AN INTELLIGENT VIRTUAL ASSISTANT USING RASPBERRY PI

G. Ashwini[1], M. Nithish Reddy[2], R. Paramesh[3], P. Akhil[4]
Department of Electronics and Communication Engineering, Anurag Group of Institutions

Abstract

The purpose of this paper is to illustrate the implementation of a Voice Command System as an Intelligent Virtual Assistant (IVA) that can perform numerous tasks or services for an individual. These tasks or services are based on user input, location awareness, and the ability to access information from a variety of online sources (such as weather or traffic conditions, news, stock prices, user schedules, retail prices, telling time, local traffic, travel assistant, events, notification from social applications plus one can ask questions to the system, invoke its machine learning otherwise get it from Wikipedia… etc…..).

Using Raspberry Pi as a main hardware to implement this model which works on the primary input of a user's voice. Using voice as an input to convert into text using a speech to text engine. The text hence produced was used for query processing and fetching relevant information. When the information was fetched, it then be converted to speech using text to speech conversion and the relevant output to the user was given. Additionally, some extra modules were also implemented which worked on the concept of keyword matching.

It can help the visually impaired to connect with the world by giving them access to Wikipedia, Calculator etc., all through their voice. This model can also keep people secure as it can be used as a surveillance system which captures the voice of the person standing at the door and similarity checking.

Also it can be a source of entertainment and information for blind/visually impaired. This model will interact with other systems by means of IOT, thus provides a fully automated system. Many experiments and results were accomplished and documented.

Keywords: Virtual Assistant, Text to speech, Speech to text, Raspberry Pi, Voice Command System, Query Processing Machine Learning.

I.Introduction

A Voice Command System essentially means a system that processes voice as an input, decodes or understands the meaning of that input processes it and generates an appropriate voice output. Any voice command system need three basic components which are speech to text converter, query processor and a text to speech converter. Voice has been a very integral part of communication nowadays. Since, it is faster to process sound and voices than to process written text, hence voice command systems are omnipresent in computer devices.

There have been some very good innovations in the field of speech recognition. Some of the latest innovations have been due to the improvements and high usage of big data and deep learning in this field. These innovations have attributed to the technology industry using deep learning methods in making and using some of the speech recognition systems, Google was able to reduce word error rate by 6% to 10% relative, for the system that had the word error rate of 17% to 52%. Text to speech conversion is the process of converting a machine recognized text into any language which could be identified by a speaker when the text is read out loud. It is two step processes which is divided into front end and back end. First part is responsible for converting numbers and abbreviations to a written word format. This is also referred to as normalization of text. Second part involves the signal to be processed into an understandable one.

Speech Recognition is the ability of machine for instance a computer to understand words and sentences spoken in any language .These words or sentences are then converted to a format that could be understood by the machine. Speech recognition is basically implemented using vocabulary systems. A speech recognition system may be a Small Vocabulary- many user system or a Large Vocabulary- small user system.

II.System Architecture

Existing Systems

The existing systems suffer from the drawback that only predefined voices are possible and it can store only limited voices. Hence, the user can't get the full information coherently.

Proposed System

The proposed system is such that it can overcome the drawback of the existing system. The project design involve text to speech. Here whatever the system receives as input after the command the output will get in the form of voice means speech.

Hardware Implementation

Microphone is used to take the audio input of the sound. This audio input when further passed through the system would be searched for keywords. These keywords are essential for the functioning of the voice command system as our modules work on the essence of searching for keywords and giving output by matching.

Raspberry Pi is the heart of the voice command system as it is involved in every step of processing data to connecting components together. The Raspbian OS is mounted onto the SD card which is then loaded in the card slot to provide a functioning operating system.

The Raspberry Pi needs a constant 5V, 2.1 mA power supply. This can either be provided through an AC supply using a micro USB charger or through a power bank.

Internet is being used to provide internet connection to the voice command system. Since the system relies on online text to speech conversion, online query processing and online speech to text conversion hence we need a constant connection to achieve all this.

Speakers, once the query put forward by the user has been processed, the text output of that query is converted to speech using the online text to speech converter. Now this speech which is the audio output is sent to the user using the speakers which are running on audio out.

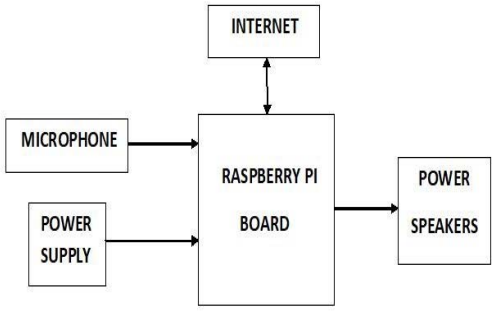

Fig: Block diagram

Flow of Events in Voice Command System

First, when the user starts the system, he uses a microphone to send in the input. Basically, what it does is that it takes sound input from the user and it is fed to the computer to process it further. Then, that sound input if fed to the speech to text converter, which converts audio input to text output which is recognizable by the computer and can also be processed by it.

Then that text is parsed and searched for keywords. Our voice command system is built around the system of keywords where it searches the text for key words to match. And once key words are matched then it gives the relevant output.

This output is in the form of text. This is then converted to speech output using a text to speech converter which involves using an optical character recognition system. OCR categorizes and identifies the text and then the text to speech engine converts it to the audio output. This output is transmitted via the speakers which are connected to the audio jack of the raspberry pi.

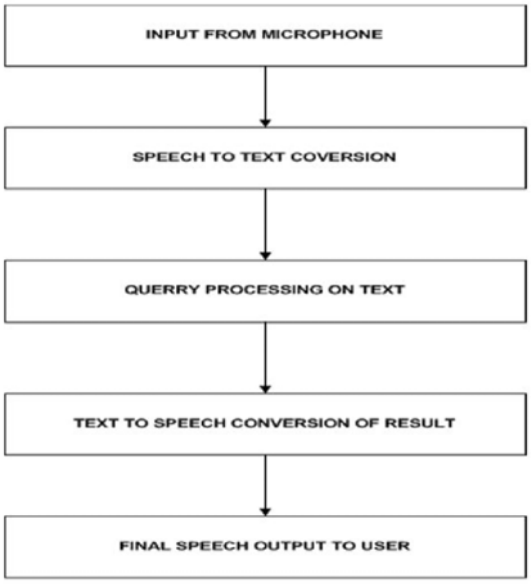

Fig: Flow of events in voice command system

Speech To Text Engine

As the name suggests, the STT engine converts the user's speech into a text string that can be processed by the logic engine. This involves recording the user's voice, capturing the words from the recording (cancelling any noise and fixing distortion in the process), and then using natural language processing (NLP) to convert the recording to a text string.

Text To Speech Engine

This component receives the output from Virtual Assistant's logic engine (query processing system) and converts the string to speech to complete the interaction with the user. TTS is crucial for making the Virtual Assistant more humane, compared to giving confirmation via text.

Query Processor

The Voice Command System has a module for query processing which works in general like many query processors do. That means, taking the input from the users, searching for relevant outputs and then presenting the user with the appropriate output. In this system we are using the site wolfram alpha as the source for implementing query processing in the system. The queries that can be passed to this module include retrieving information about famous personalities, simple mathematical calculations, description of any general object etc.

III. Result

The Voice Command System works on the idea and the logic it was designed with. Our virtual assistant uses the hotword to take a command. Each of the commands given to it is matched with the names of the modules written in the program code. If the name of the command matches with any set of keywords, then those set of actions are performed by the Voice Command System. The modules are based upon API calling. We have used open source text to speech and speech to text converters which provide us the features of customizability. If the system is unable to match any of the said commands with the provided keywords for each command, then the system apologizes for not able to perform the said task. All in all, the system works on the expected lines with all the features that were initially proposed. Additionally, the system also provides enough promise for the future as it is highly customizable and new modules can be added any time without disturbing the working of current modules.

IV.Conclusion and Future work

In this paper, introduced the idea and rationale behind the Voice Command System, the flaws in the current system and the way of resolving those flaws and laid out the system architecture of the presented Voice Command System. Many modules are of open source systems and have customized those modules according to the presented system. This helps get the best performance from the system in terms of space time complexity.

The Voice Command System has an enormous scope in the future. Like Siri, Google Now and Cortana become popular in the mobile industry. This makes the transition smooth to a complete voice command system. Additionally, this also paves way for a Connected Home using Internet of Things, voice command system and computer vision.

References

[1] Dahl, George E., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." Audio, Speech, and Language Processing, IEEE Transactions on 20.1 (2012): 30-42.

[2] Chelba, Ciprian, et al. "Large scale language modeling in automatic speech

recognition." arXiv preprint arXiv:1210.8440 (2012).

[3] Schultz, Tanja, Ngoc Thang Vu, and Tim Schlippe. "GlobalPhone: A multilingual text & speech database in 20 languages." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[4] Tokuda, Keiichi, et al. "Speech synthesis based on hidden Markov models."Proceedings of the IEEE 101.5 (2013): 1234-1252

[5] Singh, Bhupinder, Neha Kapur, and Puneet Kaur. "Speech recognition with hidden Markov model: a review." International Journal of Advanced Research in Computer Science and Software Engineering 2.3 (2012).

[6] Lamere, Paul, et al. "The CMU SPHINX-4 speech recognition system." IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong. Vol. 1. 2003.

[7] Black, Alan W., and Kevin A. Lenzo. "Flite: a small fast run-time synthesis engine." 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis. 2001.