Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.580 – 589

RESEARCH ARTICLE

Advanced Hands Free Computing

S. T. Patil[1], Snehal M. Chavan[2], Nileshwari R. Chaudhari[2], Pranali J. Patil[2]

[1]Professor, Vishwakarma Institute of Technology, Pune Stpatil77@gmail.com

[2] research Scholars, Vishwakarma Institute of Technology, Pune.Chavansnehal2010@gmail.com

[2] research Scholars, Vishwakarma Institute of Technology, Pune. Nileshwari92@gmail.com

[2] research Scholars, Vishwakarma Institute of Technology, Pune. Pranalics@gmail.com

ABSTRACT- ***Speech recognition technology is already available to Higher Education and Further Education as are many of the alternatives to a mouse. In this project we have proposed a new application for hands free computing which uses voice as a major communication mean to assist user in monitoring and computing purpose on his machine. In our project as we have mainly used voice as communication mean. Speech technology encompasses two technologies: Speech Recognition and Speech Synthesis. In this project we have directly used speech engine which uses Hidden Marcov Model and Feature extraction technique as Mel scaled frequency cepstral. The mel scaled frequency cepstral coefficients (MFCCs) derived from Fourier transform and filter bank analysis are perhaps the most widely used front ends in state-of-the-art speech recognition systems. Our aim is to create more and more functionalities which can help human to assist in their daily life and also reduce their efforts. The HMM (Hidden Marcov Model) is used internally in which the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.***

Keywords: ***Hidden Marcov Model; feature extraction; MFCC; speech recognition; speech synthesis; Fourier transform***

I. INTRODUCTION

Research in speech processing and communication for the most part, was motivated by people's desire to build mechanical models to emulate human verbal communication capabilities. Speech is the most natural form of human communication and speech processing has been one of the most exciting areas of the signal processing. Speech recognition technology has made it possible for computer to follow human voice commands and understand human languages. The main goal of speech recognition area is to develop techniques and systems for speech input to machine.

There are a number of disabilities and medical conditions that can result in barriers for those attempting to use a standard computer keyboard or mouse. This does not just include physical disabilities. Many students with reading/writing difficulties such as dyslexia can find using the keyboard to enter text into the computer a laborious exercise that can limit their creativity.

Hands-free computing is a term used to describe a configuration of computers so that they can be used by persons without the use of the hands interfacing with commonly used human interface devices such as the mouse and keyboard.

This application basically combines two technologies: Speech synthesis and Speech recognition. Through Voice Control, the computer uses voice prompts to request input from the operator. The operator is allowed to enter data and to control the software flow by voice command or from the keyboard or mouse. The Voice Control system allows for dynamic specification of a grammar set, or legal set of commands. The use of a reduced grammar set greatly increases recognition accuracy.

Speech Recognition (also known as automatic speech recognition or computer speech recognition) converts spoken words to text. Speech Recognition takes an audio stream as input, and turns it into a Command which is later mapped with an event.

In Speech synthesis text is converted to speech signal. Speech synthesis is also known as text to speech conversion. In this application speech synthesis is used to read mail and for converting text into speech.

In our project we have used The Speech Application Programming Interface or SAPI. It is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. In general all API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages. In addition, it is possible for a 3rd-party company to produce their own Speech Recognition and Text-To-Speech engines or adapt existing engines to work with SAPI. Basically Speech platform consist of an application runtimes that provides speech functionality, an Application Program Interface (API) for managing the runtime and Runtime Languages that enable speech recognition and speech synthesis (text-to-speech or TTS) in specific languages.

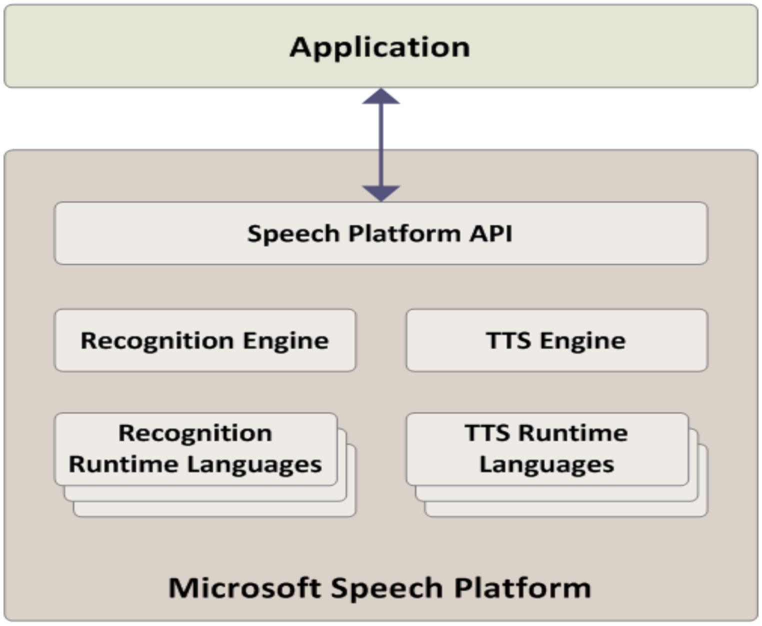


Fig.1: Overview of a Speech Platform.

A .BENEFITS OF USING SYSTEM SPEECH:

1. Microsoft .NET Framework Managed-Code APIs
2. Speech Recognition
3. Speech Synthesis (text-to-speech or TTS)
4. Standards Compatible
5. Cost Efficient.

II. LIMITATIONS IN EXISTING SYSTEM

Noises, distortions, and unforeseen speakers seldom cause difficulty for human to understand speech signals whereas they seriously degrade performances of automatic speech recognition (ASR) systems.

While extracting features from speech, it becomes difficult to recognise correct word due to noise and other environmental conditions. Windows speech recognition is efficient but it is like one way communication. When words are spoken, processing is done and reply is given by performing task or opening application. It is hardware or software response instead of voice. It is necessary to get voice feedback for the command given by user for any user friendly application. In Window Speech API only OS related commands are executed. These commands are helpful, but they are not command to assist in user life to make their life easier. This project adds commands for making device handier. All commands which can be executed by command prompt are included. Windows speech API does not contain hardware commands. We can open Google by voice command but we can’t type our query by voice.

Also there are number of limitations like environment issues due to type of noise, signal/noise ratio, working conditions, transducer issues, channel issues due to Band amplitude, distortion, echo etc., speakers issues due to Speaker dependence/independence, Sex, Age, physical and psychological state, speech style issues due to voice tone(quiet, normal, shouted) etc., production issues due to isolated words or continuous, speech read or spontaneous speech speed(slow, normal, fast), vocabulary issues due to Characteristics of available training data, specific or generic vocabulary and many more which limit the efficiency application.

III. PROPOSED SYSTEM

Speech recognition process can be completed in two parts - front end and a back end.

The front end processes the audio stream, isolating segments of sound that are probably speech and converting them into a series of numeric values that characterize the vocal sounds in the signal. The back end is a specialized search engine that takes the output produced by the front end and searches across three databases.

Following diagram shows the basic architecture of Hands free computing application

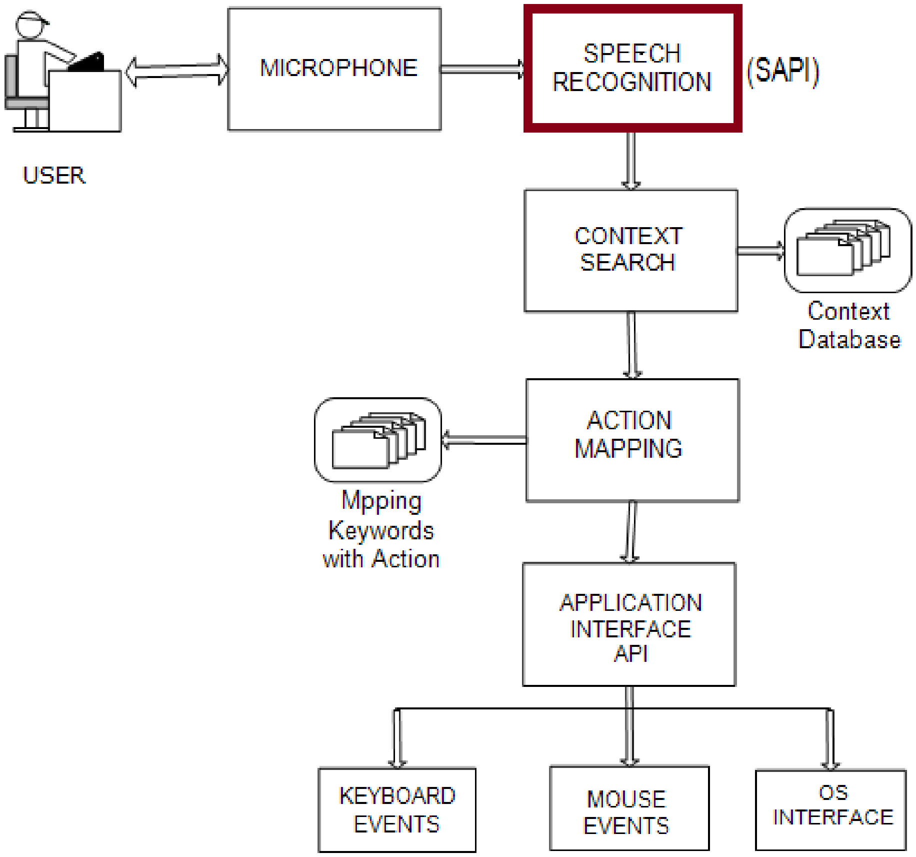


Fig.2: Basic architecture

As user gives the speech signal (simply an audio stream) with the help of microphone. Microphone processes the audio stream to the Speech Recognition system which will convert a speech signal to a sequence of words in form of digital data i.e. a command with the help of SAPI. This command is then searched in context database according to context search. If it matches then further action mapping is done in which actions or response to the specific command is specified. Using application interface APIs like keyboard events, mouse events and OS interface, appropriate action is performed according to given command To perform this whole operation Speech recognition and synthesis is used which we are going to see in detail.

A. HOW SPEECH RECOGNITION WORKS

Speech recognition fundamentally functions as a pipeline that converts PCM (Pulse Code Modulation) digital audio from a sound card into recognized speech. The elements of the pipeline are as follows.

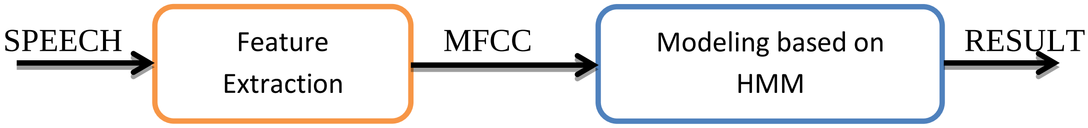


Fig.3: Speech Recognition Pipeline

1) Transform the PCM Digital Audio

The digital audio is a stream of amplitudes, sampled at about 16,000 times per second. To make pattern recognition easier, the PCM digital audio is transformed into the "frequency domain." Transformations are done using a windowed fast-Fourier transform. The fast Fourier transform analyzes every 1/100th of a second and converts the audio data into the frequency domain. Each 1/100th of a second result is a graph of the amplitudes of frequency components, describing the sound heard for that 1/100th of a second. The speech recognizer has a database of several thousand such graphs (called a codebook) that identify different types of sounds the human voice can make. The sound is "identified" by matching it to its closest entry in the codebook, producing a number that describes the sound. This number is called the "feature number."

2) Figure Out Which Phonemes Are Spoken

In an ideal world, you could match each feature number to a phoneme. If a segment of audio resulted in feature #52, it could always mean that the user made an "h" sound. Feature #53 might be an "f" sound, etc. If this were true, it would be easy to figure out what phonemes the user spoke.

Unfortunately, this doesn’t work because of a number of reasons. Every time a user speaks a word it sounds different. The background noise from the microphone and user’s office sometimes causes to recognize different feature number. The sound of a phoneme changes depending on what phonemes surround it. The "t" in "talk" sounds different than the "t" in "attack" and "mist". The background noise and variability problems are solved by allowing a feature number to be used by more than just one phoneme, and using statistical models to figure out which phoneme is spoken.

3) Convert the Phonemes into Words

4) Reducing Computation and Increasing Accuracy

The speech recognizer can now identify what phonemes were spoken. Figuring out what words were spoken should be an easy task. If the user spoke the phonemes, "h eh l oe", then you know they spoke "hello". The recognizer should only have to do a comparison of all the phonemes against a lexicon of pronunciations.

5) Context Free Grammar

One of the techniques to reduce the computation and increase accuracy is called a "Context Free Grammar" (CFG). CFG's work by limiting the vocabulary and syntax structure of speech recognition to only those words and sentences those are applicable to the application's current state. The application specifies the vocabulary and syntax structure in a text file. The speech recognition gets the phonemes for each word by looking the word up in a lexicon. If the word isn't in the lexicon then it predicts the pronunciation.

6) Adaptation

Speech recognition system "adapt" to the user's voice, vocabulary, and speaking style to improve accuracy. A system that has had time enough to adapt to an individual can have one fourth the error rate of a speaker independent system. The recognizer can adapt to the speaker's voice and variations of phoneme pronunciations in a number of ways which are done by weighted averaging.

Following diagram shows how the speech signal is recognized as specific command.

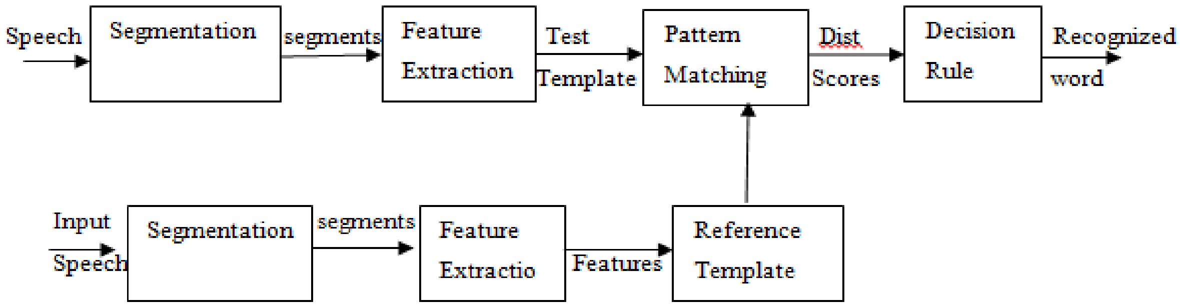


Fig.4: Model for speech recognition

B. HOW SPEECH SYNTHESIZER WORKS.

A speech synthesizer takes text as input and produces an audio stream as output. Speech synthesis is also referred to as text-to-speech (TTS).

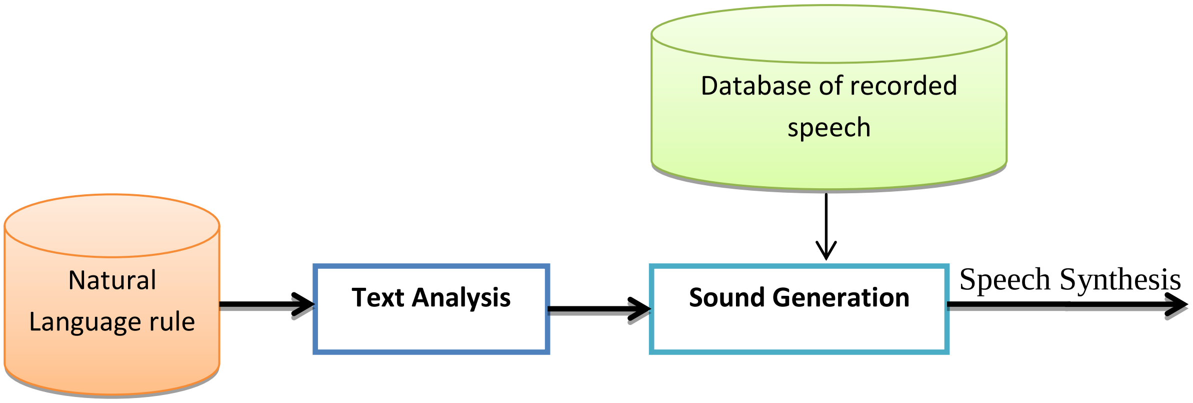


Fig.5: Basic TTS

1) Text Analysis:

The front end specializes in the analysis of text using natural language rules. It analyzes a string of characters to determine where the words are. This front end also figures out grammatical details like functions and parts of speech.

2) Sound Generation:

The back end takes the analysis done by the front end and, through some non-trivial analysis of its own, generates the appropriate sounds for the input text.

C. THE HIDDEN MARKOV MODEL

A. A. Markov first used Markov models to model letter sequences in Russian 161. Such a model might have one state per letter with probabilistic arcs between each state. Each letter would cause (or be produced by) a translation to its corresponding state. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.

Here we have used Hidden Marcov Model as the speech recognition algorithm. In the past few years, the *hidden Markov model* (HMM) formulation has been successfully applied to both isolated word and continuous speech recognition. Part of the reason is due to HMM's ability to capture some of the temporal and spectral variations in the speech signal Template comparison methods of speech recognition. E.g., dynamic time warping directly compare the unknown utterance to known examples. Instead HMM creates stochastic models from known utterances and compares the probability that the unknown utterance was generated by each model. HMMs are a broad class of doubly stochastic models for non stationary signals that can be inserted into other stochastic models to incorporate information from several hierarchical knowledge sources.

In HMM-based map-matching algorithms, words are sequentially generated and evaluated on the basis of their likelihoods. When a new word is encountered, past hypotheses of the solution are extended to account for the new observation. Among all hypotheses in the last stage, the *surviving path* with the highest joint probability is then selected as the final solution. Some of the advantages of using HMM are isolated and continuous word recognition, large vocabulary size. But in HMM training is complex.

IV. RESULTS

The resultant project operates for all control panel commands as well as all function keys, accelerator keys (combination of 2 or more shortcut keys). Gmail shortcuts can be executed. Many of the hardware commands can be operated. User can see recognised command in a textbox. Also user can dictate words in a text file using notepad. A feature named 'Speech pad' is provided. Using speech pad user can read any text files wherever it is stored. This feature is completely voice operated. Voice chat can be done with PCs connected to each other.

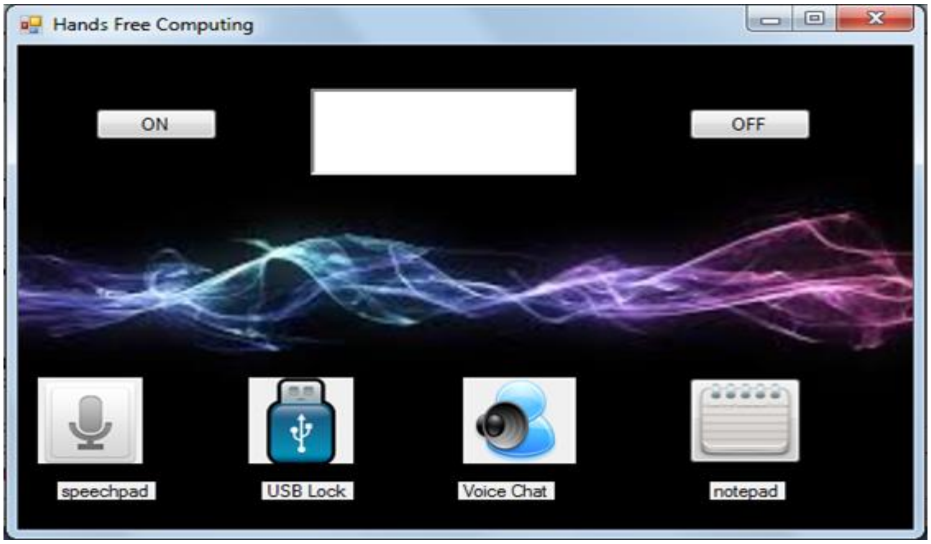
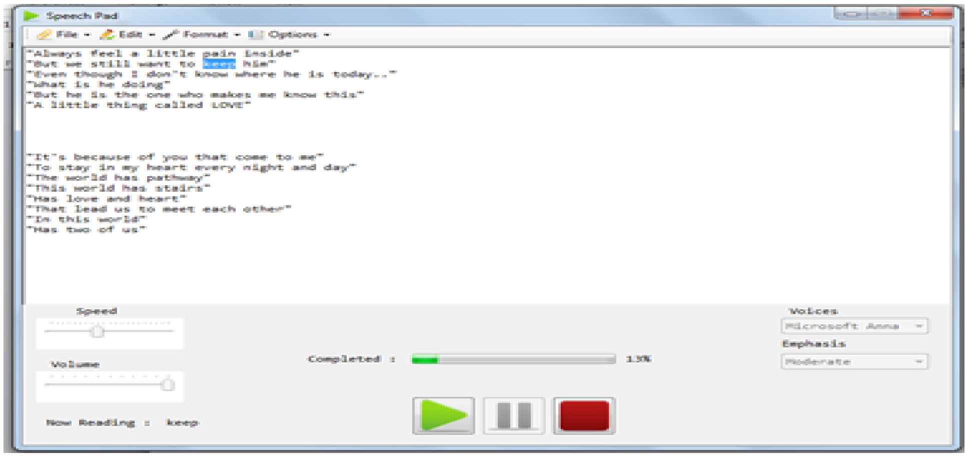


Fig.6: Form for speech recognition

Above Fig.6 shows initial GUI of our project. When user says “online jarvis” or clicks button ‘ON’, it starts recognising commands which are printed in rich textbox also so that 1 can get idea of what computer is recognising.



"Always feel a little pain inside"
"But we still want to keep him"
"Even though I don't know where he is today.."
"What is he doing"
"But he is the one who makes me know this"
"A little thing called LOVE"

"It's because of you that come to me"
"To stay in my heart every night and day"
"The world has pathway"
"This world has stairs"
"Has love and heart"
"That lead us to meet each other"
"In this world"
"Has two of us"

Voices: Microsoft Anna
Emphasis: Moderate
Completed : 13%
Now Reading : keep

Fig.7: Speech Pad

Fig.7 shows feature speech pad. User can open, save and create any text file. Functions are provided to start, pause and abort reading. This read file can also be saved as .wav file or .txt file.

V. CONCLUSION

This paper gives brief idea about ‘Hand Free Computing Application’ which will help disabled users by eliminating the use of keyboard and mouse in most of the applications. Likewise disabled persons may find hands-free computing important in their everyday lives.

REFERENCES

[1] Dr.E.Chandra, A.Akila, “An Overview of Speech Recognition and Speech Synthesis Algorithms”, Dr. E Chandra et al , Int. J. Computer Technology & Applications, Vol. 3 (4), 1426-1430

[2] Dirk Schnelle-Walka, Stefan Radomski, “An API for Voice User Interfaces in Pervasive Environments”

[3] M.A.Anusuya, S.K.Katti, “Speech Recognition by Machine: A Review”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 6, No. 3, 2009

[4] D.B. Paul, “.Speech Recognition Using Hidden Markov Models”

[5] Farzad Hosseinzadeh, Mehrdad Zarafshan, “Designing a system for the recognition of words correct pronunciation by using fuzzy algorithms and multi–layer average method”

[6] Lawrence Rabiner, B H Juang, Biing Hwang Juang, `Fundamentals of Speech Recognition',(Prentice Hall,Singapore) , ISBN: 0130151572

[7] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi, "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", Journal Of Computing, Volume 2, Issue 3, March 2010, ISSN 2151-9617 https://sites.google.com/site/journalofcomputing

[8] Sirko Molau, Michael Pitz, Ralf Schl¨uter, and Hermann Ney, "Computing Mel-Frequency Cepstral Coefficients On The Power Spectrum"

[9] Keh-Yih Su, Member, IEEE, and Chin-Hui Lee, Senior Member, IEEE "Speech Recognition Using Weighted HMM and Subspace Projection Approaches"

[10] Michael Dunn. "Speech synthesis and recognition in .NET - Give applications a voice". Redmond Developer News. Retrieved 2011-11-09.

[11] Dr. Shaila D. Apte, " Speech and Audio Processing ",Wiley India Publication, Feb 2012, ISBN13 : 9788126534081

[12] Arti V. Jadhav and Rupali V.Pawar, "Review of Various Approaches towards Speech

[13] Recognition" 2012 International Conference on Biomedical Engineering (ICoBE),27-28 February 2012,Penang ISBN:978-1-4577-1990-5

[14] Microsoft Corporation. "SAPI System Requirements". MSDN. Retrieved 2006-04-12.