaches for robust speech recognition," *Speech Commun.*, vol. 25, pp. 29-47, 1998.
[9] R. Lippmann and B. Carlson, "A robust speech recognition with time-varying filtering, interruptions, and noise," *IEEE Workshop on Speech Recognition*, pp. 365-372, 1997.
[10] H. Hermansky, N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, pp. 578—589, 1994.
[11] M. Rahim, B.-H. Juang, W. Chou, E. Buhrke, "Signal conditioning techniques for robust speech recognition," *IEEE Signal Process. Letters*, vol. 3, pp. 107-109, 1996.
[12] M. A. Anusuya, and S. K. Katti, "Speech Recognition by machine: A review," *Int. J. Computer Science and Information Security*, vol. 6, no. 3, pp. 181-205, 2009.
[13] M. Pandya, "Data Driven Feature Extraction and Parameterization for Speech Recognition," M.Tech Thesis, IIT Kanpur, 2005.
[14] B. Yegnanarayana and R. N. J. Veldhuis, "Extraction of Vocal-Tract system Characteristics from Speech Signals," *IEEE Trans. of Acoustics, Speech and Signal Processing*, vol. 6 no. 4, pp. 313-327, 1998.
[15] H. Beigi, "Fundamentals *of Speaker Recognition*," Springer.
[16] J. W. Picone, "Signal Modeling Technique in Speech Recognition," *Proc. IEEE,* vol. 81, no. 9, pp. 1215-1247, 1993.
[17] L. R. Rabiner and S. E. Levinson, "Isolated and Converted Word Recognition Theory and Selected Applications," (Invited Paper) *IEEE,* 1981.
[18] B. Yegnanarayana and R. N. J. Veldhuis, "Extraction of Vocal-Tract system Characteristics from Speech Signals," *IEEE Trans. Speech and Audio Processing*, vol. 6, no. 4, pp. 313-327, July 1998.
[19] J. Hai and E. M. Joo, "Improved Linear Predictive Coding method for Speech Recognition," *Proc. joint conference Fourth International Conference on Information, communications and signal processing and multimedia*, vol. 3, pp. 1614-1613, Oct 2003.
[20] S.B. Devis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*," IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 28 (4), 1980.
[21] H. Hermansky, "Perceptually linear predictive (PLP) analysis of speech," *J. Acoustical Society of America*, vol. 87, pp. 1738–1752, Apr. 1990.
[22] A. Zolnay, R. Schluter, and H. Ney, "Acoustic Feature Combination for Robust Speech Recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, 2005.
[23] A.Revathi, R.Ganapathy and Y.Venkataramani, "Text Independent Speaker Recognition and Speaker Independent Speech Recognition Using Iterative Clustering Approach," *Int. J. Comp. Sci. and Info. Tech.*, vol. 1, no. 2, 2009.
[24] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 2, no. 4, pp. 587-589, 1994.
[25] H. Hermansky and S. Sharma, "Temporal patterns (TRAPs) in ASR of noisy speech," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1999, vol. 1, pp. 289–292.
[26] M. De Wachter *et al.*, "Template-based continuous speech recognition," *IEEE Trans. Audio, Speech, and Language Processing,* vol.15, pp. 1377–1390, 2007.
[27] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Computer Speech and Language*, vol. 3, no. 3, pp. 329–252, 1989.
[28] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.
[29] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 35, pp. 400–401, 1987.
[30] R.K. Aggarwal and M. Dave, "Acoustic Modeling Problem for Automatic speech Recognition system : Conventional methods (Part-I)," *Int. J. Speech Tech.*, Springer Verlag, vol. 14, no. 4, pp. 297-308, Dec 2011.
[31] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 77, pp. 257–286, 1989.
[32] X.D. Huang, Y. Ariki and M. Jack, "Hidden Markov Models for Speech Recognition," *Edinburgh University Press*, Edinburgh, 1990.
[33] R.K. Aggarwal and M. Dave, "Acoustic Modeling Problem for Automatic speech Recognition system : Advances and Refinements (Part-II)," *Int. J. Speech Tech.*, Springer Verlag, vol. 14, no. 4, pp. 309-320, Dec 2011.
[34] L. Rabiner, B. H. Juang, and B. Yegnarayana, "Fundamentals of Speech Recognition," *Pearson Education*, India, 2010.
[35] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, 1973.
[36] L. R. Welch, "Hidden Markov models and the Baum-Welch algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, pp. 10–13, 2003.
[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
[38] C. Bishop, "Neural Networks for Pattern Recognition," *Clarendon Press*, Oxford, 1995.
[39] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phoneme recognition using time delay neural networks," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol.37, pp. 328-339, 1989.
[40] A.R. Douglas, C.R. Richard, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, Jan. 1995.
[41] R. Solera-Ure˜na, J. Padrell-Sendra et.al, "SVMs for Automatic Speech Recognition: A Survey*," Signal Theory and Communications Department EPS-Universidad Carlos III de Madrid,Avda.,* de la Universidad, 30, 28911-Legan´es (Madrid), SPAIN.
[42] F. P´erez-Cruz and O. Bousquet, " Kernel Methods and Their Potential Use in Signal Processing," *IEEE Signal Processing Magazine*, vol. 21, no. 3, pp. 57–65, 2004.