Automatic Speech Recognition using Different Techniques

Vaibhavi Trivedi[1], Chetan Singadiya[2]

[1]Gujarat Technological University, Department of Master of Computer Engineering,
Noble Engineering College, Parth Vatika, Near Bamangam, Junagadh 362001, India

[2]Assistant Professor, Gujarat Technological University, Department of Master of Computer Engineering,
Noble Engineering College, Parth Vatika, Near Bamangam, Junagadh 362001, India

Abstract: ***Speech technology and systems in human computer communication have witnessed a stable and remarkable advancement over the last two decades. Automatic speech recognition, which was considered to be a concept of science fiction and which has been hit by number of performance degrading factors, is now an important part of information and communication technology. Speech recognition system recognizes the speech samples. There are many speech recognition systems implemented based on well known algorithms. Generally speech recognition systems have two parts the first part is feature extraction and second part is classification. There are so many algorithms for feature extraction and classification. The Mel-Frequency Cepstral Coefficients (MFCC) algorithm as the main algorithm used for the features extraction of all the set of distinct words. Implemented the Vector Quantization (VQ) algorithm for the features classification/matching and pattern recognition***.

Keywords: Speech Recognition, MFCC, VQ.

1. Introduction

Speech is the most basic, common and efficient form of communication method for people to interact with each other. People are comfortable with speech therefore persons would also like to interrelate with computers via speech, rather than using primitive interfaces such as keyboards and pointing devices. This can be accomplished by developing an Automatic Speech Recognition (ASR) system which allows a computer to identify the words that a person speaks into a microphone or telephone and translate it into written text. As a result it has the potential of being an important mode of interaction between human and computers. Physically challenged people get computer difficult to use. Partially blind people discover reading from monitor difficult. Moreover current computer interface assumes a certain level literacy from the user [1].

Speech recognition is a popular and active area of research, used to translate words spoken by humans so as to make them computer recognizable. It usually involves extraction of patterns from digitized speech samples and representing them using an appropriate data model. These patterns are subsequently compared to each other using mathematical operations to determine their contents. Even though any task that involves interfacing with a computer can potentially use ASR. The ASR system would maintain many valuable applications like dictation, command and control, embedded applications, telephone directory assistance, spoken database querying, medical applications, office dictation devices, and automatic voice translation into foreign languages etc.

It will enable even a common man to reap the benefit of information technology. There is a special need for the ASR system to be developed in their native language. Here we focus on recognition of words corresponding to English words. The main challenges of speech recognition involve modeling the variation of the same word as spoken by different speakers depending on speaking styles, accents, regional and social dialects, gender, voice patterns etc. In addition background noises and changing of signal properties over time, also pose major problems in speech recognition.

2. Application

Speech recognizer would permit more efficient communication for everybody, but especially for children, analphabets and people with disabilities. A speech recognizer could also be a subsystem in a speech-to-speech translator. Some characteristic applications of such numeral recognition are voice-recognized passwords, voice repertory dialers, automated call-type recognition, call distribution by voice commands, credit card sales validation, speech to text processing, automated data entry etc.

2.1 Speech Samples Collection (Speech Recording)

Speech samples collection is mostly concerned with recording various speech samples of each distinct word by different speakers. However, Rabiner and Juang (1993) identified four main factors that must be considered when collecting speech samples, which affect the training set vectors that are used to train the VQ codebook. Those features include who the talkers are; the speaking conditions; the transducers and transmission systems and the speech units. The first factor is the profile of the talkers/speakers. Here five different speakers out of whom their speech samples were collected. Those five speakers contain three male and two female speakers belonging to different ages, genders and races [2].

- The first factor about the profiles of talkers/speakers.
- The second factor is the speaking conditions in which the speech samples were collected from, which basically refer to the environment of the recording stage. Here speech samples collection was done in a noisy environment.

- The third factor is the transducers and transmission systems. Speech samples were recorded and collected using a normal microphone.
- The fourth factor is speech units. The main speech units are specific isolated words. E.g. English, Computer Science, Engineering, Science etc.

Parameter	*Defined Value*
Time Length (len)	2 seconds
Sampling Rate (R_fs)	16000 Hertz per second
Frame Size (N)	256
Overlap Size (M)	156
Number of Filters (nof)	40

It used a simple Matlab function for recording speech samples. However, this function requires defining certain parameters which are the sampling rate in hertz and the time length in seconds... That the time given for recording speech samples is two seconds, because it was found that two seconds are enough for recording speech samples. If the time given for recording was more than two seconds that would result in having so much silence time in the recorded speech sample or the word's utterance. Speech samples have been recorded and collected in order to be used. The collected speech samples are then going to pass through the features extraction, features training and features testing stages.

3. Features Extraction Using MFCC Algorithm

The main objective of features extraction is to extract characteristics from the speech signal that are unique to each word which will be used to differentiate between a wide set of distinct words. The frequently used features for speech processing, also known as the Mel- Frequency Cepstral Coefficients (MFCC), are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies that have been used to capture the phonetically important characteristics of speech. The mel-frequency cepstral coefficients (MFCCs), which are obtained by first performing a standard Fourier analysis, and then converting the power- spectrum to a mel frequency spectrum. By taking the logarithm of that spectrum and by computing its inverse Fourier transform one then obtains the MFCC. The individual features of the MFCC seem to be just weakly correlated, which turns out to be an advantage for the creation of statistical acoustic models.

MFCC has generally obtained a better accuracy and a minor computational complexity with respect to alternative processing as compared to other features extraction techniques.

3.1 MFCC Parameters Definition

Mel-Frequency Cepstral Coefficient (MFCC) is a deconvolution algorithm applied in order to obtain the vocal tract impulse response from the speech signal. It transforms the speech signal which is the convolution between glottal pulse and the vocal tract impulse response into a sum of two components known as the cepstrum. This computation is carried out by taking the inverse DFT of the logarithm of the magnitude spectrum of the speech frame

x^[n] = IDFT { log (DFT{ h[n]*u[n] }) }
= h^[n] + u^[n] (1)

Where h^[n], u^[n] and x^[n] are the complex cepstrum of h[n], u[n] and x[n] respectively.

Form the above equation the convolution of the two components is changed to multiplication when Fourier transform is performed. Then by taking the logarithm, the multiplication is changed to addition. This is basically how the complex cepstrum x^[n] is obtained.

MFCC includes certain steps applied on an input speech signal. Those computational steps of MFCC include; preprocessing, framing, windowing, Discrete Fourier Transform (DFT), Mel Filter bank, Logarithm, and finally computing the inverse of DFT.

a. Preprocessing

According to Gordon (1998), preprocessing is considered as the first step of speech signal processing, which involves the conversion of analog speech signal into a digital form. It is a very crucial step for enabling further processing. Here the continuous time signal (speech) is sampled at discrete time points to form a sample data signal representing the continuous time signal. Then samples are quantized to produce a digital signal. The method of obtaining a discrete time representation of a continuous time signal through periodic sampling, where a sequence of samples, x[n] is obtained from a continuous time signal.
x(t), stated clearly in the relationship,

$$x[n] = x(nT) \quad (2)$$

Where T is the sampling period and "1/T = fs" is the sampling frequency, in samples/second, and n is the number of samples. It is apparent that more signal data will be obtained if the samples are taken closer together through making the value of T smaller.

The size of the sample for a digital signal is determined by the sampling frequency and the length of the speech signal in seconds. For example if a speech signal is recorded for 2 seconds using sampling frequency of 10000 Hz, the number of samples = 10000 x 2s = 20000 samples.

Here the speech sample is 16000 Hertz for 2 seconds of time length. The preprocessing works as to obtain an array from the microphone after recording, calculates the time graph and spectrum of the speech signal and displays both the time graph as well as the spectrum in a figure plot format .below Figure 1 displays the obtained results of the time graph for the recorded word "English".

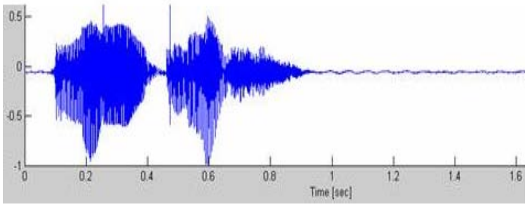

Figure 1: Plot of the Time Graph for the Recorded Word "English"

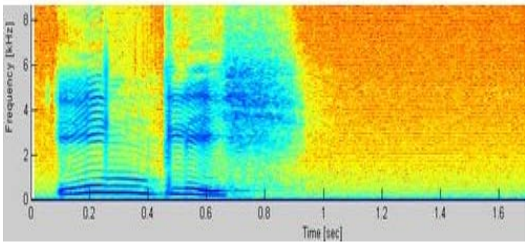

Figure 2: Plot of the Spectrum for the Recorded Word "English"

b. Framing

Framing is the process of segmenting the speech samples obtained from the analog to digital (A/D) conversion into small frames with time length in the range of (20 to 40) milliseconds. Speech signal is known to exhibit quasi-stationary behavior in a short period of time (20 – 40) milliseconds. Therefore, framing enables the non-stationary speech signal. To be segmented into quasi stationary frames, and enables Fourier transformation of the speech signal. The rationale behind enabling the Fourier transformation of the speech signal is because a single Fourier transforms of the entire speech signal cannot capture the time varying frequency content due to the non stationary behavior of the speech signal. Therefore, Fourier transform is performed on each segment separately.

If the frame length is not too long (20 – 40) milliseconds, the properties of the signal will not change appreciably from the beginning of the segment to the end. Thus, the DFT of a windowed speech segment should display the frequency – domain properties of the signal at the time corresponding to the window location. (Alan and Ronald, 1999).also said that if the frame length is long enough so that the harmonics are resolved (>80) milliseconds, the DFT of a windowed segment of voiced speech should show a series of peaks at integer multiples of the fundamental frequency of the signal in that interval. This would normally require that the window span several periods of the waveform. Whereas, if the frame is too short (<10) milliseconds, then the harmonics will not be resolved,

The general spectral shape will still be evident. This is a typical tradeoff between frequency resolution and time resolution that is required in the analysis of non stationary signals. In addition, each frame overlaps its previous frame by a predefined size. The goal of the overlapping scheme is to smooth the transition from frame to frame. Framing is meant to frame the speech samples into segments small enough so that the speech segment shows quasi-stationary behavior. The length of each segment is 256 samples which is equivalent to

$$[((256 / 16000) * 1000)] = 16 \text{ milliseconds.}$$

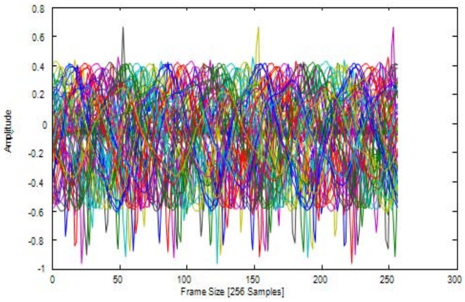

Figure 3: Segmented Speech Signal (Frame Size = 256 samples)

c. Windowing

This processing step is meant to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as *w(n), 0 _ n _ N -1*, where *N* is the number of samples in each frame, then the result of windowing is the signal as shown in (3).

$$y(n) = x(n) * w(n), 0 \le n \le N \quad (3)$$

According to Alan and Ronald (1999) and Thomas (2002), windowing is very necessary to work with short term or frames of the speech signal in order to select a portion of the speech signal that can be reasonably assumed to be stationary speech signal. It is performed in order to avoid any unnatural discontinuities in the speech segment and distortion in the underlying spectrum, in order to ensure that all parts of the speech signal are recovered and possible gaps between frames are eliminated.

Becchetti and Ricotti (1999) mentioned that hamming window is the most commonly used window shape in speech recognition technology, because a high resolution is not required, considering that the next block in the feature extraction processing chain integrates all the closest frequency lines. Hamming window, whose impulse response is a raised cosine impulse has the
form (4):

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), 0 \le n \le N-1 \\ \text{Otherwise} \end{cases} \quad (4)$$

The effect of windowing on the speech segment in Figure 4 can be seen clearly in Figure there seems to be a smooth transition towards the edges of the frame.

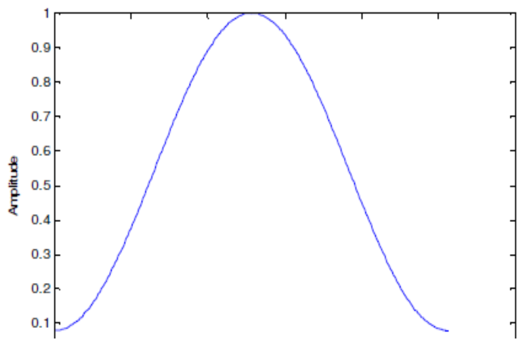

Figure 4: Hamming window

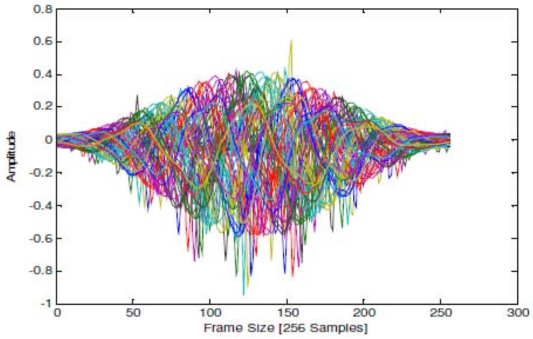

Figure 5: Windowed Speech Segment

d. Discrete Fourier Transform (DFT)

Owens (1993) stated that the discrete Fourier transform (DFT) is normally computed via the fast Fourier transform (FFT) algorithm, which is a widely used technique for evaluating the frequency spectrum of speech. FFT converts each frame of *N* samples from the time domain into the frequency domain. The FFT is a fast algorithm, which exploits the inherent redundancy in the DFT and reduces the number of calculations. FFT provides exactly the same result as the direct calculation.

According to Alexander and Sadiku (2000), Fourier series enable a periodic function to be represented as a sum of sinusoids and convert a speech signal from the time domain to the frequency domain. The same analysis can be carried out on non periodic functions using Fourier transform. Therefore, Fourier transform is used due to the non periodic behavior of the speech signal. Alexander and Sadiku (2000) also added that the basis of performing Fourier transform is to convert the convolution of the glottal pulse u[n] and the vocal tract impulse response h[n] in the time domain into multiplication in the frequency domain. This can be supported by the convolution theorem (5).

If X(w), H(w) and Y(w) are the Fourier transforms of x(t), h(t) and y(t) respectively, then: Y(w) = FT [h(t) * x(t)] = H(w) x X(w)

In analyzing speech signals, Discrete Fourier Transform (DFT) is used instead of Fourier transform, because the speech signal is in the form of discrete number of samples due to preprocessing. The discrete Fourier transform is represented by the equation (6), Where X(k) is the Fourier transform of x(n).

$$X(k+1) = \sum_{n=0}^{N-1} x(n+1) W_N^{kn}$$

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)} \qquad \text{Where} \qquad \textbf{(6)}$$

e. Mel Filterbank

The information accepted by low frequency components of the speech signal is more important than the high frequency components. In order to place more emphasize on the low frequency components, Mel scaling is applied. According to Thomas (2002), Mel scale is a unit of special measure or scale of perceived pitch of a tone. It does not correspond linearly to the normal frequency, but behaves linearly below 1 kHz and logarithmically above 1 kHz. This is based on studies of the human perception of the frequency content of sound. Therefore we can use the formula (7) in order to compute the Mels for a given frequency *f* in Hz. This formula also shows the relationship between both the frequency in hertz and Mel scaled frequency.

Frequency (Mel Scaled) = [2595log (1+f (Hz)/700] (7)

In order to implement the filterbanks, the magnitude coefficient of each Fourier transformed speech segment is binned by correlating them with each triangular filter in the filterbank. In order to perform Mel-scaling, a number of triangular filters or filterbanks are used. **Figure 6** shows the configuration of filters.

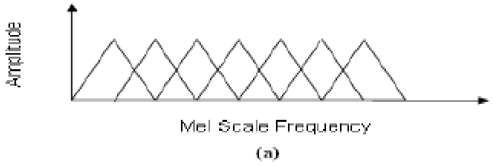

Figure 6: Filterbank in Mel Scale Frequency and

f. Logarithm

The logarithm has the effect of changing multiplication into addition. Therefore, this step simply converts the multiplication of the magnitude of the Fourier transform into addition. referred to as signal's logarithm Mel spectrum .the logarithm of the Mel filtered speech segment is carried out using the Matlab command "log", which returns the natural logarithm of the elements of the Mel filtered speech segment. According to Becchetti and Ricotti (1999), this step is meant for computing the logarithm of the magnitude of the coefficients, because of the logarithm algebraic property which brings back the logarithm of a power to a multiplication by a scaling factor.

Becchetti and Ricotti (1999) also added that the magnitude and logarithm processing is performed by the ear as well, whereby the magnitude discards the useless phase information while a logarithm performs a dynamic compression in order to make the feature extraction process less sensitive to variations in dynamics. The result obtained after this step is often referred to as signal's logarithm Mel spectrum. process less sensitive to variations in dynamics.

The result obtained after this step is often

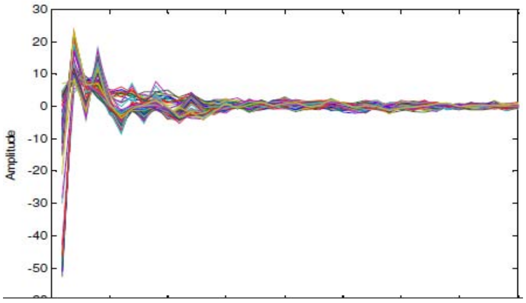

Figure 7: Result of the Matlab Logarithm Command

g. Inverse of Discrete Fourier Transform (IDFT)

According to Becchetti and Ricotti (1999), the final procedure for the Mel frequency cepstral coefficients (MFCC) computation consists of performing the inverse of DFT on the logarithm of the magnitude of the Mel filter bank output. The speech signal is represented as a convolution between slowly varying vocal tract impulse response and quickly varying glottal pulse. Therefore, by taking the inverse of DFT of the logarithm of the magnitude spectrum, the glottal pulse and the impulse response can be separated. The result obtained after this step is often referred to as signal's Mel cepstrum. this was the final step of computing the MFCCs. It required computing the inverse Fourier transform of the logarithm of the magnitude spectrum in order to obtain the Mel frequency cepstrum coefficients.

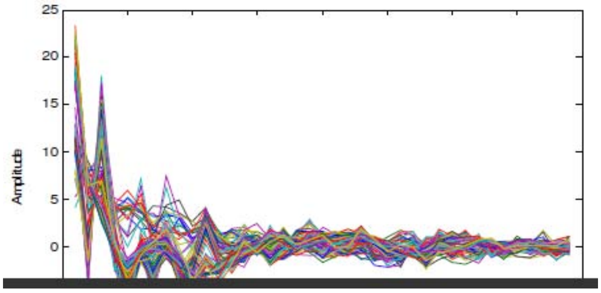

Figure 8: Mel Frequency Cepstrum Coefficients (MFCCs)

The MFCCs at this stage are ready to be formed in a vector format known as features vector. This features vector is then considered as an input for the next section, which is concerned with training the feature vectors that are randomly chosen for forming the VQ codebook. Each features vector has the vector size of [1 * 3237].

4. Features Classification using Vector Quantization (VQ) Algorithm

The features extraction process using MFCC whereby isolated-word discriminative features are extracted from the speech signal. the classification or clustering method known as vector quantization .This method is part of the decision making process of determining a word based on previously stored information, and it uses the features vectors extracted from speech signals using MFCC as the inputs for this algorithm.

This step is basically divided into two parts, namely features training and features matching/testing. Feature training is a process of enrolling or registering a new speech sample of a distinct word to the identification system database by constructing a model of the word based on the features extracted from the word's speech samples. Features' training is mainly concerned with randomly selecting feature vectors of the recorded speech samples and perform training for the codebook using the LBG vector quantization (VQ) algorithm. On the other hand, a feature matching/testing is a process of computing a matching score, which is the measure of similarity of the features extracted from the unknown word and the stored word models in the database. The unknown word is identified by having the minimum matching score in the database.

4.1 Features Training Using Vector Quantization (VQ) Algorithm

Vector quantization (VQ) is the process of taking a large set of feature vectors and producing a smaller set of feature vectors that represent the centroids of the distribution, i.e. points spaced so as to minimize the average distance to every other point. Vector quantization has been used since it would be impractical to store every single feature that is generated from the speech utterance through MFCC algorithm.

The training process of the VQ codebook applies an important algorithm known as the LBG VQ algorithm, which is used for clustering a set of *L* training vectors into a set of *M* codebook vectors. This algorithm is formally implemented by the following recursive procedure: (Linde et al., 1980). The following steps are required for the training of the VQ codebook using the LBG algorithm as described by Rabiner and Juang (1993).

1. Design a 1-vector *codebook;* this is the *centroid* of the entire set of training vectors. Therefore, no iteration is required in this step.

2. Double the size of the *codebook* by splitting each current *codebook* yn according to the following rule (8):

$$y_n^+ = y_n(1+\varepsilon)$$
$$y_n^- = y_n(1-\varepsilon) \tag{8}$$

Where ***n*** varies from 1 to the current size of the *codebook* , and is the splitting parameter, whereby is usually in the range of $(0.01 \le \varepsilon \le 0.05)$.

The initial codebook is obtained by combining all the selected feature vectors for each distinct word in one database. The purpose of this initial codebook is to serve as a starting codebook for training each selected feature vector against one another. The initial codebook is referred to as the variable "CODE" in the LBG VQ Matlab function.

3. Nearest-Neighbor Search: for each training vector, find the *codeword* in the current *codebook* that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest *centroid*). This is done using the K-means iterative algorithm.

4. Centroid Update: update the *centroid* in each cell using the *centroid* of the training vectors assigned to that cell. the centroid updates requires updating the codebook too, by taking the average of the speech vector in a cell to find the new value of the codevector.

Figure shows a flow diagram of the detailed steps of the LBG algorithm. "*Cluster vectors*" is the nearest-neighbor search procedure which assigns each training vector to a cluster associated with the closest codeword. "*Find centroids*" is the centroid update procedure. "*Compute D (distortion)*" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

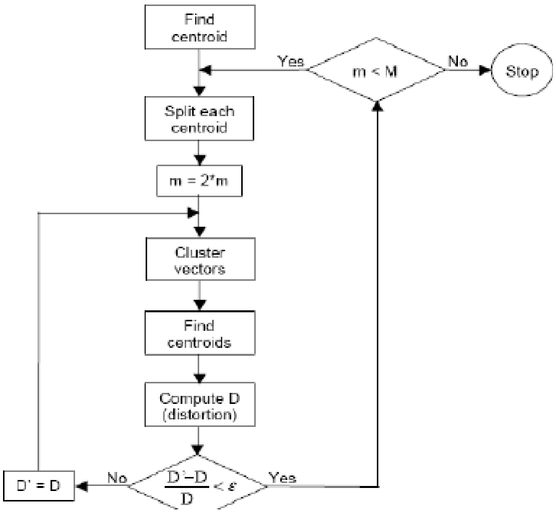

Figure 9: Flow Diagram of the LBG Algorithm

4.2 Features Matching/Testing Using Euclidean Distance Measure

Euclidean distance measure is applied in order to measure the similarity or the dissimilarity between two spoken words, which take place after quantizing a spoken word into its codebook. The matching of an unknown word is performed by measuring the Euclidean distance between the features vector of the unknown word to the model (codebook) of the known words in the database. The goal is to find the codebook that has the minimum distance measurement in order to identify the unknown word.

For example in the testing or identification session, the Euclidean distance between the features vector and codebook for each spoken word is calculated and the word with the smallest average minimum distance is picked as shown in the equation below

$$d(x,y) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2} \tag{9}$$

Where xi is the ith input features vector, y_i is the ith features vector in the codebook, and d is the distance between xi and yi. a simple Euclidean distance measure is applied on an unknown features vector compared against the trained codebook.

Therefore, there must be an unknown speech signal and a trained codebook as inputs to this algorithm in order to measure their distance and test the entire performance. The outputs of this algorithm are the ID numbers assigned for each features vector in the trained codebook as well as the distances or the squared error values. However, this algorithm picks up the ID number of the features vector which has the minimum distance to the unknown features vector. The most important purpose of performing this stage is to measure the accuracy/recognition in order to measure the validity of the algorithms used in this application.

5. Summary

Here presented a detailed technical overview of MFCC and VQ, and how those two algorithms relate to each other. It was clearly mentioned that MFCC handles the features extraction process, which then produces outputs of speech feature vectors that are then considered as the training set used in the VQ algorithm to train the VQ codebook. Therefore, VQ works as a classification or pattern recognition technique that classifies different speech signals according to the classes. LBG VQ is the most commonly used VQ algorithm, which is divided into two phases. The first phase is the training, whereby randomly selected speech signals form a training set of samples that are used as an initial codebook for training the VQ codebook. The second phase is the matching/testing that uses the Euclidean distance measure for comparing an unknown speech signal against the VQ codebook, which then selects the codeword in the codebook with the minimum distance.

The combination of MFCC and VQ has been widely used in speaker recognition. Thus, this research studies the possibility of using this combination in telephony speech recognition systems.

References

[1] M. A. M. Abu Shariah, R. N. Ainon, R. Zainuddin, and O. O. Khalifa, "Human Computer Interaction Using Isolated-Words Speech Recognition Technology," IEEE Proceedings of The International Conference on Intelligent and Advanced Systems (ICIAS'07), Kuala Lumpur, Malaysia, pp. 1173 – 1178, 2007.

[2] DOUGLAS O'SHAUGHNESSY, "Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis", Proceedings of the IEEE, VOL. 91, NO. 9, September 2003, 0018-9219/03$17.00 © 2003 IEEE

Author Profile

Vaibhavi Trivedi received her BE (Computer Engineering in 2007 and M. E. (Computer Engineering-pursing) in 2011-2013.Currently she is researcher student of Noble Engineering college from Gujarat Technological University ,Gujarat-INDIA. Her research areas are Speech Recognition and Artificial Intelligence.

Chetan Singadiya is working as Assistant Professor in Department of Master of Computer Engineering, Noble Engineering College, Gujarat Technological University