



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 3, March 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.165

 9940 572 462

 6381 907 438

 ijircce@gmail.com

 www.ijircce.com

AI Desktop Assistant Using Python

Hritik Zende, Suraj Dalvi, Sejal Agarwal, Raksha Thakur, Prof. Sujeet More

BE Students, Dept. of I.T., Trinity College of Engineering and Research, Pune, Maharashtra, India

Dept. of I.T., Trinity College of Engineering and Research, Pune, Maharashtra, India

ABSTRACT: The Project aims to Develop a Personal-Assistant for Desktops and Laptops. The Assistant draws its Inspiration from Virtual Assistant like Cortana for Windows, and Siri for iOS. It has been designed to provide User-friendly Interface for carrying out a variety of tasks. User can Interact with the assistant through Voice Commands. The Desktop Voice assistant who helps the end user to communicate with desktop computer with voice and it also responds to the voice commands of the user. Our Proposed System has capability to work with and without Internet Connectivity in Desktop Computer/ Laptop. Which takes the user Input through Voice and Process it and returns the output in various form like action to be performed or the Search result is dictated to the end user.

KEYWORDS: Voice assistant, Speech Recognition, Low cost, Internet, Speech Synthesis, Visually Challenged, Desktop Assistant, Text to Speech, Speech to Text, Language.

I. INTRODUCTION

Today the development of artificial intelligence (AI) systems that are able to 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 learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behaviour and trying to become his personalized assistant.

In our project, we mainly use voice as communication means, so the basically the Speech recognition application. Same kind of application is also developed by the Google that is "Google Voice Search" which is used for in Android Phones. But this Application mostly works with Internet Connections. The concept of speech technology really encompasses two technologies: Synthesizer and recognizer. A speech synthesizer takes as input and produces an audio stream as output. A speech recognizer, on the other hand, does the direct opposite. It takes an audio stream as input and thus turns it into text transcription. Direct analysis and synthesizing the complex voice signal is due to too much information contained in the signal. Therefore the digital signal processes such as Feature Extraction and Feature Matching are introduced to represent the voice signal. In this project, we directly use the speech engine which uses Feature extraction technique.

II. LITERATURE REVIEW

1. Survey of Technology

A. Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, Data science etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize voice, Pyttsx3 for text to speech, selenium for web automation etc. Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

B. Quepy

Quepy is a python framework to transform natural language questions to queries in a database query language. It can be easily customized to different kinds of questions in natural language and database queries. So, with little coding you can build your own system for natural language access to your database.



C. Pyttsx3

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

D. Speech Recognition

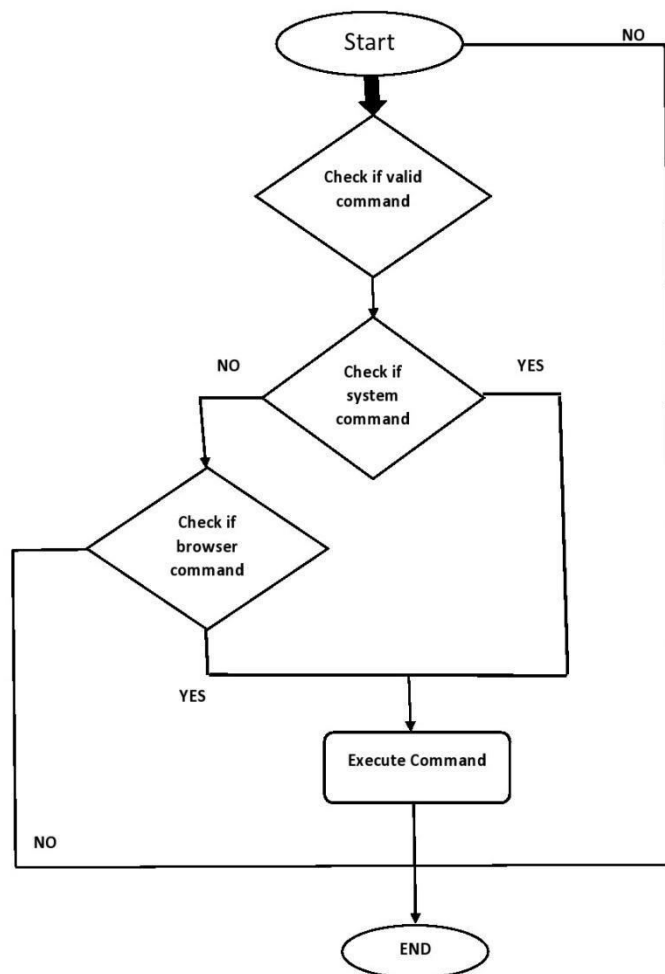
This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

E. SQLite

SQLite is a capable library, providing an in-process relational database for efficient storage of small-to-medium sized data sets. It supports most of the common features of SQL (Structured Query Language) with few exceptions. Best of all, most Python users do not need to install anything to get started working with SQLite, as the standard library in most distribution ships with the sqlite3 module. SQLite runs embedded in memory alongside your application, allowing you to easily extend SQLite with your own Python code. SQLite provides quite a few hooks, a reasonable subset of which are implemented by the standard library database driver..

III. PROPOSED SYSTEM

A. System Architecture





B. Modules:

- SPEECH RECOGNITION

The speech recognition module used the program is Google's Speech Recognition API which is imported in python using the command "import speech_recognition as sr". This module is used to recognize the voice which is given as input by the user.

This is a free API that is provided and supported by Google. This is a very light API that helps in reducing the size of our application.

- TTS & STT

The voice which is given as input is first converted to text using the speech recognition module. The text is then processed to give the result of the query given by the user. The final step is the conversion of the result of the processed query to speech which is the final output. The most time consuming among the two is STT because the system first has to listen to the user and different users have different, some are easy to understand while some are not easily audible. This is the step upon which our total execution time depends. Once the speech is converted to text executing commands and giving the results back to the user is not a time-consuming step.

- API calls

Application Programming Interface is software intermediary and the work of the API is to allow two applications to talk each other. In other words, API is the message passer that delivers the users request to the provider and then send the response back to users.

- System Calls

In System Calls a programmatic approach takes place in which a computer program requests a service from kernel of the OS it is executed on, which may include functions related to hardware services for example, accessing hard disk drive, creation and execution of new processes and communication with process scheduling. It provides an important interface between the process and the OS.

```
file Edit Selection View Go Run Terminal Help final.py - mi
final.py ×
final.py > ...
1 # importing speech recognition package from google api
2 import speech_recognition as sr
3 import playsound # to play saved mp3 file
4 from gtts import gTTS # google text to speech
5 import os # to save/open files
6 import datetime
7 import pyjokes
8 import requests
9 import json
10 import wikipedia #pip install wikipedia
11 import webbrowser
12 import wolframalpha # to calculate strings into formula
13 from selenium import webdriver # to control browser operations
14 from urllib.request import urlopen
15
```



- IMPORTED MODULES

A. PYTTSEX3

The pyttsex3 is an offline module that is used for text to speech conversion in Python and it is supported by both Python 2 & 3. The run and wait functionality is also in this module only. It determines how much time the system will wait for another input or in other words the time interval between inputs.

This is a free module available in the python community which can be installed using the pip command just like other modules.

B. DATETIME

The Date Time module is imported to support the functionality of the date and time. For example, the user wants to know the current date and time or the user wants to schedule a task at a certain time. In short this module supports classes to manipulate date and time and perform operations according to it only. This is an essential module, especially in tasks where we want to keep a track of time. This module is very small in size and helps to control the size of our program. If the modules are too large or heavy then the system will lag and give slow responses.

C. WEBBROWSER

This module allows the system to display web-based information to users. For example, the user wants to open any website and he gives input as “Open Google”. The input is processed using the web browser module and the user gets a browser with google opened in it. The browser which will be used is the default set web browser.

D. WIKIPEDIA

Wikipedia is a library in python which it possible for the virtual assistant to process the queries regarding Wikipedia and display the results to users. This is an online library and needs an internet connection to fetch the results.

The no. of lines that the user wants to get as a result can be set manually.

E. OS MODULE

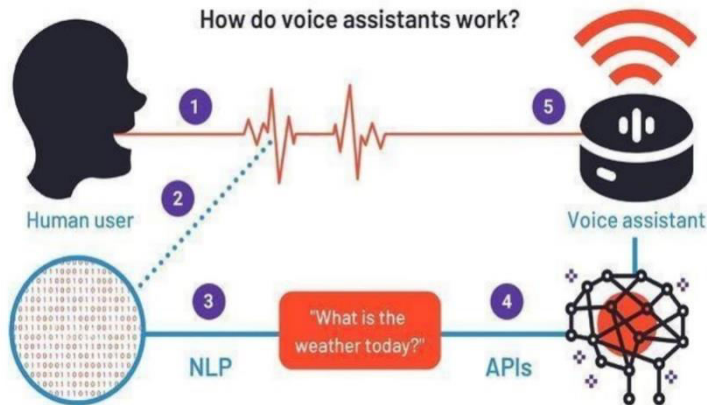
OS Module provides an operating system dependent functionalities. If we want to perform operations on files like reading, writing, or manipulate paths, all these types of functionalities are available in an OS module. All the operations available raise an error “OSError” in case of any error like invalid names, paths, or arguments which may be incorrect or correct but just no accepted by the operating system.

F. SMTPLIB

Python has this module for in the standard library for working with emails & email servers. The SMTPLIB defines an object known as “SMTP client session object” which is used to send mails by the user. There are 3 steps involved - initialize, sendmail(), quit. When the optional parameters which are host and port, are provided connect method is called with these arguments during the first step which is initialization

G. WOLFRAM ALPHA

Wolfram alpha — Wolfram Alpha is an API which can compute expert-level answers using Wolfram’s algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.



IV. SIMULATION RESULTS

4.1 Test Case Design

A. Test Case 1

Test Title: Response

Time Test ID: T1

Test Priority: High

Test Objective: To make sure that the system respond back time is efficient.

Description: Time is very critical in a voice-based system. As we are not typing inputs, we are speaking them. The system must also reply in a moment. User must get instant response of the query made.

B. Test Case 2

Test Title: Accuracy

Test ID: T2

Test Priority: High

Test Objective: To assure that answers retrieved by system are accurate as per gathered data.

Description: A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system

C. Test Case 3

Test Title: Approximation

Test ID: T3

Test priority: Moderate

Test Objective: To check approximate answers about calculations.

Description: There are times when mathematical calculation requires approximate value. For example, if someone asks for value of PI the system must respond with approximate value and not the accurate value. Getting exact value in such cases is undesirable

Virtual assistant is a less time consuming. Virtual assistant is a software that understands commands and complete task assigned by client. Virtual assistant use NLP to match user voice or text input with executable commands. With the help of virtual assistant you able to run your machine like laptop or PC's on your own command. It is the fast process, therefore it saves time. Virtual assistant is working for you at set times, so always available to you and able to adapt to changing needs quickly. Virtual assistant will be available to you and, should their workload enable, help others too, such as family and colleagues



V. CONCLUSION

In this paper we have discussed about Personal Desktop Assistant for Windows Using Python. Desktop assistant makes life easier to humans. Assistant is the flexibility to contract for just the services they need. As like Alexa, Cortana, Siri, Google assistant we also make virtual assistant using python for all windows versions. We use Artificial Intelligence technology for this project. Virtual Desktop Personal Assistants are effective way to manage or organize your schedule. Virtual Desktop assistants are also reliable than Human Personal Assistant because, virtual Assistants are more portable, loyal and available to use anytime. Our virtual assistant will be intimate you with suggestions and taking instructions, and will know more about you. We can expect this device to be permanent..

REFERENCES

- [1]. Cortana Intelligence, Google Assistant, Apple Siri
- [2]. <https://data-flair.training/blogs/artificial-intelligence-project-ideas/>
- [3]. <https://www.upgrad.com/blog/top-artificial-intelligence-project-ideas-topics-for-beginners/>
- [4]. <https://www.activestate.com/blog/how-to-build-a-digital-virtual-assistant-in-python/>
- [5]. <https://towardsdatascience.com/how-to-build-your-own-ai-personal-assistant-using-python-f57247b4494b>
- [6]. <https://www.section.io/engineering-education/creating-a-virtual-assistant-using-python/>
- [7]. <https://medium.com/codex/making-your-own-ai-virtual-assistant-with-python-5c2046dadfa7>
- [8]. <https://nevonprojects.com/voice-based-intelligent-virtual-assistance-for-windows/>
- [9]. <https://www.codewithharry.com/>
- [10]. www.stackoverflow.com
- [11]. www.pythonprogramming.net
- [12]. www.codecademy.com
- [13]. www.tutorialspoint.com
- [14]. www.google.co.in
- [15]. www.python.org
- [16]. Python Programming - Kiran Gurbani
- [17]. Learning Python - Mark Lutz
- [18]. Designing Personal Assistant Software for Task Management using Semantic Web Technologies and Knowledge Databases.
- [19]. Pythoncode for Artificial Intelligence: Foundations of Computational Agents, David L. Poole and Alan K. Mackworth
- [20]. Chatbot Learning: Everything you need to know about machine learning chatbots (2020). <https://www.whoson.com/chatbots-ai/chatbot-learning-everything-need-know-machine-learning-chatbots/>
- [21]. How to use an API with Python (Beginner's Guide). Retrieved from the link <https://rapidapi.com/blog/how-to-use-an-api-with-python/>
- [22]. Introduction to Machine Learning using Python (January 2019). Retrieved from <https://www.geeksforgeeks.org/introduction-machine-learning-using-python/>



INNO  **SPACE**
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi[®]
cross **ref**

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 **9940 572 462**  **6381 907 438**  **ijircce@gmail.com**



www.ijircce.com

Scan to save the contact details