SPA: Smart Personal Assistant

Kaushik C Nagarajan
Department of Information Technology,
Vidyalankar Institute of Technology, Mumbai, Maharashtra

Shriraj Desai
Department of Information Technology,
Vidyalankar Institute of Technology, Mumbai, Maharashtra

Sarvesh Shinde
Department of Information Technology,
Vidyalankar Institute of Technology, Mumbai, Maharashtra

ABSTRACT

This paper discusses ways in which new technology could be harnessed to create an intelligent Voice Controlled Virtual Personal Assistant (VCVPA) with a focus on user-based information. In the Modern Era of fast moving technology we can do things which we never thought we could do before but, to achieve and accomplish these thoughts there is a need for a platform which can automate all our tasks with ease and comfort. This engages the ability to communicate socially through natural language processing, holding (and analyzing) information within the context of the user. Thus we need to develop a Personal Assistant having brilliant powers of deduction and the ability to interact with the surroundings just by one of the materialistic form of human interaction i.e. 'HUMAN VOICE'. The Hardware device captures the audio request through microphone and processes the request so that the device can respond to the individual using in-built speaker module.

Keywords

Smart Personal Assistant, Natural language processing,

1. INTRODUCTION

An intelligent virtual assistant (IVA) or intelligent personal assistant (IPA) is a software agent that can perform tasks or services for an individual based on commands or questions. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands.

An easy access to machine with voice commands is the revolutionary way of human-system interaction. To achieve this, we need to use speech to text API for understanding the input. Many companies like Google, Amazon and Apple are trying to achieve this in generalized form. Isn't it amazing that you can set reminders by just saying remind me to, or set alarm with wake me up. Understanding the importance of this we have decided to make a system that can be placed anywhere in vicinity and you can ask it to help you do anything for you just by speaking with it. This device can be very handy for day to day use and it can help you function better by constantly giving you reminders and updates.

2. SCOPE

Virtual assistants are quickly evolving to provide more capabilities and value to users. As speech recognition and natural language processing advances, so too will a virtual assistant's ability to understand and perform requests. And as voice recognition technology improves, virtual assistant use will move deeper into business workflows. Tomorrow's virtual assistants will be built with more advanced cognitive computing technologies, which will allow a virtual assistant to understand and carry out multistep requests and perform more complex tasks, such as making a plane reservation.

3. SYSTEM DESIGN

In the first phase, the data is collected in the form of speech and stored as an input for the next phase for processing. In second phase, the input voice is continuously processed and converted to text using STT[Speech to Text]. In next phase the converted text is analyzed and processed using Python Script to identify the response to be taken against the command. Finally once the response is identified, output is generated from simple text to speech conversion using TTS[Text to Speech]. The basic architecture of the personal assistant is shown in above figure.
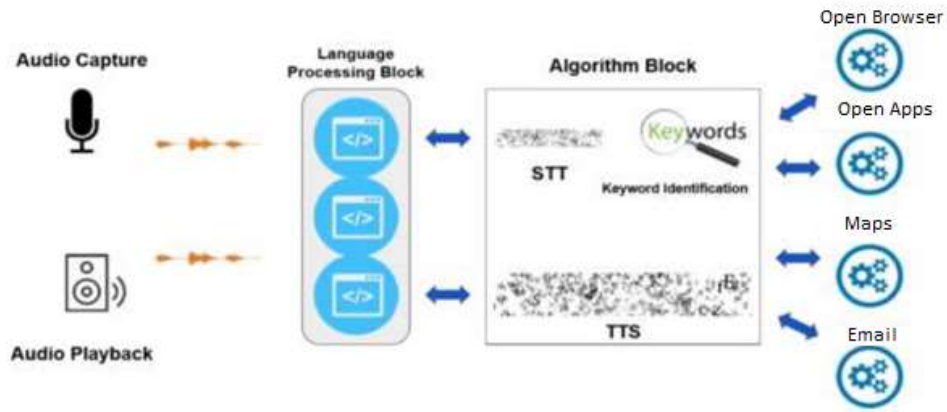
Fig 1: System architecture

4. MATHEMATICAL REPRESENTATION

Input Set

The personal assistant takes voice input, text input and text input through remote signal. So the set of inputs will be,

I_1 = {predefined command, fixed pattern sentential command, random sentences as command}

I_2 = {voice, text, remote text}

$$\text{Thus, } I = I_1 \cup I_2 \qquad (1)$$

Input I = {all sentences in English via speech, all sentences in English as text, remote text input}

Output Set

The outputs for the desired inputs are response determined by the system according to the input given and the database containing all the necessary inputs and their respective outputs.
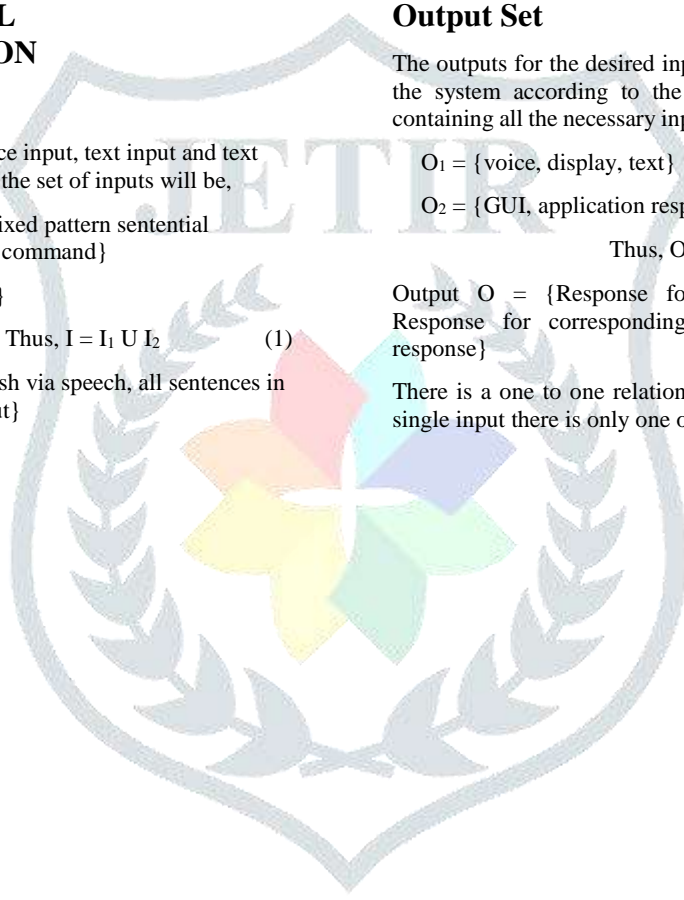
O_1 = {voice, display, text}

O_2 = {GUI, application response}

$$\text{Thus, } O = O_1 \cup O_2 \qquad (2)$$

Output O = {Response for corresponding voice input, Response for corresponding input via GUI, application response}

There is a one to one relation between input and output. For single input there is only one output.

Functions

Following are the operations performed on the input in the system:

Recognize( )

This operation basically gets the input from the user. For text, the input is saved directly into the database while in case of remote signal, the signal is directly converted to the command. For speech input, it converts the voice into text and saves it.

Extract( )

This operation analyses the input string saved and system gets an idea about which command is expected to be executed for the respective input.

Search( )

This operation searches the local database for the response of the command extracted by the previous operation.

Response( )

This operation gives the output that we see on the screen or through speech for the given command.

5. IMPEMENTATION

Following is the list of features implemented:

1. Access specific file
2. News alerts
3. Sets alarm
4. Displays date and time
5. Open applications
6. Locks the workstation
7. Power-off the system
8. Hibernates the system
9. Sleep your machine
10. Play typical song or video
11. Weather forecast
12. Know definition of words
13. Search information on Google or Wikipedia

Let's see the implementation of above tasks-

Weather Forecast

For weather reports, we have used an API called W-underground. By registering as a developer on the w-underground web site we get an API-key, which is used in the request that the executable sends to the API. The executable gets the weather information in xml format.

http://api.wunderground.com/api/"+wunderground_ key + "/conditions/q/autoip.xml

This xml file detects your location and gives the weather information of your location only.

If we want to get the information of specific city or place then we can use following link,
http://api.wunderground.com/api/"+wunderground_ key + "/conditions/q/" + city_name + ".xml"

The xml from API contains information like-
Place
Observation time
Type of date like weather of forecast etc.
Temperature
Wind
Pressure
Dew point

Visibility km
Latitude Longitude and many more.

Locking, Hibernation, Shutting Down and Rebooting the System

For these different system calls are used directly from the application these calls are given and the calls are as follows,

Shut Down

shutdown -s -t 10

Lockdown

Rundll32.exe User32.dll

Hibernate

Rundll32.exePowrProf.dll, SetSuspendState

Remote Input

This is the main feature of this system. We can give input to the assistant by using remote module. Input to the assistant is through internet and LAN or Wi- Fi. For LAN or Wi-Fi, we have created a local Apache web server which contains a webpage which makes the user login into the portal. As the application on user's computer starts, another module is executed in background which sends requests to MySQL databases and check if some new row is inserted. If a new row is detected that command is executed accordingly.

Following query is executed to get the latest command-

SELECT * FROM `command1` WHERE `id`=" + remote_id + " order by `srno` desc LIMIT 0,1

The remote_id is given on every setup on client PC. It is stored on the user as well as website MySQL databases.

Adding New Commands

We have used an Ms-Access database to store custom commands as well as some predefined commands. It contains regular expressions with which the input from user are matched and the executable file with arguments given in database only are executed. To add command, user fills the fields like user input, speech response etc. then click on one of the following buttons open a file, open a folder, open a web page and custom speech response. On submitting information required, a simple insert query is performed on Ms- Access database which store the relevant information so that the user expected response is obtained for the given speech input.

Play Typical Songs and Video

The .NET has a function called Directory.GetFiles (folder_path, filename_pattern); using this function we search all the folders and drives recursively. For videos, we use different extensions like mp4, avi, wmv, flv, mkv, mpg etc. and for songs we search the input song string from user with as, Directory.GetFiles (folder, "*" + name + "*.exe").

Once the file matching with the pattern is found, that file is executed and the respective video or songs starts. This process continues until all the drives and folders are not searched.

6. RESULTS

When the application is executed it starts remote module in background. When it gets the command it first tries to match the input with the commands stored in the Ms-Access database if matched it executes the command accordingly. If not then it check the command for some hardcoded flows of matched then the relevant output is given to user.

7. CONCLUSION

Voice Controlled Personal Assistant System will use the Natural Language processing and can be integrated with artificial intelligence techniques to achieve a smart assistant that can control IoT applications and even solve user queries using web searches. It can be designed to minimize the human efforts to interact with many other subsystems, which would otherwise have to be performed manually. By achieving this, the system will make human life comfortable. More specifically, this system is designed to interact with other subsystems intelligently and control these devices, this includes IoT devices or getting news from Internet, providing other information, getting personalized data saved previously on the system.

number of assistants are available in the market for smartphones, personal assistants for PCs (Windows) are not that popular in our country.

This paper describes the approach that is used to develop a personal assistant. All the technical and implementation details are described to understand the system better. Most of the basic features needed regularly are implemented though a number of additional features can be added. Thus we have developed a personal assistant such that all most of the user tasks are now implemented at the users command easily and quickly.

8. REFERENCES

[1] Johann Hauswald, Michael A. Laurenzano, Yunqi Zhang, Cheng Li ,Austin Rovinski, Arjun Khurana, Ronald G. Dreslinski, Trevor Mudge, Vinicius Petrucci1, Lingjia Tang, Jason Mars, "Sirius: An Open End-to-End Voice and Vision Personal Assistant and Its Implications for Future Warehouse Scale Computers",ASPLOS'15, March14–18,2015,Istanbul,Turkey.

[2] P. Milhorat, S. Schl¨ogl, G. Chollet, J. Boudy, A Esposito and G. Pelosi," Building The Next Generation

Of Personal Digital Assistants",1st International Conference on Advanced Technologies for Signal and Image Processing - ATSIP'2014 March 17-19, 2014, Sousse, Tunisia

[3] Hong-wen Sie 1, Dau-Cheng Lyu 1, Zhong-Ing Liou 1, Ren-Yuan Lyu 1,2,Yuang-Chin Chiang3 1Dept. of Electrical Engineering, Chang Gung University, Taoyuan," A Multilingual Automatic Speech Recognition (ASR) Engine Embedded on Personal Digital Assistant (PDA)" Cellular Neural Networks and Their Applications, 2005 9th International Workshop

[4] Kaoruko Ohtani, Takahito Niwa,Naohiro Ishii, Nahoko Shirahama, Ippei Torii," Voice Output Communication id pplication for Personal i ital ssistant for Autistic Children ",2012 IEEE/ACIS 11th International Conference on Computer and Information Science