A Study on Speech Recognition Techniques in Regional Languages

Pramod Mathur[1], Harshika Mathur[2]
Assistant Professor, CSE
Pratap Institute of Technology and Science, Sikar
erpramodmathur@gmail.com
harshimathur17@gmail.com

Abstract- Speech interface to computer is the next big step that computer science needs to take for general users. *Speech recognition* will play a important role in taking technology to them. The need is not only for speech interface, but speech interface in local languages. Our goal is to create speech recognition software that can recognize Hindi words. It will tell a brief look at the basic building block of a speech recognition engine.

*****

I. INTRODUCTION

Keyboard, although a popular medium is not very convenient as it requires a certain amount of skill for effective usage. A mouse on the other hand requires a good hand-eye co-ordination. It is also cumbersome for entering non-trivial amount of text data and hence requires use of an additional media such as keyboard. Physically challenged people find computers difficult to use. Partially blind people find reading from a monitor difficult. Current computer interfaces also assume a certain level of literacy from the user. It also expects the user to have certain level of proficiency in English. In our country where the literacy level is as low as 50% in some states, if information technology has to reach the grass root level; these constraints have to be eliminated. Speech interface can help us tackle these problems. Speech Synthesis and Speech Recognition together form a speech interface. A speech synthesizer converts text into speech. Thus it can read out the textual contents from the screen. Speech recognizer had the ability to understand the spoken words and convert it into text. We would need such software's to be present for Indian languages.

1.1 Existing Systems [9]

Although some promising solutions are available for speech synthesis and recognition, most of them are tuned to English. The acoustic and language model for these systems are for English language. Most of them require a lot of configuration before they can be used. There are also projects which have tried to adapt it to Hindi or other Indian Languages. Explains how an acoustic model can be generated using aexisting acoustic model for English. ISIP and Sphinx are two of the known Speech Recognition software in open source. Gives a comparison of public domain software tools for speech recognition. Some commercial software like IBM's Via Voice is also available.

1.2 Some important discussions:

Sound Recording and Word Detection[10]

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.

Sound Recorder [6]

The recorder takes input from the microphone and saves it or forwards it depending on which function is invoked Recorder supports changing of *sampling rate, channels* and *sizeof the sample*. Default sampling rate of the recorder is 44100 samples per second, at a size of 16 bits per sample and dual channel.

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 deviceparameters. 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. The Sourcecode of *Yet Another Recorder* (yarec) was referred for making the sound recorder.The Recorder class takes care of converting the raw audio signal into WAV format andstores it to a file. The format of WAV file if given in the appendix for reference.

Word Detector [7]

The Principle In speech recognition it is important to detect when a word is spoken. The system does detect the region of silence. Anything other than silence is considered as a spoken word by the system. The system uses energy pattern present in the sound signal and zero crossing rate to detect the silent region. Taking both of them is important as only energy tends to miss some parts of sounds which are important. This technique has been described in

For word detection a sample is taken every 10 milli-seconds. Energy and zero crossing for this duration is calculated. Energy is calculated by adding the square of the value of waveform at each instance and then dividing it by to number of instances over the period of sample. Zero crossing rate is the number of times the value of the wave goes from the negative number to positive of vice-versa.

Word Detector assumes that the first 100 milli-second is silence. It uses the average Energy and average Zero Crossing Rate obtained during this time for identifying the background noise. Upper threshold for energy and zero crossing is set to 2 times the average value of background noise. Lower thresholds are set to 0.75 times the upper threshold. While detecting the presence of word in the sound, if the energy or zero crossing goes above the upper threshold and stays above for three consecutive sample words is assumed to be present and the recording is started. The recording continues till the energy and zero crossing both fall below the lower threshold and stay there for at-least 30 milli-seconds.

Feature Extractor[7]

Humans have a capacity of identifying different types of sounds (phones). A phone 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.

Feature Vector specification[10][13]

Vectors were generated at frame duration of 10 milli-seconds. Window used was Hamming window with duration of 25 milli-seconds. 12 MFCC and energy level are generated for each frame. These features can now be used for either recognition or for training the HMM.

II. LITERATURE SURVEY

2.1 HMM Recognition and Training [17]

Hidden Markov Model (HMM) is a state machine. The states of the model are represented as nodes and the transition are represented as edges. The difference in case of HMM is that the symbol does not uniquely identify a state. The new state is determined by the symbol and the transition probabilities from the current state to a candidate state.

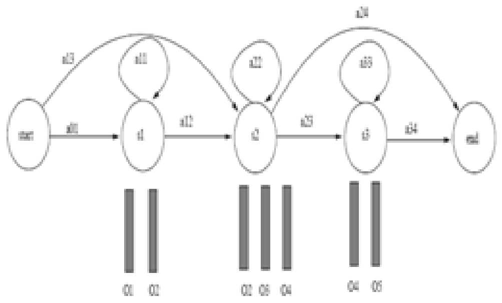

Figure2.1 HMM

The above figure shows a diagrammatic representation of a HMM. Nodes denoted as circles are states. *O*1 to *O*5 are observations. Observation *O*1 takes us to states *S*1. *aij*defines the transition probability between *Si* and *Sj*. It can be observed that the states also have self transitions. If we are at state *S*1 and observation *O*2 is observed, we can either decide to go to state *S*2 or stay in state *S*1. The decision is made depending on the probability of observation at both the states and the transition probability.

Recognition using HMM[17]

We need to recognize a word using the existing models of words that we have. Sound recorder need to record the sound when it detects the presence of a word. This recorded sound is then passed through feature vector extractor model. The output of the above module is a list of features taken every 10 msec. These features are then passed to the Recognition module for recognition. The list of all the words that the system is trained for and their corresponding models are given in a file called *models* present in the *hmms*. All models corresponding of the words are then loaded in memory. The feature vectors generated by the feature vector generator module act as the list of observation for the recognition module. Probability of generation of the observation given a model, *P*(*O*|¸), is calculated for each of the model using *find probability* function.

Training the Model[16]

Before we can recognize a word we need to train the system. *Train* command is used to train the system for a new word. The command takes at-least 3 parameters:

- No of states the HMM model should have *N.*
- The size of the feature vector *D.*
- One or more filenames each containing a training set.

For generating an initial HMM we take the N equally placed observations (feature vector) from the first training set. Each one is used to train a separate state. After training the states have a *mean* vector which is of size D. And a *variance* matrix of size D ¤ D containing all zeros. Then for each of the remaining observations, we find the Euclidean distances between it and the mean vector of the states. We assign a observation to the closest state for training. The state assigned to consecutive observations are tracked to find the transitional probabilities.

2.2 Segmental K-means Algorithm[18]

This algorithm tries to modify the initial model so as to maximize *P*(*O*, *I*|¸). Where O is the training sets used for training and I is a state sequence in the given HMM. The maximized (optimal) path for a training set is denoted by I*. Those observations that were assigned to a different state then the one in which they should be present according the optimal path are then moved to the state. This improves *P*(*O*, *I* ¤ |¸). The model is evaluated again so with this changed assignments of observations. We do the above process iteratively till there are no more reassignment needed. The calculation of mean, variance, and transitional probabilities are done as shown before.

Process of converting speech to text[18]

Three fundamental issues need to be addressed for text -to-phoneme mapping with staged neural networks: training datasets, window alignment, and context. The 2000 Most Common Words in American English (MCWE) were selected for developing text -to-phoneme staged NNs and performance metrics were estimated for the 5000, 7000 and 10000 MCWE (See References, CMPD web page). The 1000 MCWE were used in order to compare our results with NetTalk (Sejnowski (1987)).

The Carnegie Mellon Pronouncing Dictionary (CMPD) was utilized for generating the windowed training dataset. CMPD is a publicly available web based machine-readable pronunciation dictionary for North-American English that contains over 100,000 words and their phonetic transcriptions. The implemented CMPD phoneme set contained 39 phonemes, for which the vowels may carry lexical stress (0 for no stress, 1 for primary stress, and 2 for secondary stress). The staged networks were trained based on the 2000 MCWE. Window alignment is important because a unique map from text to phonemes is needed to generate the training/test sets for the NNs. Words are sliced starting with the first letter in the window and shifted to the left until the whole word has been passed by. This implies that the number of patterns per word will be equal to number of letters in the word. Bullinaria (1997) considered issues related to the choice for the appropriate window size (Bullinaria considered only central windowing). A first issue relates to the proper choice for the window size in order to accommodate any long-range dependencies. Also, a large window sizes implies that many units and Second Position Central and Third Position connections would be vastly underutilized because of the prevalence of empty window spaces. Different arrangements for central windowing and Second Position Asymmetric Windows (SPAW) alignments were investigated. Sometimes two different phonemes can be mapped from the same information window: this is considered an inconsistency. Looking at Figure 2a, it can be seen that different alignments can lead to inconsistencies between {"CAME", "CAMERA", "BECAME"}, and {"THOUGH", "THOUGHT"}. Only a 2-6 SPAW avoids any inconsistency. The main idea was to explore how much information (spaces) were needed in order to minimize the number of inconsistencies per arrangement. Inconsistencies were counted for different window representations of the objective letter for both central and SPAW alignments

III. Some important Definitions:

3.1 Text-To-Speech Synthesis [TTS][15]

Text-To-Speech Synthesis (TTS) quality has increased by the employment of large corpora and unit-selection techniques. This type of systems, generally called corpus-based TTS, generate synthetic speech by selecting the most appropriated sequence of acoustic units from a speaker-dependent database and then apply a smoothing strategy to join the selected units together. Corpus-based TTS can synthesize only speech having the septic style present in the corpus. Therefore, in order to synthesize other types of speech, e.g. emotional or expressive speech or speech of various speakers, representative speech samples should be recorded in advance. Moreover, the representative speech samples need to be large-sized corpora, in order to maintain the output quality. Speech recording and data processing for a TTS is expensive and time consuming. VC may be a fast and a cheap way to build new voices for a TTS. Thanks to VC, it will be possible to read e-mails or SMS with their sender's voice, to assign our and our friends' voices to characters when playing on a computer game, or to apply deferent voices to deferent computer applications. VC may be also applied to emotional speech synthesis, as an aid to prosodic medications on a neutral sentence.

3.2 Automatic Speech-to-Speech translation[15]

The Voice Conversion technology will be very useful in interpreted telephony, when the translation task requires speaker identication by listeners. For example, in a conference call with more than two participants it is very important to be able to differentiate speakers by their voices. Education. When learning foreign languages, proper

IV. Some uses:[19]

- **Education.** When learning foreign languages, proper intonation of sentences and pronunciation of non-existing phonemes in the native language is one of the most midcult tasks for students. VC may help to learn foreign languages especially in pronunciation exercises, when students would listen to their own voices pronouncing foreign sounds properly.
- **Medical aids.** Another application field of VC is speaking aids for people with speech impairments. Voice transformation systems may be used to improved the intelligibility of abnormal speech uttered by a speaker who has speech organ problems [Hos03]. On the other hand, VC may be also useful for designing hearing aids appropriated for septic hearing problems. VC technology may morph speech signals to other frequency ranges in order to improve the recognition rate.
- **Entertainment.** One of the most obvious applications of VC in the entertainment field is karaoke, where the singer is helped to success in every kind of songs. Other experiments have been done in _lm dubbing and looping 1 [Tur02]. By employing only several dabbers, voices of famous actresses/actors can be generated in any language, and new utterances of actresses/actors who are not alive can be synthesized. Another dubbing application may be to regenerate the voices of actresses/actors who have lost their voice characteristics due to old age. Integrated with 3-D facial animation techniques, VC can be employed to create virtual characters with a desired speaker identity for multiple applications, such as videogames. Moreover, acquiring a high level of knowledge about speaker individuality will help the success of other speech technologies, as speech or speaker recognition tasks.

V. PROPOSED WORK

- Implementation a system which will take input as Hindi speech and recognize the use.
- Recognize a user by his speaking.
- Recognize user uniquely.
- Conversion of speech to text.
- Noise removing from the sound.
- Gathering of the idea about to different part of the sound reorganization.
- Provide greater security through human voice

VI. CONCLUSION

Here I will use a word based acoustic model. This model can be used only for limited vocabulary. We would have to move towards a phone based acoustic model. This problem of lack of good public domain acoustic model for India language needs to be addressed. There are a few public domain speech recognition systems that are available. But they have their own drawback. For example, there is a speech recognition software called ocvolume. The problem with ocvolume is that it does not use HMM, but vector quantization. Thus it will not scale up to many words. Similarly there are helper program and libraries that are available. Example GHMM is a open source HMM library, but it does not support multi-dimensional HMMs. Thus as future work it would be useful if someone could do some work in improving these libraries and software so that they can be integrated together and can become more useful.

7. REFERENCES

[1]. Cmu sphinx - open source speech recognition engines.
[2]. Internet-accessible speech recognition technology. http://www.cavs.msstate.edu/hse/ies/projects/speech/index.html.
[3]. Brent M. Dingle. Calculating determinants of symbolic and numeric matrices. Technical report,Texas A&M University, 2005.
[4]. RakeshDugal and U. B. Desai. A tutorial on hidden markov models. http://uirvli.ai.uiuc.edu/dugad/hmm_tut.html.
[5]. Samudravijaya K and Maria Barot. A comparison of public domain software tools for speech recognition. pages 125–131. Workshop on Spoken Language Processing, 2003.
[6]. L. Lamel and J. Gauvain.High performance speaker-independent phone recognition using cdhmm.EUROSPEECH-93.
[7]. N. Rajput M. Kumar and A. Verma. A large-vocabulary continuous speech recognition system for hindi. IBM Journal for Research and Development.
[8]. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition.pages 256–286. Proceedings of the IEEE, 1989.
[9]. L. R. Rabiner and M. R. Sambur.An algorithm for determining the endpoints of isolated utterances. Bell System Technical Journal, Vol. 54, pages 297-315, 1975,.
[10]. Arciniegas,F., and Embrechts, M., 2000, "Artificial NNs (ANNs) for Phoneme Recognition for Text -to-Speech Applications," *Proceedings of the 2000*
[11]. *IEEE-INNS-ENNS International Joint Conference on NNs*, Como, Italy. Bullinaria, J. A., 1994, "Internal Representations of a Connectionist Model of Reading Aloud," *Proceedings of the Sixteenth Annual Conference of theCognitive Science Society*, pp. 84-89.
[12]. Bullinaria, J. A., 1995, "NN Learning from Ambiguous Training Data," *Connection Science*, Vol. 7, pp. 99-122.
[13]. Bullinaria, J. A., 1997, "Modeling Reading, Spelling, and Past Tense Learning with Artificial NNs," *Brain and Language*, Vol. 59, pp. 236-266.
[14]. Colheart, M., Curtis, B., and Atkins, P., 1933, "Models of Reading Aloud: Dual-Route and Parallel-Distributed-Processing Approaches," *Psychological Review*, Vol. 100, pp. 589-608.
[15]. Patel, M., 1996, "Using Neural Nets to Investigate Lexical Analysis," *Proceedings of PRICAI'96: Topics in Artificial Intelligence*, pp. 241-252.

ISSN: 2454-4248

[16]. Seidenberg M., and McClelland, J., 1989, "A Distributed, Developmental Model of Word Recognition and Naming," *Psychological Review*, Vol. 96, No. 4, pp. 523 - 568.

[17]. Sejnowski, T. J., and Rosenberg, C. R., 1987, "Parallel Network that Learn to Pronounce English Text," *Complex Systems*, Vol. 1, No. 1, pp. 145-168.
www.speech.cs.cmu.edu/cgi-bin: The CMPD Pronouncing Dictionary.

[18]. Carnegie Mellon University.
http://gopher://gopher.sil.org/11/gopher_root/linguistics/info/

[19]. LOB and ACL_DCI corpus frequency list.
http://www.uri.edu/comm_service/cued_speech/1000most.html.