Ensemble Feature Extraction Modules for Improved Hindi Speech Recognition System

[1]Malay Kumar, [1]R K Aggarwal, [2]Gaurav Leekha and [1]Yogesh Kumar

[1]Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India

[2]Department of Computer Science and Engineering, M. M. University, Solan, Himachal Pradesh, India.

Abstract

Speech is the most natural way of communication between human beings. The field of speech recognition generates intrigues of man – machine conversation and due to its versatile applications; automatic speech recognition systems have been designed. In this paper we are presenting a novel approach for Hindi speech recognition by ensemble feature extraction modules of ASR systems and their outputs have been combined using voting technique ROVER. Experimental results have been shown that proposed system will produce better result than traditional ASR systems**.**

Keywords: ASR, MFCC, PLP, LPCC, ROVER.

1. Introduction

In the world of science fiction, computers have always understood human mimics. This idea generates interest to make such speech recognition systems which are able to understand human mimics because it is always convenient to interact with a computer, robot or any machine through speech rather than complex instructions. Our daily needs like railway inquire system, mobile applications, weather forecasting, agriculture, healthcare etc can be benefited by speech recognition because communicating with an information gathering system in natural language for getting information is much easier than interacting through keyboard or mouse. Many research groups and major companies like Microsoft, SAPI and Dragon Naturally Speech are working on this field but especially they are focusing on European languages and English. Although significant work has been done for South Asian language including Hindi but none of them have given satisfactory results. This paper aims to ensemble feature extraction modules (MFCC, PLP and LPCC) of ASR systems and the outputs of individual ASR system has been combined using voting technique ROVER. Paper has been prepared in following order Section 2 presents architecture of ASR and its function. Section 3 explains ROVER and proposed model combination. Section 4 presents implementation and comparison of proposed system to conventional systems. Section 5 is conclusion.

2. System Architecture of for Automatic Speech Recognition System

The basic model of ASR system is divided into two parts front end and back end as shown in Figure 1. Front-end covers preprocessing and feature extraction phase while back-end covers acoustic modeling, language model, pattern recognition.

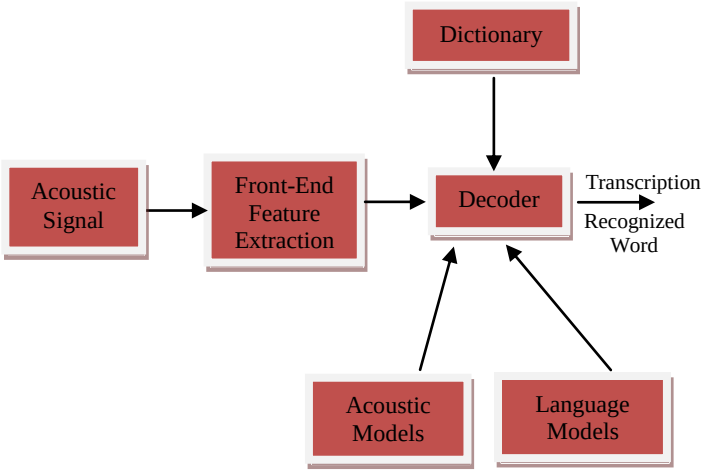

Fig. 1 ASR Architecture.

2.1 Preprocessing/Digital Processing

The recorded acoustic signal is an analog signal that cannot be directly processed by ASR systems, so these speech signals are transformed in the form of digital signals so that they can be processed. The digital signal is made to pass through the first order filters to spectrally flatten the signals. The result of this step is to increase the magnitude of higher frequency as compared to lower frequency. The next step is to divide the speech signals

into frames of 10 to 25 milliseconds with an overlap of 50% to 70% between consecutive frames.

2.2 Feature Extraction

The goal of feature extraction module is to find a set of parameters of utterance that have acoustic correlation with speech signals and these parameters are computed through processing of the acoustic waveform. Such parameters are termed as features. The main aim of feature extractor is to keep the relevant information and discard irrelevant one. To perform this operation, feature extractor divides the acoustic signal into 10-25 ms of window size with an overlap of 50% to 70% between consecutive frames. The data acquired in these frames is multiplied by window function. There are several types of windows like Rectangular, Hamming, Hanning, Bartlett, Blackman, Welch or Gaussian can be used. After that, features have been extracted from every frame. There are several methods to extract features from each frame such as Linear Predictive Cepstral Coefficient (LPCC),Mel-Frequency Cepstral Coefficient (MFCC), Perceptual Linear Prediction (PLP), Wavelet and RASTA-PLP (Relative Spectral Transform) processing.

2.3 Acoustic Modeling

In this subsystem, the connection between the acoustic information and phonetics is established. The connection can be established either at word or at phoneme level. There are many models for this purpose, but Hidden Markov Model (HMM) is the most widely used and accepted technique because of its efficient algorithm for training and recognition. It is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters. This algorithm is often used due to its simplicity and feasibility of use. Hidden Markov models (HMM) are the most popular (parametric) model at the acoustic level.

2.4 Language Modelling

A language model contains the structural constraints available in the language to generate the probabilities of occurrence. Intuitively speaking, it determines the probability of a word occurring after a word sequence. It is easy to see that each language has its own constraints for validity. ASR systems use bi-gram, Tri-gram, n-gram language models to guide the search for correct word sequences by predicating the likelihood of the n^{th} word, using the n-1 previous words. The probability of occurrence of word *W* is calculated as

$$P(W) = P(w_1, w_2, ...w_{n-1}, w_n)$$
$$= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2).....P(w_n \mid w_1 w_2 ... w_{n-1})$$
$$P(W) = \prod_{k=1}^{n} P(w_k \mid w_1, w_2 .... w_{k-1}) \quad (1)$$

2.5 Recognition

Once the pre-processing of user's input is complete the recognizer is ready to perform recognition. The recognition problem can be stated as finding the most probable sequence of words *W* given the acoustic input *O*, which is computed as:

$$P(W \mid O) = \frac{P(O \mid W).P(W)}{P(O)} \quad (2)$$

Given an acoustic observation sequence *O*, classifier finds the sequence of words *W* which maximizes the probability $P(O \mid W).P(W)$. The quantity $P(W)$, is the prior probability of the word which is estimated by the language model. $P(O \mid W)$ is the observation likelihood, known as acoustic model. Given an acoustic observation sequence *O*, the efforts on the maximization of $P(W|O)$ can be moved to the search for sequence of words *W* which maximizes the numerator of the right-hand side of equation i.e. $P(O|W).P(W)$. The quantity $P(W)$, usually referred to as the **Language Model (LM)** depends on high-level constraints and linguistic knowledge about the allowed word strings for a specific task. The quantity $P(O|W)$ is known as the **Acoustic Model (AM)**. It describes the statistics of sequences of parameterized acoustic observations in the feature space given the corresponding uttered words.

3 System Combination

A variety of tools and techniques have been proposed by researchers for the development of speech recognition systems. Each approach has some merits and demerits, means some ASR system performs better in some environment while its performance degraded in other environment. For instance, the feature extraction technique PLP outperforms MFCC, when training and testing conditions are mismatched. But with similar training and testing conditions, MFCC is better than the PLP. Both the techniques are computationally expensive. LPCC works well in clean environment but its performance gets degraded in noisy environment; it takes low computation power and little time to extract the features. Both the feature extraction techniques MFCC and PLP perform better than LPCC. Above explanations show the positive and negative performances of different feature extraction techniques in different environment and conditions. If we are able to ensemble these feature extraction techniques in one system, the developed system will perform better in general field conditions as well as in clean environment. The developed

combination system is shown in Fig 2. The developed system encapsulates three individual ASR systems and the system will produce output using voting technique ROVER.

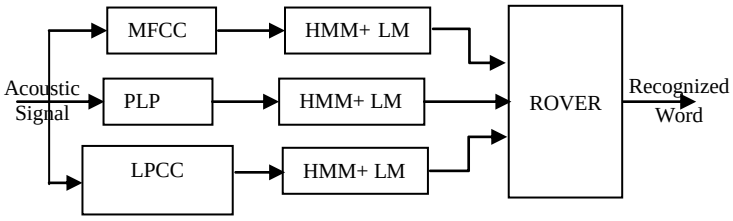

Fig. 2 Ensemble System

The developed ensemble system will produce reduced error rate for ASR systems, by exploiting differences in the nature of error produced by multiple ASR systems.

3.1 Related Work

Ensemble of multiple ASR systems was first proposed by Waibel. Where several time delay neural networks were developed for different subsets of confusable consonants and the outputs of these sub-networks were combined to determine the consonant class. In another approach by combining a BU system based on stochastic segment models (SSM) and a BBN system based on Hidden Markov Models. It was a generalization of integrating two or more speech recognition technologies, working on different strategy.

3.2 Rover

A system combination method was developed at National Institute of Standards and Technology (NIST) to produce a composite Automatic Speech Recognition (ASR) system output when the outputs of multiple ASR systems were available, and for which, in many cases, the composite ASR output had a comparatively lower error rate. It was referred to as a NIST Recognizer Output Voting Error Reduction (ROVER) system developed by Fiscus. It is implemented by employing a "voting" scheme to exploit differences in ASR system outputs. The outputs of multiple of ASR systems have been combined into a single minimal cost word transition network (WTN) via iterative applications of dynamic programming alignments. The ROVER system is implemented in two modules as shown in Figure 3. The first module takes the outputs from two or more ASR systems and combined those outputs into a single word transition network. The network is created using a modification of the dynamic programming alignment protocol traditionally used by NIST to evaluate ASR technology. Once the network is generated, the second module evaluates each branching point using a voting scheme, which selects the best scoring word having the highest number of votes for the new transcription.

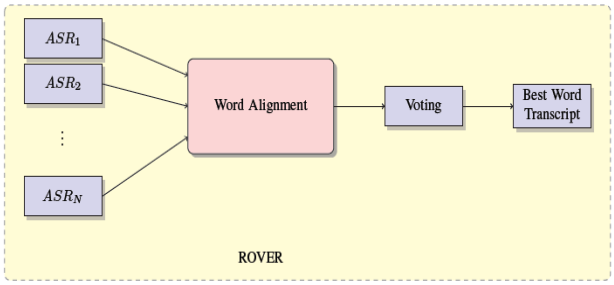

Fig. 3 Rover System Architecture

3.2.1 Word Alignment

Word alignment process is governed by four rules named Correct, Substitution, Deletion and Insertion.
These are four rules have taken in consideration during the WTN-alignment process

a) In **Correct** rule a copy of the word transition arc from WTN-2 is added to the corresponding word in WTN - BASE.
b) In **Substitution** rule a copy of the word transition arc from WTN-2 is added to WTN-BASE.
c) In **Deletion** rule a no-cost, NULL word transition arc is added to WTN-BASE.
d) In **Insertion** rule a sub-WTN is created and inserted between the adjacent nodes in WTN-BASE to record the fact that the WTN-2 network supplied a word at this location.

The above rules are given by J.Fiscus, here we are explaining these rules by applying them on our proposed system, figure 4 shows the three WTNs before alignment and the first WTN, WTN-1 is designated as the base WTN from which the composite WTN is developed. We align the second WTN to the base WTN using the DP alignment protocol and augment the base WTN with word transition arcs from the second WTN as appropriate. The alignment yields a sequence of correspondence sets between WTN-BASE and WTN-2.

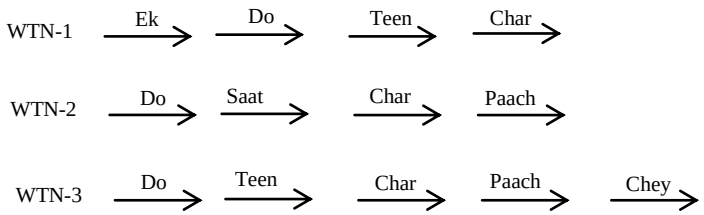

Fig. 4 WTN-before Alignment

A new composite WTN-base is made by copying word transition arcs from WTN-2 to new WTN-BASE as shown in figure 5. When copying the arcs into WTN-BASE, the above mentioned rules are applied as follows,

(DO and CHAR in the example implementation of rule1 Correct): A copy of the word transition arc from WTN-2 is added to the corresponding word in WTN-BASE. Substitution, ***(SAAT*** in the example): a copy of the word transition arc from WTN-2 is added to WTN-BASE. Deletion, (@ in the example): a no-cost, NULL word transition arc is added to WTN-BASE. Insertion, (Paach in the example): a sub-WTN is created and inserted between the adjacent nodes in WTN-BASE to record the fact that the WTN-2 network supplied a word at this location and new WTN-Base is created.

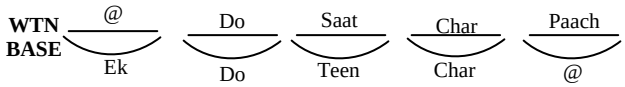

Fig. 5 WTN 2 align with WTN-Base

The process is again repeated for WTN-3 that will merge WTN-3 into WTN-BASE. Figure 6 shows the final base WTN which is passed to the next module to select the best scoring word sequence.

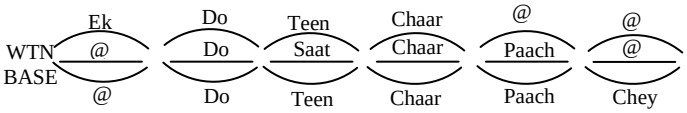

Fig. 6 Final WTN-BASE

3.2.2 Rover Voting Mechanism

The combined ASR systems necessarily have to supply a word confidence ranging between 0 and 1 for each word. These word confidences can be considered as the amount of confidence of each ASR pertaining to each word output. For this purpose, Confidence estimation is performed for each training set before combining them. The voting scheme is controlled by parameters α and null confidence N_c that weigh Frequency of occurrence and Average Confidence score. These two parameters, tuned for a particular training set which are later used for validations. The voting mechanism of ROVER can be performed in three ways by prioritizing frequency of Occurrence, frequency of Occurrence and average word confidence, frequency of Occurrence and Maximum confidence score. The score $S(W_i)$ is calculated as

$$S(W_i) = \alpha F(W_i) + (1-\alpha)C(W_i) \quad (3)$$

where $F(W_i)$ is the frequency of occurrence and $C(W_i)$ is the word confidence.

Frequency of Occurrence

Setting the value of α to 1.0 in Eq. 3 nullifies confidence scores in voting. The major disadvantage of this method of scoring is that the composite WTN can contain deletions or missing words.

Average Word Confidence

Missing words are substituted by a null confidence score. Optimum null confidence score $C(W_i)$ is determined during training.

Maximum Confidence Score

This voting scheme selects the word sequence that has maximum confidence score by setting the value of α to 0.

3.3 Ensemble Subsystems

The different feature extraction modules used with ASR systems has some benefits and some shortcomings; recognition of speech by using only a single module is retro approach, in this paper we are giving a novel approach for Hindi speech to combine multiple ASR systems with different feature extraction module. When we discuss about ensemble multiple ASR system the main question come across, how these systems have been combined, a simple and straight forward approach is that each ASR system decodes the speech signal individually and combine their output hypothesis using recognizer output voting error reduction(ROVER) technique. The next question arise how many such ASR systems should be combined to give an optimal result and how it will work in real time environment with its performance better than individual system. Schwenk and Gauvain have given an analysis that only three best recognizers should be combined to produce an optimal result. Thus we prepare three different ASR systems having their own feature extraction technique. The first system uses PLP as its feature extraction module, second uses MFCC and third uses LPCC. The ASR system produces their output hypothesis individually and ROVER is used to combine the output hypothesis by first aligning the most likely hypothesis and then selecting the one having best score.

3.4 Data preparation for Hindi language

Hindi is most widely used language in India and it is also the national language of India. Although different places have different dialects and way of talking, so it is also phonetic in nature. It is usually written in Nagari or Devnagari script. Any script can be broadly divided into two main categories vowels and consonants. Hindi language has 12 vowels and 36 consonants, total 48 characters in Hindi alphabets. There is a unique specialty of Hindi language that the orthographical representation of Hindi language is similar to its phoneme that is because of close correspondence of grapheme to phoneme. English language doesn't have this functionality because English letters don't have unique pronunciation. In Hindi, there is separate symbol for

each vowel. The consonants themselves have an implicit vowel + (अ). To indicate a vowel sound other than the implicit one (i.e. अ), a vowel-sign (Matra) is attached to the consonant. The vowels with equivalent Matras are shown in table 1.

Table 1: Hindi Vowels

Vowel	Matra	Vowel	Matra
अ	-	ए	े
आ	ा	ऐ	ै
इ	ि	ओ	ो
ई	ी	औ	ौ
उ	ु	ऋ	ृ
ऊ	ू	ऋ	ृ

The consonants of Hindi language are given in the table 2 in which consonants are divided into five groups and each group have five characters, four semi vowels, three sibilants and a aspirate, besides these consonants there are three more consonants क्ष, त्र , ज्ञ so total of 36 consonants.

Table 2: Hindi Character Set

Phonetic Property	Primary Consonants (unvoiced)		Secondary Consonants (voiced)		Nasal
Category	Un-aspirated	Aspirated	Un-aspirated	aspirated	
Gutturals (कवर्ग)	क	ख	ग	घ	ङ
Patatals (चवर्ग)	च	छ	ज	झ	ञ
Cerebrals (टवर्ग)	ट	ठ	ड	ढ	ण
Dental (तवर्ग)	त	थ	द	ध	न
Labials (पवर्ग)	प	फ	ब	भ	म
semivowels	य, र, ल, व				
Sibilants	श, ष, स				
Aspirate	ह				

Other Characters: - Besides vowels and consonants there are some other characters also exist known as anuswar (ं), visarga (ः), chanderbindu (ँ).

4. Implementation Work

4.1 System Description

The system is developed on Linux platform (Ubuntu 10.04 LTS edition). Many open source speech development tools are present mainly are HTK Tool-Kit developed by Cambridge University, Sphinx from Carnegie Mellon University and Julius developed by Japanese for LVCSR. We have developed the system using HTK Tool-Kit (V3.4). The reason behind to chosen HTK Tool-Kit is its availability of proper documentation [HTK Book Cambridge University]. Development of ASR system is starts with collecting the data; audacity is used for the purpose of recording. The specification of speech file is 16 KHz sampling rate with 16 bits/sec and mono channel, Next phase is preprocessing and feature extraction HCOPY command is used for this purpose with a separate configuration file defining parameters for each feature extraction technique. For acoustic modeling HINIT command is used to initialize HMMs and in a separate file prototype is define for each phone, initialization is an important step because successive iteration depends on this step, in this stage own topology and number of states can be defined in a prototype file. HREST is used to re–estimate HMMs. HPARSE command is used to change the standard grammar to HTK Standard Lattice Format (SLF) in which each word instance and each word-to-word transition is listed explicitly. HTK provides a command called HVITE to decode direct audio input.

4.2 Data Preparation

The main research has been done in the area of speech recognition is mainly for European languages and some significant development for Japanese but unfortunately researchers took less initiative for Indian languages. We have developed our own corpus, due to the unavailability of speech and text corpus and for the purpose of recording we are using 120 VA, unidirectional Sony microphone. The speech database contains 200 words, data is recorded using unidirectional microphone its distance from speaker is 5-10 cm, and recording had been done in clean environment, with sampling rate of 16 KHz and 16 bits /sample. Voices of ten persons (8 males, 2 females) have been used to train the system. Every word was recorded four times. Thus the total number of 200(10*4) speech files has stored in system in .wav format.

4.3 Evaluating Performance Result

Once the ASR system is properly has been trained, it is ready for test against the test data. HTK provides HRESULT to produce the result of ASR system. It

compares the transcribed output by HVITE with original reference output and then it gives the various statistics. The performance of ASR system is evaluated as

$$Percentage\ of\ Correct\ Words = \frac{N-D-S}{N} * 100 \quad (4)$$

Where N is the total number of words in the test set, D is the number of deletions, S number of substitutions.
The *Accuracy evaluation* is computed as

$$Percentage\ of\ Accuracy = \frac{N-D-S-I}{N} * 100 \quad (5)$$

Where 'I' is the number of insertions.
The performance of ASR systems in terms of word error rate is evaluated as

$$WER = \frac{S+D+I}{N} * 100 \quad (6)$$

4.4 System Performance

The developed system has been tested in clean environment and in general field conditions with seen and unseen speakers (seen speakers means their data sample is recorded in corpus while unseen speakers means system does not have their samples), on the basis of performance analysis the percentage of correct word recognition of individual systems and combination system is shown in the figure 7 and the recognition result

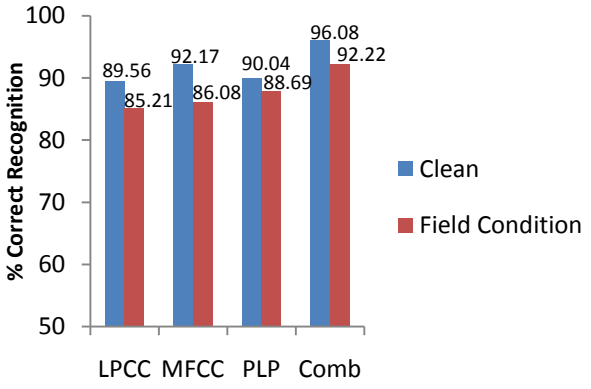

Fig. 7 System performance percentage of correct word recognition

of the combination system is better than individual ASR systems. The comparison of recognition results of individual ASR system and ensemble system given in table 3, table shows the number of correct words recognized by each ASR system and our proposed ensemble system results show significant improvement in the result.

Table 3: Comparison of Combination System Performance with Traditional Systems.

		Speaker 1 Seen	Speaker 2 Seen	Speaker 3 Unseen	Speaker 4 Unseen
No of Words		50	55	60	65
LPCC	Clear	46	53	51	56
	Field Condition	45	52	48	51
MFCC	Clear	49	55	52	56
	Field Condition	46	52	49	51
PLP	Clear	48	54	51	55
	Field Condition	48	53	50	53
Comb System	Clear	50	55	56	60
	Field Condition	49	54	52	57

5 Conclusion

The paper presents ensemble of speech recognition system for Hindi language using Rover technique. The system has been developed in Linux environment using HTK Tool-Kit. System was trained for 200 words, with ten different speakers have been used to record the speech data and three feature extraction techniques are used MFCC, PLP and LPCC. The proposed system has been used ROVER technique to combine these feature extraction modules. Results have shown that the proposed system give better recognition results in comparison to the traditional systems with an increment of 4% in accuracy. The proposed system has given good performance with vocabulary size of 200 words with the accuracy of 96% in clean environment and 92% in general field condition. The future work involves the development of system for more vocabulary size and improves the accuracy of system in different environments.

References

[1] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)", In Proceeding of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'97), Santa Barbara, 1997, pp. 347–352.
[2] Becchetti Claudio and Ricotti Klucio Prina, Speech Recognition Theory and C++ Implementation, Wiley Publisher, 2004.
[3] H. Hermansky, "Perceptually predictive (PLP) analysis of speech", Journal of Acoustic Society of America, Vol. 87, pp. 1738-1752, 1990.
[4] M. Kumar, A. Verma, and N. Rajput, "A Large Vocabulary Speech Recognition System for Hindi," Journal of IBM Research, Vol. 48, 2004, pp. 703-715.
[5] Hidden Markov Model Toolkit (HTK-3.4.1): http://htk.eng.cam.ac.uk.
[6] C. M. Bishop, Pattern Recognition and Machine Learning Springer, 2006.

[7] S. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 28, 1980, pp.357-366.
[8] S. Furui S., "Cepstral Analysis Technique for Automatic Speaker Verification", IEEE Transactions on ASSP, Vol. 29, No. 2, 1981, pp. 254-272.
[9] M. Gale and S. Young, "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends in Signal Processing, Vol.1, No. 3, 2007, pp. 195-304.
[10] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, 1989, pp. 257-286.
[11] S. Young, G. Evermann, M. Gales and P. Woodland, The HTK Book. Microsoft Corporation and Cambridge University Engineering Department, 2009.
[12] R.K. Aggarwal and M. Dave, "Discriminative Techniques for Hindi Speech Recognition System", Communication in Computer and Information Science (Information Systems for Indian Languages), Springer-Verlag Berlin Heidelberg, Vol. 139, 2011, pp. 261-266.
[13] Kuldeep Kumar and R.K. Aggarwal, "A Hindi speech recognition system for connected words using HTK", Int. J. of Computational Systems Engineering, Vol.1, No.1, 2012, pp.25 - 32.
[14] R.K. Aggarwal and M. Dave, "Integration of multiple acoustic and language models for improved Hindi speech recognition", Int. J. of Speech Technology, Springer, DOI 10.1007/s10772-012-9131, 2012.
[15] R.K. Aggarwal and M. Dave, "Acoustic Modeling problem for speech recognition system: conventional methods (PART I)", Int. J. of Speech Technology, Springer, Vol 14, No 4, 2011, pp. 297-308.
[16] R.K. Aggarwal and M. Dave, "Acoustic Modeling problem for speech recognition system: advances and refinement (PART II)", Int. J. of Speech Technology, Springer, Vol 14, No 4, 2011, pp. 309-320.
[17] M. Ostendorf et. al, "Integration of diverse recognition methodologies through reevaluation of nbest sentence hypotheses", In Proceedings DARPA Speech and Natural Language Processing Workshop, 1991, page 83-87.
[18] A. Waibel, H. Sawai, and K. shikano, "Modularity and scaling in large phonemic neuralnetworks", IEEE Transaction on ASSP, Vol. 37, No. 12, 1989, pp. 1888-1898.
[19] Schwenk Holger and Gauvain Jean-Luc, "Combining Multiple Speech Recognizers using Voting and Language Model Information", IEEE International Conference on Spoken Language Processing (ICSLP), Pekin, 2000, pp. 915–918.

First Author Malay Kumar was received his B. Tech. degree from Kanpur University, Kanpur, India in 2010 and pursuing his M. Tech. degree from prestigious National Institute of Technology, Kurukshetra, India. He is working in the area of speech processing from last one and half year and also opt this area as his dissertation work, his research work involves around working with different open source recognition tools, implementation of various modeling units' word, phoneme, triphone and syllable models and working with system integration techniques like Rover for Hindi language.

Second Author R. K. Aggarwal was received his M. Tech. degree in 2006 and pursuing PhD from National Institute of Technology, Kurukshetra, INDIA. Currently he is also working as an Associate Professor in the Department of Computer Engineering of the same Institute. He has published more than 24 research papers in various International/National journals and conferences and also worked as an active reviewer in many of them. He has delivered several invited talks, keynote addresses and also chaired the sessions in reputed conferences. His research interests include speech processing, soft computing, statistical modeling and science and spirituality. He is a life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). He has been involved in various academic, administrative and social affairs of many organizations having more than 20 years of experience in this field.

Second Author Gaurav Leekha has received his M.Tech degree in 2010 from Kurukshetra University, INDIA. Currently he is working as an Asst. Professor in Computer Science and Engineering department of M.M. University, Solan, Himachal Pardesh, INDIA. He is working in the area of speech recognition for Indian languages from last 3 years and published several papers in National/International conferences. He has also attended many workshops on speech recognition in various reputed institutes.

Third Author Yogesh Kumar is M.Tech. student in National Institute of Technology, Kurukshetra, India. He have great interest in the area of speech processing for Indian languages.