Voice assistant using Python

*

Dhruv Dholariya
Department of Electronics and Communication Engineering
Nirma University
Ahmedabad
20bec024@nirmauni.ac.in

Aditya Jha
Department of Electronics and Communication Engineering
Nirma University
Ahmedabad
20bec005@nirmauni.ac.in

Abstract **- In modern times, everyday life has become smarter and interlinked with technology. We already know some voice services like google and Siri. etc. Now in our voice support system, it can work like automatic chrome, open LMS for you in a go, open a social media website in a web browser, tell you the time and you can ask him to brief you about anyone from Wikipedia, etc. You can also change your Voice Assistant at any time. This project works by entering voice and rendering voice output along with displaying text on the screen. Our main voice help agenda makes people smarter and deliver faster results with a computer in a lot easy manner. The Assistant captures voice input with our microphone and transforms our voice into understandable computer language providing the necessary solutions and answers that the user asks. This service is linked to the World Wide Web to provide the results the user has requested. The Natural Language Processing algorithm enables computer systems to engage in communication using the natural human language in many ways.**

Index Terms – Pyttsx3(Python speech to text library)

I. INTRODUCTION

Nowadays, almost all duties are completed digitally. The entire world is essentially at our fingertips when we have a smartphone in our hands. Even the use of fingertips has disappeared. The job is merely briefly mentioned and then finished. There are Voice Assistant that let us do tasks such as text Dad to let him know I'll be late today. That's what a voice assistant does. Additionally, it assists with specific tasks that all contribute to the automation of search, discovery, and online order processes, such as finding the cheapest book available online from several e-commerce sites and then providing an interface to place an order. Voice Assistants are software programs that help you ease your day-to-day tasks, such as opening Learning Management System (LMS), play songs on YouTube, Opening Bank Account Page, Booking Movie Tickets, etc. This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. It is effortless to use. Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 70% of searches will be via voice by 2024. This project was started that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that can help us in day to day life.

II. OBJECTTIVE

The main objective of developing Virtual Assistant and Desktop Assistant Software for Utilizing user-generated content, semantic data sources available on the web, and knowledge databases to provide expertise. The main purpose of an intelligent virtual assistant is to inform users. A chat interface on the business website, for example, can be used to conduct this in a formal manner. You can save a tonne of time by using virtual assistants. Before applying our understanding to create the report, we spend hours researching web research. One of the main advantages of voice searches is their quickness. Speaking is really claimed to be approximately 4 times faster than writing because it can be done at a rate of about 150 words per minute as opposed to writing, which we can do at a rate of about 40 words per minute. In this regard, the ability of personal assistants to accurately perceive spoken language is a prerequisite for their adoption by consumers.

III. WORKING

In our Voice assistant we have tried to make the program User Interactive as much as possible. We have compiled both Voice Assistant and Desktop Assistant in a single program. A user can

speak about what has to be done by the Assistant, the program will record the command and convert it into computer understood code and will force the engine to work accordingly. The assistant can be changed at the run time only it doesn't need to change any line of code or hard coding. We have added many features in our program which are listed below. There are a lot to add in the Assistant but we have tried to add only main and unique features in our Assistant and make the code as compact as possible.

IV. Features

- Opening LMS
- Amazon
- Google Search
- Movie Ticket
- Play on YT
- Speak Time
- Redirect to Net Banking websites
- Get the info from Wikipedia
- Change the Assistant

V. Python Libraries used

- **Speech Recognition Library:**

Library for performing speech recognition, with support for several engines and APIs, online and offline. Speech recognition engine/API support: ○ CMU Sphinx (works offline) ○ Google Speech Recognition ○ Google Cloud Speech API ○ Wit.ai ○ Microsoft Bing Voice Recognition ○ Houndify API ○ IBM Speech to Text

○ Snowboy Hotword Detection (works offline)

- **PyAudio :**

PyAudio provides Python bindings for PortAudio v19, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple macOS.PyAudio is required if and only if you want to use microphone input (Microphone). PyAudio version 0.2.11+ is required, as earlier versions have known memory management bugs when recording from microphones in certain situations.

- **PyWhatKit :**

PyWhatKit is a Python library with various helpful features. It's easy-to-use and does not require you to do any additional setup. Currently, it is one of the most popular library for WhatsApp and YouTube automation. New updates are released frequently with new features and bug fixes.
Features

- Sending Message to a WhatsApp Group or Contact
- Sending Image to a WhatsApp Group or Contact
- Converting an Image to ASCII Art
- Converting a String to Handwriting
- Playing YouTube Videos
- Sending Mails with HTML Code
- **Pyttsx3 :**

Pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male.

V. References

[1] R. Belvin, R. Burns, and C. Hein, "*Development of the HRL route navigation dialogue system*," in Proceedings of ACL-HLT, 2001

[2] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T.J.Hazen,and L.Hetherington, "*JUPITER: A Telephone Based Conversational Interface for Weather Information*," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, pp. 85–96, 2000.

[3] M. Kolss, D. Bernreuther, M. Paulik, S. St¨ucker, S. Vogel, and A. Waibel, "*Open Domain Speech Recognition & Translation: Lectures and Speeches*," in Proceedings of ICASSP, 2006.

[4] D. R. S. Caon, T. Simonnet, P. Sendorek, J. Boudy, and G. Chollet, "*vAssist: The Virtual Interactive Assistant for Daily Homer-Care*," in Proceedings of pHealth, 2011.

[5] Crevier, D. (1993). *AI: The Tumultuous Search for Artificial Intelligence*. New York, NY: Basic Books, ISBN 0465-02997-3.

[6] Sadun, E., &Sande, S. (2014). *Talking to Siri: Mastering the Language of Apple's Intelligent Assistant*.

Annexure

CODE :

```
import speech_recognition as sr
import pywhatkit
import pyttsx3  # Python text to speech library
import datetime
import webbrowser
import wikipedia

engine = pyttsx3.init()  # Initiating the engine | engine is object
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)  # Assigning female voice , for male 1 -> 0
recognizer = sr.Recognizer()  # Initiating the speech recoganization library
currassistant = 1

command = ''
currcommand = ''


def assistant():  # Declaring function
    go = True
    count = 1
    while go:
        with sr.Microphone() as source:  # Converting microphone as variable named source
            print("\n Clearing background noises...Please wait")
            if count == 1:
                engine.say('\n How Can I Help You .... \n')     # Engine will start interacting
                engine.runAndWait()
                print('\n How Can I Help You ? \n')
                count = count + 1

            elif count == -1:
                engine.say('Sorry I was unable to understand!! Please try again .. ') # Engine will start interacting
                engine.runAndWait()
                print('\n Sorry I was unable to understand!! Please try again .. \n')

            else:
                count = count + 1
                engine.say('Have I heard and  executed correctly?')
                engine.runAndWait()
                recognizer.adjust_for_ambient_noise(source, duration=0.5)
                currenrec = recognizer.listen(source)

                try:
                    currcommand = recognizer.recognize_google(currenrec, language='en_US')
                    currcommand = currcommand.lower()
                    print('Your message:', format(currcommand))

                except Exception as ex:
                    print(ex)

                if 'yes' in currcommand:
                    engine.say("Yay")
                elif 'no' in currcommand:
```

```
            engine.say("Sorry for the inconvenience.....")

        engine.runAndWait()
        engine.say('What to do next? ')
        engine.runAndWait()

    recognizer.adjust_for_ambient_noise(source, duration=0.5)
    recordedaudio = recognizer.listen(source)  # Recording audio from source(Microphone)
    print('Done recording')

try:
    command = recognizer.recognize_google(recordedaudio, language='en_US')
    command = command.lower()
    print('Your message: ', format(command))
except Exception as ex:
    print(ex)

if 'lms' in command:
    c = 'Opening LMS'
    engine.say(c)
    engine.runAndWait()
    webbrowser.open('https://lms.nirmauni.ac.in/login/index.php')

elif 'amazon' in command:
    c = 'Opening Amazon'
    engine.say(c)
    engine.runAndWait()
    webbrowser.open('https://www.amazon.in/')

elif 'google' in command:
    c = 'Opening Google'
    engine.say(c)
    engine.runAndWait()
    webbrowser.open('www.google.com')

elif 'ticket' in command or 'movie' in command:
    c = 'Opening Bookmyshow'
    engine.say(c)
    engine.runAndWait()
    webbrowser.open('https://in.bookmyshow.com/explore/home/ahmedabad')

elif 'play' in command:
    a = 'opening youtube..'
    engine.say(a)
    engine.runAndWait()
    pywhatkit.playonyt(command)

elif 'time' in command:
    time = datetime.datetime.now().strftime('%I:%M %p')
    print(time)
    engine.say(time)
    engine.runAndWait()

elif 'youtube' in command:
    b = 'opening youtube'
    engine.say(b)
```

```
            engine.runAndWait()
            webbrowser.open('www.youtube.com')

        elif 'bank' in command:
            c = 'Opening HDFC Bank Netbanking'
            engine.say(c)
            engine.runAndWait()
            webbrowser.open('https://www.hdfcbank.com/')

        elif 'assistant' in command:
            global currassistant
            if currassistant == 0:
                currassistant = 1
            else:
                currassistant = 0
            engine.setProperty('voice', voices[currassistant].id)
            engine.say("Changes applied")

        elif 'exit' in command:
            go = False

        elif 'about' in command:
            try:
                print('Printing your message...Please wait')
                text = recognizer.recognize_google(recordedaudio, language='en-US')
                # print('Your Message:{}', format(text))
            except Exception as ex:
                print(ex)

            wikisearch = wikipedia.summary(text)
            val = wikisearch.split(".")
            engine.say(val[0])
            engine.runAndWait()

        else:
            count = -1


    # Main Function
    while True:

        print("\n")
        print("----------------------------------------")
        print("|                                      |")
        print("|  1 - Connect to Assistant            |")
        print("|  2 - Exit                            |")
        print("|                                      |")
        print("----------------------------------------")
        print("\n")
        userinput = int(input("Enter your input :- "))
        if userinput == 1:
            assistant()
        elif userinput == 3:
            break
```