Recognition of Vernacular Language Speech for Discrete Words using Linear Predictive Coding Technique

Omesh Wadhwani, **Amit Kolhe, Sanjay Dekate**

Abstract*—Vernacular language spoken in various countries creates a limitation on software associated with speech recognition. This paper is an attempt to overcome such problem. The suggested work makes use of Linear Predictive Technique for better interpretation of spoken words. The rule based structure of fuzzy suits very well with closeness of vernacular speech recognition. In this paper we study the feasibility of Speech Recognition with fuzzy neural Networks for discrete Words Different Technical methods are used for speech recognition. Most of these methods are based on transfiguration of the speech signals for phonemes and syllables of the words. We use the expression "word Recognition" (because in our proposed method there is no need to catch the phonemes of words.). In our proposed method, LPC coefficients for discrete spoken words are used for compaction and learning the data and then the output is sent to a fuzzy system and an expert system for classifying the conclusion. The experimental results show good precisions. The recognition precision of our proposed method with fuzzy conclusion is around 90 percent.

***Index Terms*— Automatic Speech Recognition, Feature Extraction, Linear Predictive Coding, LPC Coefficients, Vernacular, Words Recognition, Word error rate.**

I. INTRODUCTION

The vernacular speech of a particular community is the ordinary speech used by people in a particular community that is noticeably different from the standard form of the language. Especially where Europeans languages were concerned, the linguists of the past normally concentrated on the standard forms of languages. Nonstandard vernacular forms were silently ignored, excepting only in the study of regional dialects, for which the speech of elderly rural speakers was considered most appropriate; at the same time, the speech of younger speakers or of urban speakers was similarly ignored. Interest in vernacular forms developed only slowly during the twentieth century [1], but it became increasingly prominent with the rise of sociolinguistics in the 1960s. Today, there is intense interest in vernacular forms of the speech.

Manuscript Received October 09, 2011.

Omesh Wadhwani, M-Tech Student, Department of Electronics and Telecommunication, Chhattisgarh State University (CSVTU), Rungta College of Engineering and Technology, (e-mail: omesh.wadh@gamil.com).

Prof. Sanjay Dekate, Department of Electronics and Telecommunication, Chhattisgarh State University (CSVTU), Rungta College of Engineering and Technology.

Prof. Amit Kolhe, Department of Electronics and Telecommunication, Chhattisgarh State University (CSVTU), Rungta College of Engineering and Technology.

II. AUTOMATIC SPEECH RECOGNITION

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to text. The term "voice recognition" is sometimes used to refer to recognition systems that must be trained to a particular speaker as is the case for most desktop recognition software. Recognizing the speaker can simplify the task of translating speech.

Speech recognition is a broader solution that refers to technology that can recognize speech without being targeted at single speaker—such as a call system that can recognize arbitrary voices.

The goal of automatic speech recognition (ASR) is to take a word from microphone as input, and produce the text of the words spoken. The need for highly reliable ASR lies at the core of many rapidly growing application areas such as speech interfaces (increasingly on mobile devices) and indexing of audio/video databases for search. While the ASR problem has been studied extensively for over fifty years, it is far from solved. There has been much progress and ASR technology is now in widespread use; however, there is still a considerable gap between human and machine performance, particularly in adverse conditions.

The performance of any speech recognition system can be improved by choosing proper symbols for representation. Characters of the language are chosen as symbols for the signal-to-symbol transformation module of our speech-to-text system being developed for the Indian language Hindi. The aim here is to emulate human processes as much as possible at the signal-to symbol transformation stage itself. In this case, the expert systems approach permits a clear distinction between the domain knowledge and the control structure needed to manipulate the knowledge. A number of speech recognition systems for continuous speech have been with varied success. The main drawback in these systems is that they use a simple approach for signal-to-symbol transformation with some abstract units as symbols, thereby increasing the complexity at higher levels of processing. Recent efforts [2, 5] try to improve the performance of signal-to-symbol transformation using speech specific knowledge.

Speech recognition refers to the ability to listen (input in audio format) spoken words and identify various sounds present in it, and recognize them as words of some known language.

Speech recognition in computer system domain may then be defined as the ability of computer systems to accept spoken words in audio format - such as wav or raw - and then generate its content in text format.
Speech recognition in computer domain involves various steps with issues attached with them. The steps required to make computers perform speech recognition are: Voice recording, word boundary detection, feature extraction, and recognition with the help of knowledge models.

III. APPROACH

A. Step One:

Sound Recording and Word detection component is responsible for taking input from microphone and identifying the presence of words. Word detection is done using energy and zero crossing rate of the signal.

B. Step Two:

Feature Extraction component generated feature vectors for the sound signals given to it. It generates Mel Frequency Cepstrum Coefficients and Normalized energy as the features that should be used to uniquely identify the given sound signal.

C. Step Three:

Recognition component is the most important component of the system and is responsible for finding the best match in the knowledge base, for the incoming feature vectors.

D. Step Four:

Knowledge Model Component consists of Word based Acoustic. Acoustic Model has a representation of how a word sounds.

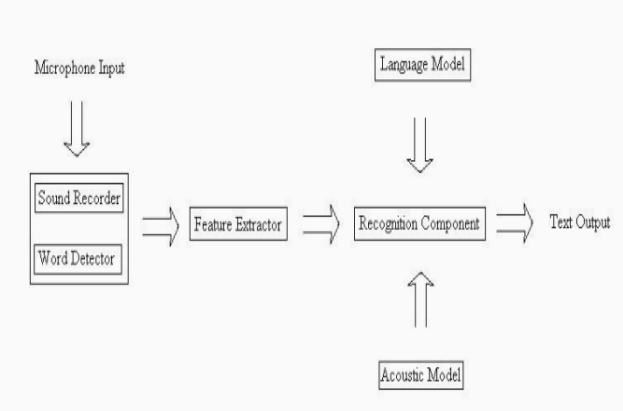

Fig.1. Block Diagram of Speech Recognition

IV. LINEAR PREDICTIVE CODING (LPC)

Linear predictive coding (LPC) is a tool used mostly in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a linear predictive model. It is one of the most powerful speech analysis techniques, and one of the most useful methods for encoding good quality speech at a low bit rate and provides extremely accurate estimates of speech parameters.

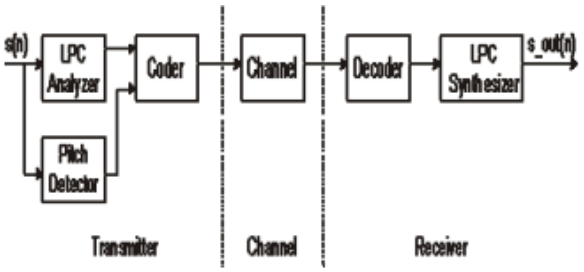

Fig.2. Block Diagram of an LPC Vocoder

It has two key components: analysis or encoding and synthesis or decoding [2]. The analysis part of LPC involves examining the speech signal and breaking it down into segments or blocks. Each segment is than examined further to find the answers to several key questions:

- Is the segment voiced or unvoiced?
- What is the pitch of the segment?
- What parameters are needed to build a filter that? models the vocal tract for the current segment?

All vocoders, including LPC vocoders, have four main attributes: bit rate, delay, complexity, quality. Any voice coder, regardless of the algorithm it uses, will have to make tradeoffs between these attributes.
The first attribute of vocoders, the bit rate, is used to determine the degree of compression that a vocoder achieves. Uncompressed speech is usually transmitted at 64 kb/s using 8 bits/sample and a rate of 8 kHz for sampling. Any bit rate below 64 kb/s is considered compression. The linear predictive coder transmits speech at a bit rate of 2.4 kb/s, an excellent rate of compression.
Delay is another important attribute for vocoders that are involved with the transmission of an encoded speech signal. Vocoders which are involved with the storage of the compressed speech, as opposed to transmission, are not as concern with delay. The general delay standard for transmitted speech conversations is that any delay that is greater than 300 ms is considered unacceptable. The third attribute of voice coders is the complexity of the algorithm used. The complexity affects both the cost and the power of the vocoder. Linear predictive coding because of its high compression rate is very complex and involves executing millions of instructions per second. The final attribute of vocoders is quality. Quality is a subjective attribute and it depends on how the speech sounds to a given listener.
LPC starts with the assumption that a speech signal is produced by a buzzer at the end of a tube (voiced sounds), with occasional added hissing and popping sounds (sibilants and plosive sounds). The glottis (the space between the vocal folds) produces the buzz, which is characterized by its intensity (loudness) and frequency (pitch). The vocal tract (the throat and mouth) forms the tube, which is characterized by its resonances, which give rise to formants, or enhanced frequency bands in the sound produced. Hisses and pops are generated by the action of the tongue, lips and throat during sibilants and plosives.
LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the
remaining buzz.

The process of removing the formants is called inverse filtering, and the remaining signal after the subtraction of the filtered modeled signal is called the residue.
The numbers which describe the intensity and frequency of the buzz, the formants, and the residue signal, can be stored or transmitted somewhere else. LPC synthesizes the speech signal by reversing the process: use the buzz parameters and the residue to create a source signal, use the formants to create a filter (which represents the tube), and run the source through the filter, resulting in speech.
Because speech signals vary with time, this process is done on short chunks of the speech signal, which are called frames; generally 30 to 50 frames per second give intelligible speech with good compression.

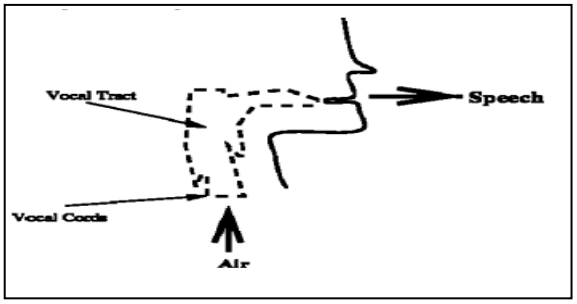

Fig.3: Physical Model of Human.

According to Robert M. Gray of Stanford University, the first ideas leading to LPC started in 1966 when S. Saito and F. Itakura of NTT described an approach to automatic phoneme discrimination that involved the first maximum likelihood approach to speech coding. In 1967, John Burg outlined the maximum entropy approach. In 1969 Itakura and Saito introduced partial correlation, May Glen Culler proposed real time speech encoding, and B. S. Atal presented an LPC speech coder at the Annual Meeting of the Acoustical Society of America. In 1971 real time LPC using 16-bit LPC hardware was demonstrated by Philco-Ford; four units were sold.
In 1972 Bob Kahn of ARPA, with Jim Forgie (Lincoln Laboratory, LL) and Dave Walden (BBN Technologies), started the first developments in packetized speech, which would eventually lead to Voice over IP technology. In 1973, according to Lincoln Laboratory informal history, the first real time 2400 bit/s LPC was implemented by Ed Hofstetter. In 1974 the first real time two-way LPC packet speech communication was accomplished over the ARPANET at 3500 bit/s between Culler-Harrison and Lincoln Laboratories. In 1976 the first LPC conference took place over the ARPANET using the Network Voice Protocol, between Culler-Harrison, ISI, SRI, and LL at 3500 bit/s. And finally in 1978, Vishwanath *et al.* of BBN developed the first variable-rate LPC algorithm.

LPC Coefficients Representation:

LPC is frequently used for transmitting spectral envelope information, and as such it has to be tolerant of transmission errors. Transmission of the filter coefficients directly is undesirable, since they are very sensitive to errors. In other words, a very small error can distort the whole spectrum, or worse, a small error might make the prediction filter unstable. There are more advanced representations such as Log Area Ratios (LAR), line spectral pairs (LSP) decomposition and reflection coefficients. Of these, especially LSP decomposition has gained popularity, since it ensures stability of the predictor, and spectral errors are local for small coefficient deviations.

V. FEATURE EXTRACTION

Humans have a capacity of identifying different types of sounds (phones).Phones put in a particular order constitutes a word. If we want a machine to identify the spoken word, it will have to differentiate between different kinds of sound the way the humans perceive it.

The point to be noted in case of humans is that although, one word spoken by different people produces different sound waves humans are able to identify the sound waves as same. On the other hand two sounds which are different are perceived as different by humans. The reason being even when same phones or sounds are produced by different speakers they have common features. A good feature extractor should extract these features and use them for further analysis and processing.

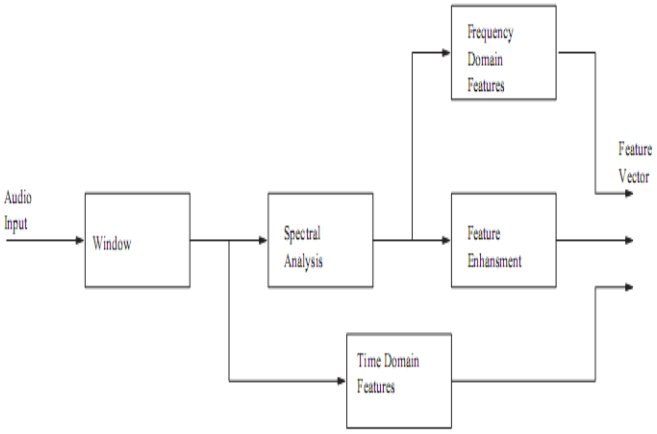

Fig.4: Block Diagram of Feature Extractor.

Feature Extraction refers to the process of conversion of sound signal to a form suitable for the following stages to use. Feature extraction may include extracting parameters such as amplitude of the signal, energy of frequencies, etc.
Linear prediction is a good tool for analysis of speech signals. Linear prediction models the human vocal tract as an *infinite impulse response* (*IIR*) system that produces the speech signal. For vowel sounds and other voiced regions of speech, which have a resonant structure and high degree of similarity overtime shifts that are multiples of their pitch period, this modeling produces an efficient representation of the sound.
Features get periodically extracted. The time for which the signal is considered for processing is called a window and the data acquired in a window is called as a frame.
Typically features are extracted once every 10ms, which is called as frame rate. The window duration is typically 25ms. Features are then extracted from each of frame.
Most of the features can be categorized into two categories:
Temporal Feature

– Power spectral analysis (FFT)
– Linear predictive analysis (LPC)
– Mel scale cepstral analysis (MEL)
– First order derivative (DELTA)

Spectral Feature
– Energy normalization
– Zero Crossing Rate
In pattern recognition and in image processing, **feature extraction** is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called *feature extraction*. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.
Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over fits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

VI. SOUND RECORDING AND WORD DETECTION

The component responsibility is to accept input from a microphone and forward it to the feature extraction module. Before converting the signal into suitable or desired form it also does the important task of identifying the segments of the sound containing words. It also has a provision of saving the sound into WAV files which are needed by the training component.
Internally, it is the job of Sound Reader class to take the input from the user. The Sound Reader class takes sampling rate, sample size and number of channels as parameters.
Sound Reader has three basic functions: open, close and read. Open function opens the /dev/dsp device in the read mode. It makes appropriate ioctl calls to set the device parameters. Close function releases the dsp device. Read function reads from the dsp device checks if there is a valid sound present and returns the sound content.
In speech recognition it is important to detect when a word is spoken. The system does detect the region of silence.

VII. EXPERIMENTAL RESULTS AND OBSERVATIONS

MATLAB software has various inbuilt functions to implement audio functions and coefficient evaluation. Spoken words of speaker were stored in a bank and their LPC coefficients were determined along with energy and zero crossover detection as well [3]. Later the words were spoken by another speaker whose phonemes and accent were sent to fuzzy analyzer for correct interpretation in intelligent way using fuzzy approach discussed above. Expert system was tested on 120 utterances in English spoken by two male speakers. It was observed that with just two parameters (total energy and first linear prediction coefficient) along with their fuzzy thresholds, spoken words were identified with more than 90% accuracy.

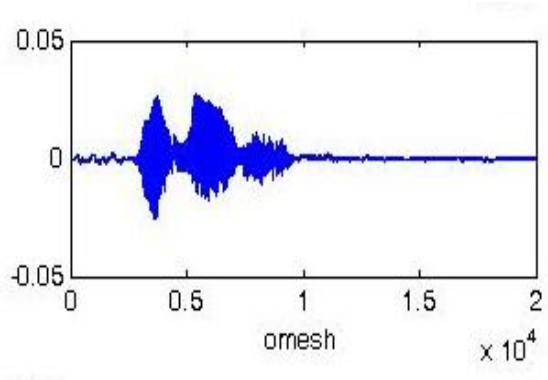

Fig.5. Wave Plot for word 'omesh'

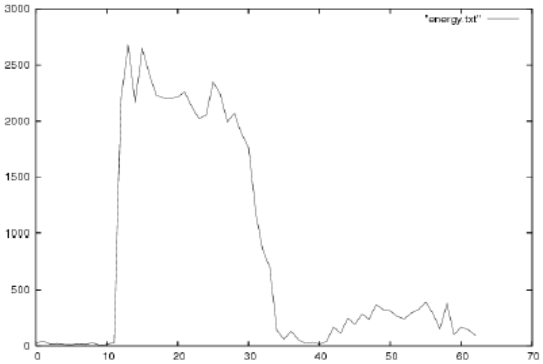

Fig.6. Energy plot for spoken word "omesh"

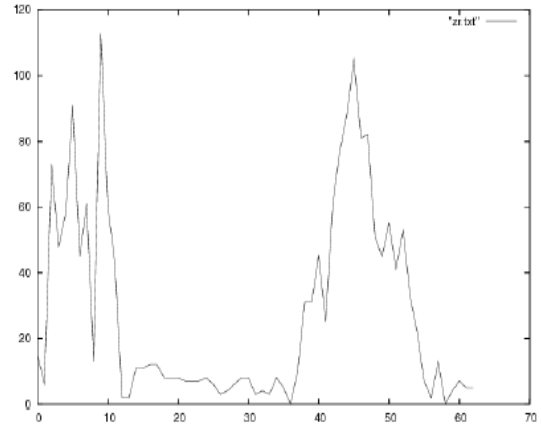

Fig.7. Zero Crossing plot for spoken word "omesh"

VIII. APPLICATIONS

LPC is generally used for speech analysis and resynthesis. It is used as a form of voice compression by phone companies, for example in the GSM standard. It is also used for secure wireless, where voice must be digitized, encrypted and sent over a narrow voice channel; an early example of this is the US government's Navajo I.
LPC synthesis can be used to construct vocoders where musical instruments are used as excitation signal to the time-varying filter estimated from a singer's speech. This is somewhat popular in electronic music. Paul Lansky made the well-known computer music piece not just more idle chatter using linear predictive coding.

A 10th-order LPC was used in the popular 1980's Speak & Spell educational toy. Waveform ROM in some digital sample-based music synthesizers made by Yamaha Corporation may be compressed using the LPC algorithm.

IX. CONCULSION

Linear Predictive Coding is an analysis/synthesis technique to lossy speech compression that attempts to model the human production of sound. Linear Predictive Coding achieves good bit rate which makes it ideal for secure telephone systems. Secure telephone systems are more concerned that the content and meaning of speech, rather than the quality of speech, be preserved. The trade off for LPC's low bit rate is that it does have some difficulty with certain sounds and it produces speech that sound synthetic. LPC encoders break up a sound signal into different segments and then send information on each segment to the decoder. The encoder send information on whether the segment is voiced or unvoiced and the pitch period for voiced segment which is used to create an excitement signal in the decoder. Vernacular language work is not concluded yet. Hence this paper is light on such approach for enhancing recognition power of intelligent techniques along with feature extraction. Experimental results also confirm the same.

ACKNOWLEDGMENT

The Authors place on record their grateful thanks to the authorities of Rungta College of Engineering for providing all the facilities for accomplishing this paper.

REFERENCES

1. L. R. Rabiner and R. W. Schafer, Digital Speech Processing. Prentice-Hall, 1978.
2. M. Forsberg. 2003. Why Speech Recognition is Difficult. Chalmers University of Technology.
3. J. R. Deller, J. H. L. Hansen, J. G. Proakis. Discrete-time Processing of Speech Signals. IEEE Press. 1993.
4. Thiang, Suryo Wijoyo. Speech Recognition using LPC and Artificial Neural Network for controlling the movements of robot, 2011.
5. N.Uma Maheswari, A.P.Kabilan, R.Venkatesh "Speech Recognition system based on phonemes using neural networks". JCSNS International Journal of Computer Science and Network Security, Vol.9 No.7, July 2009.
6. Asim Shahzad, Romana Shahzadi, Farhan Aadil "Design and software implementation of efficient speech recognizer". International Journal of Electrical & Computer Sciences IJECS-IJENS Vol. 10.
7. R. Rodman, Computer Speech Technology. Artech House, Inc. 1999, Norwood, MA 02062.
8. J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
9. C. H. Lee; F. K. Soong; K. Paliwal "An Overview of Speaker Recognition Technology". Automatic Speech and Speaker Recognition: Advanced Topics. Kluwer Academic Publishers, 1996, Norwell, MA.
10. S. R. Jang, "Neuro Fuzzy Modeling Architectures, Analyses and Applications". University of California, Berkeley, Canada, 1992.
11. P. P. Bonissone, "Adaptive Neural Fuzzy Inference Systems (ANFIS): Analysis and Applications", technical report, GE CRD, Schenectady, NY USA, 2002.
12. P. Eswar, S.K. Gupta, C. Chandra Sekhar, B. Yegnanarayana and K. Nagamma Reddy, An acoustic phonetic expert for analysis and processing of continuous speech in Hindi, in *Proc. European Conf. on Speech Technology*, Edinburgh, vol. 1 (1987) 369-372.
13. J.P. Haton, Knowledge based approach in acoustic phonetic decoding of speech, in: H. Niemann, M. Lang and G. Serger, Eds., *Recent Advances in Speech Understanding and Dialog Systems*, NATO-ASI Series, vol. 46, (1988) 51-69.
14. W.A. Lea, Ed., *Trends in Speech Recognition* (Prentice Hall, Englewood Cliffs, N J, 1980).

AUTHORS PROFILE

Mr.Omesh Wadhwani is a B.E graduate from Kavikulguru Institute of Technology and Science, Ramtek with the specialization in Information Technology. He owns more than two year of Teaching experience. **Currently**, He is doing his Master of Technology in Digital Electronics from Rungta College of Engineering and Technology, Bhilai. His area of interest are Speech processing and Pattern Recognition, Image Processing, Database Management systems and Artificial Intelligence.

Prof.Amit Kolhe is a Associate Professor in Rungta College of Engineering and Technology and His areas of Interest are Information theory and Coding, Digital Speech and Image Processing, Signal Representation and Processing.

Prof.Sanjay Dekate is a Associate Professor in Department of Electronics and Telecommunication Engineering, Rungta College of Engineering and Technology(RCET, Bhilai) and He is having more than 10 years of Teaching Experience. His areas of Interest are Pattern Recognition, Information Security, Mobile communication and Bio-metric Applications.