JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Virtual Personal Assistance the use of Python

[1]Rimpi Datta, [2]Himat Pal Singh, [3]Piyush Pyne, [4]Atanu Maity, [5]Tiyasa Sarkar, [6]Soham Sen

Narula Institute of Technology, ECE Department, Agarpara, Kolkata-109

Abstract:

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like Google, Siri. etc. Now in our voice assistance system, it can act as a basic medical prescriber, daily schedule reminder, note writer, calculator and a search tool. This project works on voice input and gives output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the World Wide Web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

Keywords: Pyttsx3, Shutil, SubProcess, Speech Recognition , CType etc.

Introduction

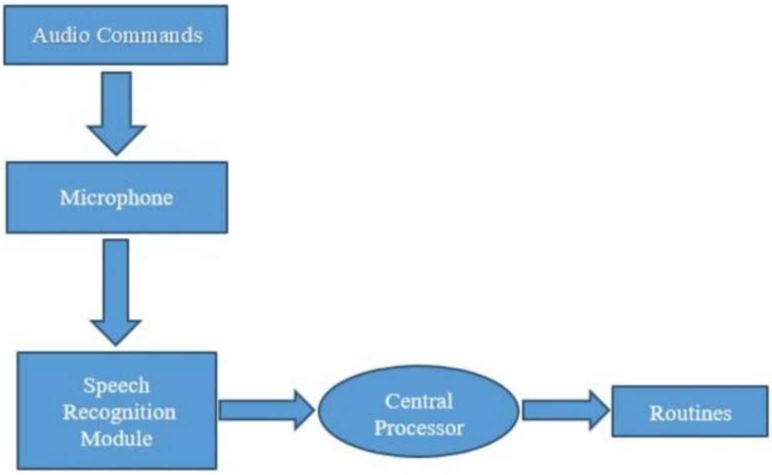
Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behavior and trying to become his personalized assistant.

Innovativeness:

Each company developer of the intelligent assistant applies his own specific methods and approaches for development, which in turn affects the final product. One assistant can synthesize speech more qualitatively, another can more accurately and without additional explanations and corrections perform tasks, others can perform a narrower range of tasks, but most accurately and as the user wants. Obviously, there is no universal assistant who would perform all tasks equally well. The set of characteristics that an assistant has depends entirely on which area the developer has paid more attention to. Since all systems are based on machine learning methods and use for their creation huge amounts of data collected from various sources and then trained on them, an important role is played by the source of this data, be it search systems, various information sources or social networks. The amount of information from different sources determines the nature of the assistant, which can result as a result. Despite the different approaches to learning, different algorithms and techniques, the principle of building such systems remain approximately the same. Figure 1 shows the technologies that are used to create intelligent systems of interaction with a human by his natural language. The main technologies are voice activation, automatic speech recognition, Teach-To-Speech, voice biometrics, dialogue manager, natural language understanding and named entity recognition.

Proposed Plan of Work

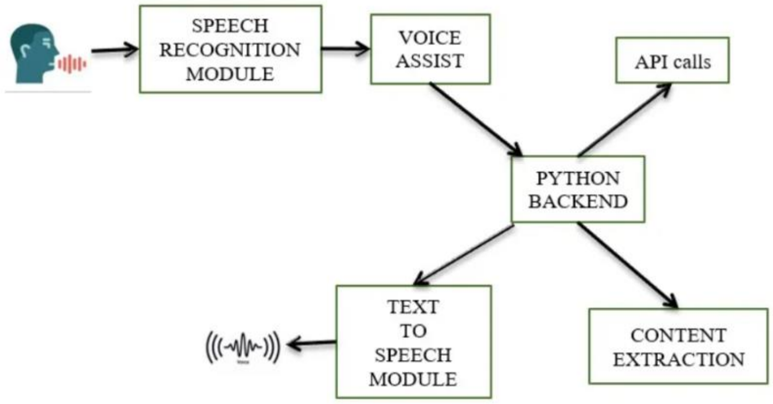
The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.



Basic Workflow

This shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution. But, the complexities don't stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

Detailed Workflow



Modules of Virtual Assistant Using Python:

1) SubProcess Module: Subprocesses with accessible I/O streams. This module allows you to spawn processes, connect to their input/output/error pipes, and obtain their return codes. For a complete description of this module see the Python documentation.

2) Pyttsx3:

   This give you two types of assistance voices like mail and female.

3) speech recognition: Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction on how to make use of the SpeechRecognition library of Python. This is useful as it can be used on microcontrollers such as Raspberri Pis with the help of an external microphone.

4) DateTime:

5) Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps

6) Wikipedia:

   Python provides the Wikipedia module (or API) to scrap the data from the Wikipedia pages. This module allows us to get and parse the information from Wikipedia.

7) Webbrowser:

   The webbrowser module provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.

8) OS:

OS routines for NT or Posix depending on what system we're on.
This exports:
- all functions from posix or nt, e.g. unlink, stat, etc.
- os.path is either posixpath or ntpath
- os.name is either 'posix' or 'nt'
- os.curdir is a string representing the current directory (always '.')
- os.pardir is a string representing the parent directory (always '..')
- os.sep is the (or a most common) pathname separator ('/' or '\\')
- os.extsep is the extension separator (always '.')
- os.altsep is the alternate pathname separator (None or '/')
- os.pathsep is the component separator used in $PATH etc
- os.linesep is the line separator in text files ('\r' or '\n' or '\r\n')
- os.defpath is the default search path for executables
- os.devnull is the file path of the null device ('/dev/null', etc.)

Programs that import and use 'os' stand a better chance of being portable between different platforms. Of course, they must then only use functions that are defined by all platforms (e.g., unlink and opendir), and leave all pathname manipulation to os.path (e.g., split and join).

9) Ctypes:

Ctypes is a foreign function library for Pyhton. It provides C compatible data types, and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure Pyhton.

10) Time:

This module provides various functions to manipulate time values. There are two standard representations of time. One is the number of seconds since the Epoch, in UTC (a.k.a. GMT). It may be an integer or a floating point number (to represent fractions of seconds). The Epoch is system-defined; on Unix, it is generally January 1st, 1970. The actual value can be retrieved by calling gmtime(0).
The other representation is a tuple of 9 integers giving local time. The tuple items are: year (including century, e.g. 1998) month (1-12) day (1-31) hours (0-23) minutes (0-59) seconds (0-59) weekday (0-6, Monday is 0) Julian day (day in the year, 1-366) DST (Daylight Savings Time) flag (-1, 0 or 1)
If the DST flag is 0, the time is given in the regular time zone; if it is 1, the time is given in the DST time zone; if it is -1, mktime() should guess based on the date and time.

11) Shutil:

Shutil module in Python provides many functions of high-level operations on files and collections of files. It comes under Python's standard utility modules. This module helps in automating process of copying and removal of files and directories.

12) Urlopen:

Urllib package is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and is able to fetch URLs using a variety of different protocols.

Result Analysis:

assname=input("Enter Assistance name: ")
This is the input you have to provide to give the assistance name.

uname = input("Enter your name: ")
This is the input you have to provide to give the user name.

In the wishme() function the assistance will first greetings you like "Good Morning __". Then it will ask you "How can I help you?".

In takecommand() function the system take all the command and quarry from user with the help of system microphone and recognize module.

The quarries one can ask this assistance:

How are you?

Change assistance name

What is your name?

Exit

Who are you?

Who I am?

Why you have come to this world

Who made you?

Reason for you

What is love ?

Change background

Lock window

Shut down the system

Stop listening

Restart

Hibernate/sleep

Search Wikipedia

Will you be my GF

I love you

Open google/youtube/facebook

Play music/videos

Tell the current time

Conclusion:

In this paper "Virtual Assistant Using Python" we discussed the design and implementation of Digital Assistance. The project is built using open source software modules with Visual Studio Code backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.