Volume 2, Issue 3, March 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Preprocessing In ASR for Computer Machine Interaction with Humans: A Review

Bhupinder Singh, Vanita Rani, Namisha Mahajan
Dept. of Computer Sc. & Engg., IGCE Abhipur, Mohali (Pb.), India

Abstract:-Automatic speech recognition (ASR) is a developing field and large numbers of research followers are showing their interest to develop a perfect automatic speech recognition system. Lot of work is being done in ASR and DSP (Digital Speech Processing). Today human is able to interact with computer hardware and related types of machines in human language. Research followers are trying to develop a perfect ASR system because we have all these advancements in ASR and research in digital signal processing but computer machines are unable to match the performance of their human utterances in terms of accuracy of matching and speed of response. So in this paper the study of preprocessing is focused area of speech recognition process is to build a perfect speaker independent ASR system. The reasons behind that it's vast number of applications, and drawbacks of available techniques of automatic speech recognition. We will discuss one of the process of speech recognition namely Preprocessing. Commonly used Background Noise Removal, End Point Detection, Pre-emphasis, Framing in Preprocessing are discussed in detail.

Keyword: Automatic speech recognition (ASR), Digital Speech Processing (DSP).

Preprocessing

In speech recognition first phase is preprocessing which deals with a speech signal which is an analog signal at the recording time, which varies with time. To process the signal by digital means, it is necessary to sample the continuous-time signal into a discrete-time discrete-valued (digital) signal. The properties of a signal change relatively slowly with time, so that the speech can be divided into a sequence of uncorrelated segments or frames and process the sequence as if each frame has fix properties. Under this assumption, one can extract the features of each frame based on the sample inside the frame only. And usually, the feature vector will replace the original signal in the further processing, which means the speech signal is converted from a time varying analog signal into a sequence of feature vectors. The process of converting sequences of speech samples to feature vectors representing events in the probability space is called Signal Modeling [1].

The purpose of preprocessing is to derive a set of parameters to represent speech signals in a form which is convenient for subsequent processing and to process the sampled speech signal and produce representation which is independent a amplitude variations, speaker stress and noise which introduced from the transmission medium. Both time domain and frequency domain approaches can be used. Time domain approaches, such as parameters of energy and zero crossing rate, directly dealing with the waveform of the speech signal, are usually simple to implement. Frequency domain approaches involve some form of spectral analysis that is not directly evident in the time domain. The latter approaches are more widely used in speech recognition.

(a) Background Noise Removal: Background noise is usually produced by air conditioning system, fans, fluorescent lamps, type writers, computer systems, back conversation, footsteps, traffic, opening and closing the doors etc. the designers of speech recognition system usually have litter control over these things in the real life environments. Type of noise additive in nature and usually steady state except for impulse noise sources like type writers [2]. Depending on the environment, the noise levels will vary from about 60 dB to 90 dB. The most commonly used technique to minimize the effect the background noise is to use a head mounted close speaking microphone. When a speaker is producing speech at normal conversational level, the average speech level increase by about 3dB each time when the microphone is filtering the speech signal. The filter used to remove the background noise is as follows:

$$E_s = 10 \times \log_{10}\left\{ \in + \frac{1}{N} \sum_{n=1}^{N} S^2(n) \right\}$$

Where, the *Es* is log energy of a block of N samples and $\in$ is a small positive constant added to prevent the computing of log zero. S(n) be the n^{th} speech sample in the block N samples.

(b) Speech Word detection (End Point Detection): In speech recognition, there is need to process the utterance consisting of speech, silence and other background noise. The detection of the presence of speech embedded in the various types of non-speech events and background noise is called an end point detection, speech detection or speech activity detection. A good end point detection algorithm affects and performance of system in the terms of accuracy and speech for several reasons. First is, the silence frame can be removed prior to recognition, the accumulated utterance likelihood score will focus more on the speech portion of an utterance instead of on both noise and speech [3]. Second, it is hard to model noise and silence accurately in changing environments [4]. This effect can be limited by background noise frame in advance. Third, removing non-speech frames when the number of non speech frames is large can significantly reduce the computation time [5] [6].
Steps for speech word detection are as follows:
(i) Measurements for endpoint detection: Measurements for this is as given below:

- Zero crossing count, NZ, the number of zero crossing the block.
- The log energy E*s* of a block of length N samples is defined as:

$$E_s = 10 \times \log_{10}\left\{ \in + \frac{1}{N}\sum_{n=1}^{N} S^2(n) \right\}$$

Where, $\in$ a small positive is constant added to prevent the computing of log zero. Obviously, $\in$ << mean-squared value of the speech samples.

- Normalization auto-correlation coefficient at unit sample delay C_1 is defined as:

$$C_1 = \frac{\sum_{n=1}^{N} S(n)s(n-1)}{\sqrt{\left[\sum_{n=1}^{N} S^2(n)\right]\left[\sum_{n=0}^{N-1} S^2(n)\right]}}$$

(ii) Filter for end point detection: It is assumed that one utterance may have several speech segments separated by possible pauses. Each of the segments can be determined by detecting a pair of endpoints named segment beginning and ending points. On the energy contours of utterance, there is always a raising edge following a beginning point and a descending edge preceding an ending point. These are known as beginning and ending edges as shown in figure 1.

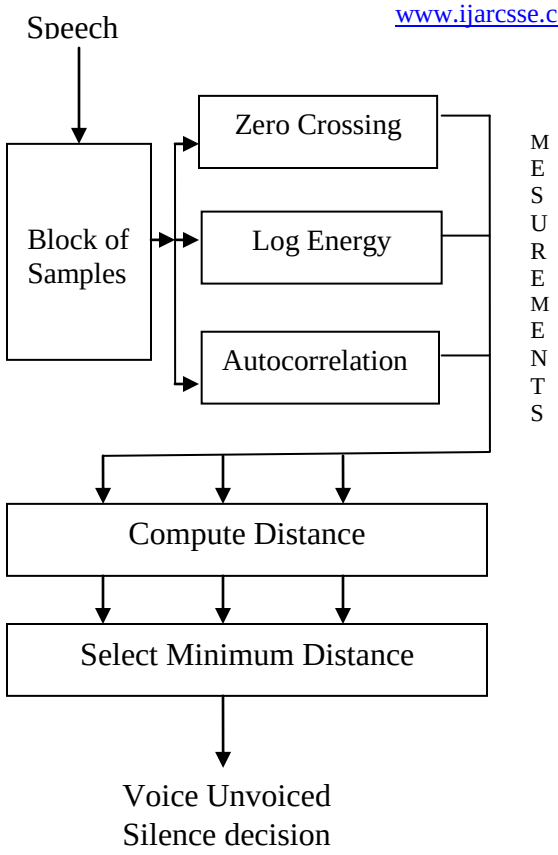

Figure 1: Block Diagram of End Point Detection

The approach is first to detect the edge and then to find the corresponding endpoints. Thus for accurate and robust endpoint detection, there is need of a detector that can detect all possible endpoints from energy feature. Once dimensional short term energy in the data sample, to be the feature for endpoint detection. It is given as:

$$E(l) = 10 \times \log 10 \sum_{j=n(l)}^{n(l)+l-1} o(j)$$

Where, *o(j)* is data sample, *l* is frame number, *'l'* is window length, E(*l*) is frame energy in decibel, n(*l*) is number of first data sample in the window.
(iii) Energy normalization: The purpose of normalization of energy is to normalize the utterance energy E_l. The normalization of energy is performed by finding the maximum energy value E_{max} over the words as:
$E_{max} = \max (E_l)$, $1 \leq l \leq L$
By subtracting E_{max} from E_l to give
$\hat{E}_l = E_l - E$
In this way the peak energy value of each word is zero decibel and the recognition system is relatively insensitive to the difference in gain between different recordings. In performing the above calculations, there is a constraints that word energy contour normalization cannot take place until the end of the word is located.

(c) Pre-emphasis: It is aimed to compensate for lip radiation and inherent attenuation of high frequencies in the sampling process. High frequency components are

emphasized and low frequency components are attenuated. This is quire a standard preprocessing step. Typically, the speech signal produced by human being has a spectral slope of approximately-6dB/octave for voiced sounds. This slope is because of two reasons: (a) the shape of the glottal pulse introduces a slope of -12dB/octave and (b) The lip radiation introduces a slope of +dB/octave. Therefore, the resultant slope of approximately -6dB/octave exists in the recorded voiced speech sounds. Pre-emphasis is performed to remove this slope of -6 dB/octave. To accomplish the task, the speech signal is passed through a high-pass finite impulse response (FIR) filter of order 1. The pre-emphasis is defined by:

$y[n] = s[n] - P \times s[n-1]$

Where, s[*n*] is the nth speech sample, y[*n*] is the corresponding pre-emphasized sample and P is the pre-emphasis factor typically having a value between 0:9 and 1. Pre-emphasis ensures that in the frequency domain all the formats of the speech signal have similar amplitude so that they get equal importance in subsequent processing stages [5]. In the frequency domain, it looks like:

$H(z) = 1 - az^{-1}$

(d) Buffering (Framing (Frame Blocking)): Speech is a quasi-stationary signal and is stationary only for a short interval of time. This allows us to use block processing techniques such as Discrete Fourier Transform (DFT) to analyze speech signal. In this step the pre-emphasized speech signal, $s_o(n)$, is blocked into frame buffers of N samples with an adjacent frame separated by M sample. Speech is typically analyzed in overlapping short frames of about 20 msec long with a 10 msec frame shift. Overlapping ensures the smooth transition of estimated parameters from frame to frame. If the sampling frequency is 8 kHz, then *N = 240 and M = 80, Let x, (m)* represent the i^{th} short-time signal buffer, then.

$x_i(m) = s(m+iM)\ m=0,\ldots\ldots,N-1$

(e) Windowing: The next step in the process is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The window *w(n)*, determines the portion of the speech signal that is to be processed by zeroing out the signal outside the region of interest [1]. If the window is defined as *w(m)*, then the result of windowing is the signal $x_i(m)$:

$x_i(m) = x_i(m)w(m) \qquad m=0,\ldots\ldots,N-1$

The choice of window *w(n)* is a grade-off between several factors:

- The window shape may reduce distortion, but it may increase signal shapre alteration.
- The length is proportional to the frequency resolution and inversely proportional to the time resolution.
- The overlap is proportional to the frame rate, but it is also proportional to the correlation of subsequent frames.

Types of windows for speech are given below:

(i) Rectangular window: w(n)

$$= \begin{cases} 1 & ;\leq n \leq N-1 \\ 0 & ;otherwise \end{cases}$$

(ii) Bartlett window: w(n)

$$= \begin{cases} \dfrac{2n}{N-1} & ;\ 0 \leq n \leq \dfrac{N-1}{2} \\ 2 - \dfrac{2n}{N-1} & ;\ \dfrac{N-1}{2} \leq n \leq N-1 \\ 0 & ;\ otherwise \end{cases}$$

(iii) Hamming window: w(n) =

$$\begin{cases} 0.54 - 0.46\cos(2\pi m \mid (N-1)),\ 0 \leq n \leq N \\ 0, \qquad otherwise \end{cases}$$

(iv) Hanning window: w(n) =

$$\begin{cases} 0.5 - 0.5\cos(2\pi \dfrac{n}{N-1}) & ;\ 0 \leq n \leq N-1 \\ 0, \qquad otherwise \end{cases}$$

Conclusion

The study of preprocessing has been carried out to develop a voice based user machine interface system. This system can be used in various applications related with disable persons those are unable to operate computer through keyboard and mouse, these type of persons can use computer with the use of Automatic Speech Recognition system, with this system user can operate computer with voice commands so extra advantages of computer machine interface with human will be that if any disable person is using this system he/she feel that they are working in real environment as what they want to do. Second application for those computer users which are not comfortable with English language and feel good to work with their native language i.e. English, Punjabi, Hindi.

References

1. Picone, L. (1993), "*Signal modeling technique in Speech Recognition",* IEEE ASSP Magazine, Vol. 81, Issue 9, pp. 1215-1247.
2. Hwang, T. and Chang, S. (2004), "*Energy Contour enhancement for noisy speech recognition*", International Symposium on Chinese Spoken Language Processing, Vol. 1, pp. 249-252.
3. Rabiner, L. and Sambur, M. (1976), "*Some Preliminary experiments in the recognition of connected digits*", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 24, Issue 2, pp. 170-182.
4. Abdulla, W. (2002), "*HMM – based techniques for speech segment extraction*", Scientific programming, IOS Press, Amesterdam, The Netherlands, Vol. 10, Issue 3, pp. 221–239.
5. Becchetti, C. and Ricotti, L. (2004), "*Speech Recognition Theory and C++ Implementation",* John Wiley & Sons, Wiley Student Edition, Singapure, pp. 121-188.
6. Ney, H. (2003), "*An optimization algorithm for determining the end points of isolated utterances*", in proceedings of IEEE International Conference on Acoustics, Speech, and Siganl Processing (ICASSP), Vol. 7, Issue 3, pp. 26-41.
7. Singh B. and Singh P.(2011) "Voice Based user Machine Interface for Punjabi using Hidden Markov Model", in the proceeding of IJCST Vol. 2, Issue 3, pp. 222-224.

