International Journal of Aquatic Science
ISSN: 2008-8019
Vol 14, Issue 01, 2023

Voice Automated System

Vivek Sahare[1], Rajkumar Rathod[2], Sakshi Nile[3], Sakshi Nimbalkar[4], Nikita Onkar[5], Rupali Sawant[6]

[1,2,3,4,5]*Student, Jagadamba College of Engineering Yavatmal, Maharashtra, India*
[6]*Assistant Professor, Jagadamba College of Engineering Yavatmal, Maharashtra, India*

Abstract: In this paper we propose a new product "Voice Automated System". It can be used to operate the entire app functions on the user's voice commands. It makes use of the Speech Recognition technology that allows the app system to identify and recognize words spoken by a human using a microphone. This Software will be able to recognize your spoken words and enable user to interact with his app. This interaction includes user giving commands to his app which will then respond by performing several tasks, actions or operations depending on the commands you gave. It can also help user to dictate and app would convert their spoken words into text for word processing application or E-mail. For Example: Opening /closing a file, starting an application, Mouse click or movement.

Keywords: Speech Technology, Voice Response System, Voice Commands, Human-App Verbal Interaction.

1. INTRODUCTION

Speech Recognition is a technology which helps app to understand the words spoken by its user. The aim of this technology is to be able to understand the words spoken by user and provide human-verbal interaction.

Even after years of research in this area, software using speech recognition still cannot recognize user's speech with complete accuracy. There are few applications which can recognize over 80% of words and only when spoken under some specific conditions. Hence, there are only limited uses of this technology.

Our product uses this technology to provide a naïve user the ability to interact with app. Since speech is man's most effective form of communication, then why not use it to communicate with the app? Our Software makes that happen by allowing you to have conversations with the app.

This conversation involves you giving some commands and the app responding by some actions or operations based on the commands.
Our product includes the following characteristics:

- Multi user voice recognition
- Better accuracy as compared to other existing systems
- Works for game applications
- Comprises of an enhanced dictionary with more than 1500 words
- Comprises of 1000 plus commands
- Works for web applications-Facebook

This System also helps physically challenged users who cannot use the keyboard and mouse. For example: Users without hands, or without eye sight. Control Mouse and App System Using Voice Commands allow dictation as well as control of many app tasks.

Literature Survey

The earliest attempts in speech recognition were made during 1950 and 1960s. In 1952, at Bell Laboratories, Davis, Biddulph, and Balashek built an isolated digit recognition system for a single speaker using the formant frequencies measured/estimated during vowel regions of each digit.

In 1956 at RCA Laboratories, Olson and Belar tried to recognize 10 distinct syllables of a single speaker, as embodied in 10 monosyllabic words [10]. In 1959, at University College in England, Fry and Denes tried to build a phoneme recognizer to recognize four vowels and nine consonants. They used spectrum analyser and pattern matcher for the recognition. By incorporating statistical information, they increased the overall phoneme recognition accuracy for words consisting of two or more phonemes. Their work marked the first use of statistical syntax in automatic speech recognition.

In 1960s, Martin and his colleagues at RCA Laboratories developed a set of elementary time normalization methods to detect speech starts and ends that significantly reduced the variability of the recognition scores. At the same time, in the Soviet Union, Vintsyuk proposed the use of dynamic programming methods generally called dynamic time warping for time aligning a pair of speech utterances including algorithms for connected word recognition.

In 1970, the area of isolated word or discrete utterances became a viable and usable technology based on the studies in Russia and Japan. The Itakura of Bell laboratories introduced that through the use of an appropriate distance measure based on LPC spectral parameters, linear predictive coding (LPC) could be used in speech recognition. Also researchers here, started experiments aiming at making speaker independent systems. A wide range of clustering algorithms was used to achieve this goal. In 1973, Hearsay I system by CMU was able to use semantic information to significantly reduce the number of alternatives considered by the recognizer. CMU's Harpy system was able to recognize speech using vocabulary of 1011 words with reasonable accuracy. These projects were funded by DARPA (Defence Advanced Research Projects Agency).

In 1980, there was a shift in methodology from template based to more rigorous statistical modelling framework. One of the key technologies was Hidden Markov Model (HMM) although this technique became widely applied in mid-1980s. Furui proposed the use of cepstral coefficients as spectral features in speech recognition. The n-gram model defining the probability of occurrence of an ordered sequence of n words was introduced by IBM for large vocabulary speech recognition systems. The primary focus was the development of a language model which describes how likely a sequence of language symbols appear in a speech signal.

In 1990's DARPA program was continued. The emphasis was laid on the different speech understanding application areas such as transcriptions of broadcast news and conversational speech. The BN transcription technology was integrated with information extraction and retrieval technology, and many application systems, such as automatic voice document indexing and retrieval systems, were developed [12]. Various other techniques were developed viz. the maximum likelihood linear regression (MLLR), the model decomposition, parallel model composition (PMC) , and the structural maximum a posteriori (SMAP)

method to reduce the mismatch caused by background noise , microphones , voice individuality etc.
Human-app interfaces facilitate communication, assist on the exchange of information, process commands and controls and perform several additional interactions. Spoken natural language is more user-friendly mean of interacting with a app. From the human perspective point, this kind of interaction is easier since it does not urge humans to learn additional interactions. Humans can rely on natural ways of communications instead.
Human-app interaction varies from understanding simple commands to extracting all the information in the speech signal such as words, meanings and emotions of the user. To develop an interface with natural language understanding ability, several factors arise and must be taken into account such as dealing with the ungrammatical nature of many spoken utterances, the detection of problems in speech recognition, and the design of intelligent clarification dialogues
There are also verbal interaction problems such as background noise, word echoing and repetition, and different and background sound sources overlapping the speech.
Solution for these problems must be developed.
Most state-of-the-art of the human-app verbal interaction based systems relies on using simple voice commands, hence may not be considered as complete speech systems. Having these concepts in mind and addressing human-app verbal issues, the speech recognition products should be well developed and be resilient enough for effective human-app verbal interaction.
The system we are proposing here consists of providing an augmented speech dialogue structure used as and tested to study verbal human-app interaction (user inputs app responds to the command by performing the desired task).

Proposed System
The principle aim of our project is to create a user independent automatic speech recognizer with an altered grammar unit.
The system will be able to retrieve folders, sub-folders, documents & other application/software using voice commands. Our system can also launch any file or macro. Dictation of texts using various products like Microsoft Office, Notepad & other Text editors is also made possible through our system.
Our software helps open websites, documents, or programs. Simulation of keystrokes and running of any file is also made possible. By voice commands we can enable shutdown, restart or log off operations on the app.
It is a thrilling technology that will change the way app user's interaction with the app. User can speak to his app and it will respond by performing the tasks ordered by the user. The speech that user and his app interact with is scripted. In other words, user can talk to his app using a set of predefined commands and instructions given in the grammar (i.e., a script). For example, you can say: "Mouse Left" and the app would respond by moving the mouse cursor left. Or you can say: "My App" and the app would open the "My App". As for the programming language we used Microsoft's Visual Basic.NET. Also we are going to make use of Microsoft Speech Application Programming Interface-SDK. We will try to take minimum required inputs from the user & will give the desired output in the system.

Implementation
Voice Automated System takes user's voice as input.
Brief Summary of System Flowchart:

- User gives voice command to system.
- It is recognized by intermediate tool SAPI.
- The SAPI converts analog signal into its digital representation.
- System will search in its Grammar File for matching respective command.
- Based on the match it will execute operation (For example: User Speaks “Open MS Word”. The system will start MS Word.).

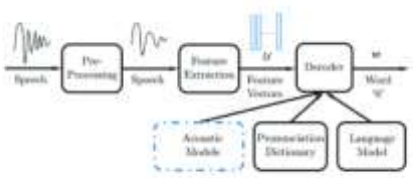


Fig: Automated Speech Recognition System

Feature Extraction

First of all, recording of various speech samples of each word of the vocabulary is done by different speakers. After the speech samples are collected, they are converted from analog to digital form by sampling at a frequency of 16 kHz. Sampling means recording the speech signals at a regular interval. The collected data is now quantized if required to eliminate noise in speech samples. The collected speech samples are then passed through the feature extraction, feature training & feature testing stages. Feature extraction transforms the incoming sound into an internal representation such that it is possible to reconstruct the original signal from it. There are various techniques to extract features like MFCC, PLP, RAST, LPCC, but mostly used is MFCC.

Mel Frequency Cepstral Coefficients MFCCs are used because it is designed using the knowledge of human auditory system and is used in every state of speech recognition system or art speech. MFCC is a standard method for feature extraction in speech recognition tasks. MFCC include certain steps applied on an input speech signal. These computational steps of MFCC include: - Framing, Windowing, DFTH, Mel filter bank algorithm, computing the inverse of DFT.

Decoding

It is the most important step in the speech recognition process. Decoding is performed for finding the best match for the incoming feature vectors using the knowledge base. A decoder performs the actual decision about recognition of a speech utterance by combining and optimizing the information conveyed by the acoustic and language models.

Acoustic

Modelling There are two kinds of acoustic models i.e. word model and phoneme model. An acoustic model is implemented using different approaches such as HMM, ANNs, dynamic Bayesian networks (DBN), support vector machines (SVM). HMM is used in some form or the other in every state of the art speech and speech recognition system.

Hidden Markov Model

A hidden Markov model (HMM) is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. We call the observed event a `symbol' and the invisible factor underlying the observation a `state'.

Language Modelling

Language models are used to guide the search correct word sequence by predicting the likelihood of nth word using (n-1) preceding words. Language models can be classified into:

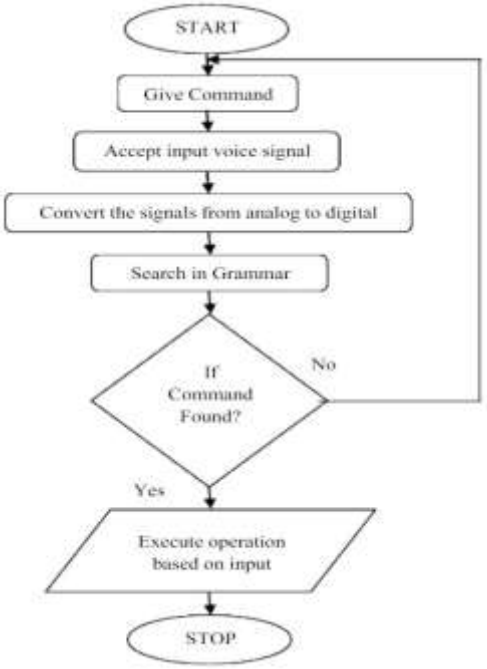
- Uniform model: each word has equal probability of occurrence.
- Stochastic model: probability of occurrence of a word depends on the word preceding it.
- Finite state languages: languages use a finite3 state network to define the allowed word sequences.
- Context free grammar: It can be used to encode which kind of sentences is allowed.

Pronunciation Modelling

In pronunciation modelling, during recognition, the sequence of symbols generated by acoustic model HMM is compared with the set of words present in dictionary to produce sequence of words that is the system's final output contains information about which words are known to the system and how these words are pronounced i.e. what is their phonetic representation. Decoder is then used for recognizing words by combining and optimizing the information of acoustic & language models.

Flowchart and Algorithm

Flowchart



Algorithm

- Start
- Give Instructions (Speak Command).
- Accept the voice input through sound card.

- Convert the voice analog signal into digital signal. Search the Grammar to identify whether signal (command) is matching or not.
- If the command not found go to step 2.
- Execute operations based on the match found from grammar (instructions retrieved from matched Value).
- Stop

Voice Automated System

A. Speech recognition system can be developed for the grammatical structure and some statistical model can be used to improve word predication, but still there is a problem that how much world knowledge of speaking and encyclopedia can be modeled? Of course, we cannot model the world knowledge. So we cannot measure app system up to human comprehensive.
B. Only speech does not participate in human communication, even some body signals are also used such as hand waving, eye moment and others. Consequently in any ASR system such information is completely missed
C. Any unwanted information in any sound signal is a noise. While speaking in any environment, a radio playing somewhere downs the corridor, a clock ticking, another human speaker in the background are all examples of noise. ASR should be intelligent enough to detect such noise and filter it out from the speech signal.
D. Written language and spoken language are essentially different in nature. Written language is one way communication while spoken language is dialog oriented. In spoken language we give feed back to the sound that we understand. So in last few years it has been observed that spoken language is grammatically less complex whereas in written language, grammatical possibilities should always be kept in mind. As normally speech contains repetitions, slips of the tongue, changes of subject in the middle of phrase, hesitations etc. such disfluencies are commonly ignored by human listener. In ASR, such kind of behaviour should be represented by the machine and these differences should be identified and addressed carefully.
E. Communication does not have natural pause between words of a spoken sentence, usually pauses comes at the beginning and end of a speech. ASR should be capable to convert a sound wave into a sequence of spoken words.
F. All persons in this world have their special voices; because of their distinctive physical body. There are some variations, even within a one specific speaker, listed below.

If a person speaks same word again and again then there will definite be a small variation in same spoken word. The realization of sound changes over time.

All humans speak according to place and their emotions. For example a person speaks differently with parents, with friends, with teachers, in banks, in market, same as speaking style also varies on expressions. We speak differently when we are happy, sad, stressed, frustrated, disappointed etc

Different aged persons have different speaking style and different speaking sound. Some persons have different speaking sound at different ages.

Output of the Program

Fig 7.1 : Home Page

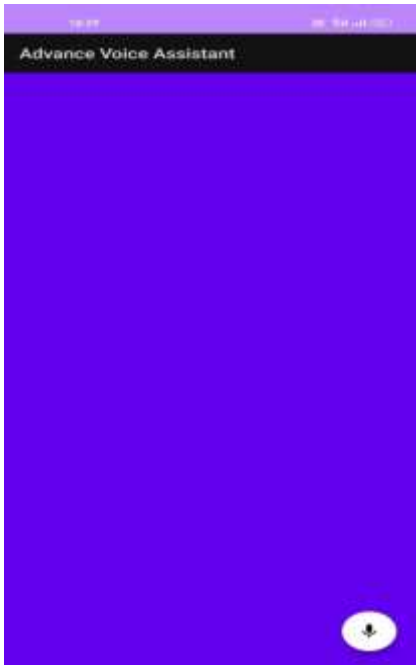


Fig 7.2 : Command Page

COMMAND'S

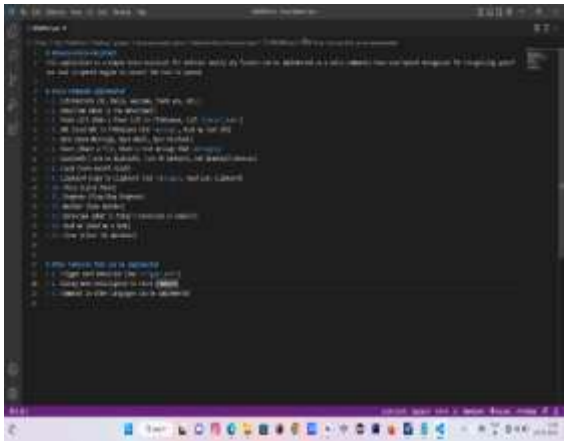


Fig : Command's of the System

Future Work

Today, when we call most telephone line providers, ISPs, or customer care, we can hear an automated voice recording which instructs you to press buttons to move through some option menus. There are some companies which provide similar system, but instead of you pressing the buttons you just have to speak to get what you need. The system used in these companies is a also type of speech recognition program .

Our product is made available for use at home also apart from business uses. 'Control Mouse and App System Using Voice Commands' allow user to dictate and app recognize his spoken words and converts it to text in a word processing application or e-mail document. User can also give some specific commands to perform app operations, such as opening files and accessing menus, and also mouse related events.

In future we would like to enhance our project in the best possible ways. We plan to add more commands. Enhancement of our dictionary with more words is also our next objective. At present our system works with Windows. However, we plan to enable multiplatform support with Linux and Mac OS. We also would like to enable multi language support. Besides, main objective is to improve the accuracy of the system and create full web support.

CONCLUSION

The 'Control Mouse and App System Using Voice Commands' is built with the aim of making easy the operations of a app system with the help of voice. This system may be useful for a regular user. However since a regular user is accustomed to handling the app system with the help of a keyboard and a mouse, this system prove to be useful to the physically handicapped individuals who may not be able to operate the app. Thus the current 'Control Mouse and App System Using Voice Commands developed fulfils this social cause of reaching out to the physically handicapped and ensuring that they too can avail of the functionalities of a app system.

Our project thus making use of the programming language Visual Basic.NET and Microsoft Speech Application Programming Interface SDK- the speech application programming interface, helps create a completely functional voice driven operating system. We hope this system eases out the complexity of using hardware to operate app systems and makes it easier to operate them with voice.

Accuracy will become better and better. Dictation speech recognition will gradually become accepted.
Microphone and sound system will be designed to adapt more quickly to changing background noise level, different environments, with better recognition of extraneous material to be discarded.

REFERENCES

[1] Basanta, H., Huang, Y.P., Lee, T.T., Using voice and gesture to control living space for the elderly people, 2017 International Conference on System Science and Engineering (ICSSE). IEEE, pp. 2023.

[2] P. K. O'Neill. V. Levrukhin, S. Majumdar, V. Norooz, Y. Zang et al. "SPOISpeech 5,500 h of unscribed financial audio for tilly formedend-to-end speech recognition," subeuned to INTERSPEECH, 2021

[3] Mane, M. A., Pol, M P., Patil, M. A., and Patil, M;. IOT based Advanced Home Automation using Node MCU controller and Blynk App.; 13th Intl. Conf. on Recent Innovations in Science, Engineering and Management, Feb. 2018, pp. 178-183.