The Voice Recognist (Personal Assistant): P.E.T.E.R

Pyhton Project Presentaion

Presented By: -

Rishu Yadav (2K20/EE/218)

Saurav Kumar (2K20/CO/416)

Under the Guidance of

Prof. Madan Mohan Rayguru

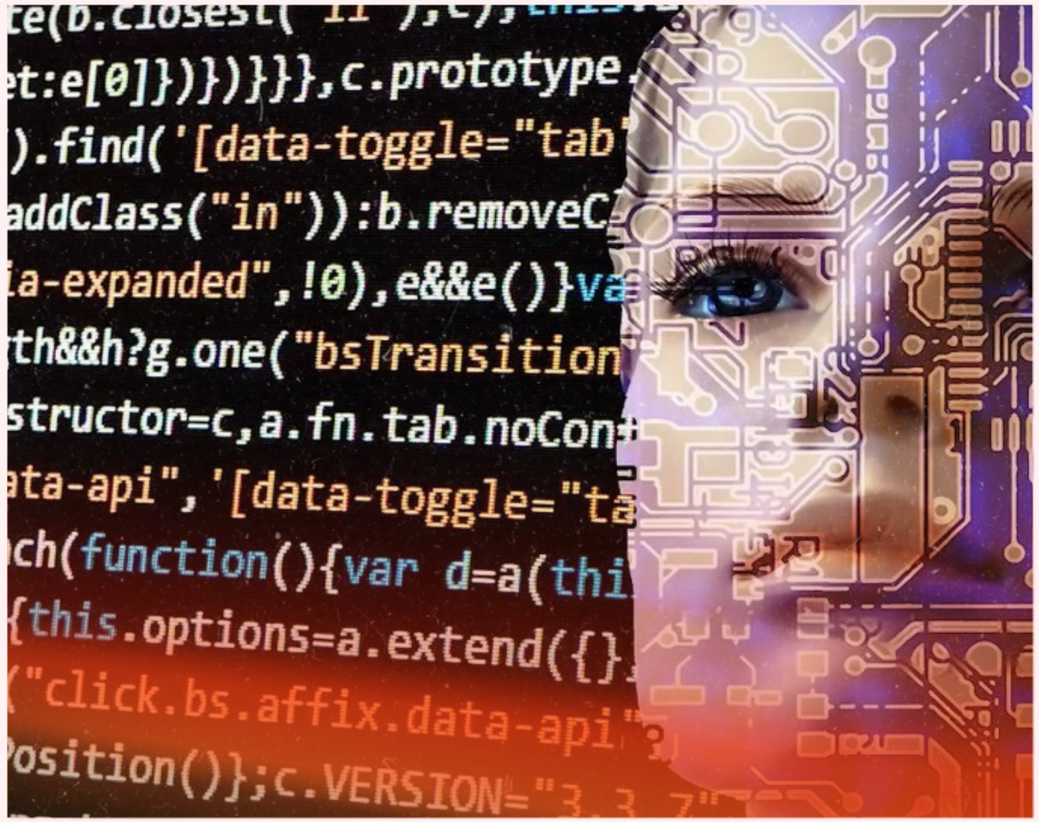
PPT link for viewers

TABLE OF CONTENTS

Speech Recognition

Speech recognition is an interdisciplinary subfield of computer science and computational linguistics that develops methodologies and technologies that enable the recognition and translation of spoken language into text by computers with the main benefit of searchability.

KODAK PORTA 400



KODAK PORTA 400

Application of Speech Recognition

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. The project aims to provide an introduction to 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.

Speech recognition technology and the use of digital assistants have moved quickly from our cellphones to our homes, and its application in industries such as business, banking, marketing, and healthcare is quickly becoming apparent.

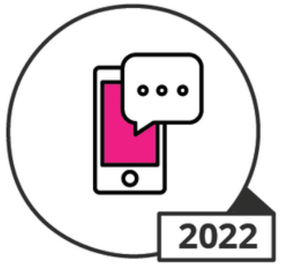
1. In the workplace
2. In banking
3. In marketing
4. In Healthcare
5. With the Internet of Things
6. In language learning

Popular digital assistants, include:

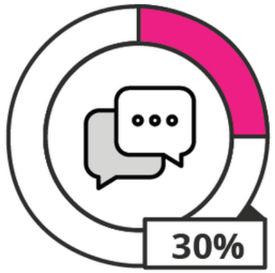
- Amazon's Alexa
- Apple's Siri
- Google's Google Assistant
- Microsoft's Cortana

Facts and Figures

Why Voice Recognition ?



With voice-based searches said to be the fastest-growing mobile search type,[3] it's predicted that **50% of searches will be in the voice-activated category by 2020,**[4] with smart speakers predicted to reach 55% of households in the United States by 2022.[5]



By 2021, early adopter brands optimising their websites to support voice-search will **increase digital commerce revenue by 30%.**[12]

Aim and Objective

Have you ever wondered how cool it would be to have your own A.I. assistant? Imagine how easier it would be to send emails without typing a single word, doing Wikipedia searches without opening web browsers, and performing many other daily tasks like playing music with the help of a single voice command. Through this Project, we will show you how you can make your personal A.I. assistant using Python.

What can this A.I. assistant do for you?

- It can send emails on your behalf.
- It can play music for you.
- It can do Wikipedia searches for you.
- It is capable of opening websites like Google, Youtube, etc., in a web browser.
- It is capable of opening your code editor or IDE with a single voice command.

Enough talks! Let's start building our own P.E.T.E.R.

Module used

- **pyttsx3** ~ is a text-to-speech conversion module 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.
- **Speech Recognition** ~ is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction to how to make use of the Speech Recognition library of Python. This is useful as it can be used on microcontrollers such as Raspberri Pis with the help of an external microphone.
- **Datetime** ~ In Python, date and time are not a data type of their own, but a module named date time can be imported to work with the date as well as time. Python Date time module comes built into Python, so there is no need to install it externally.
- **Wikipedia** ~ is a multilingual online encyclopedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system. Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia.
- **Webbrowser** ~ is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users.

- **Os** ~ The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.
- **Smtplib** ~ The smtplib module defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP.
- **PyAutoGUI** ~ is a Python module which can automate your GUI and programmatically control your keyboard and mouse.
- **Time** ~ is a Python module which can automate your GUI and programmatically control your keyboard and mouse.

```
import pyttsx3  # pip install pyttsx3
import speech_recognition as sr  # pip install speechRecognition
import datetime
import wikipedia  # pip install wikipedia
import webbrowser
import os
import smtplib
import pyautogui as pag # pip install pyautogui
import time
```

Let's Walk Through The Code

```
11    engine = pyttsx3.init('sapi5')
12    voices = engine.getProperty('voices')
13    engine.setProperty('voice', voices[0].id)
```

The Function Of The Following Code is :-

- To get the voices from our computer .
- Here we are using male voice as our peter's voice.
- In voices we are storing default voices in our system.

What is sapi5?

- Microsoft developed speech API.
- Helps in synthesis and recognition of voice.

This speak function :-

```
15    def speak(audio):
16        engine.say(audio)
17        engine.runAndWait()
```

- Say () - Take Argument as text you wish to hear
- runAndWait() - All the say() texts won't be said unless the interpreter encounters runAndWait()
- The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our J.A.R.V.I.S. talk, we will make a function called speak()..

This Wishme function :-

- we have made a wishme() function that will make our PETER wish or greet the user according to the time of computer or pc.
- To provide current or live time to A.I., we need to import a module called datetime.

```
19    def wishMe():
20        hour = int(datetime.datetime.now().hour)
21        if hour >= 0 and hour < 12:
22            speak("Good Morning!")
23
24        elif hour >= 12 and hour < 18:
25            speak("Good Afternoon!")
26
27        else:
28            speak("Good Evening!")
29
30        speak("peter here. Sir how may I help you")
```

Take command Function :

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system.

- So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.
- Similarly , takeMessage():, takeName():, takeText():, takeYN(): - These functions work in same Manner but with different context.

```
def takeCommand():
    # It takes microphone input from the user and returns string output
    r = sr.Recognizer()# it will help to recognize the voice
    with sr.Microphone() as source:# it will use the microphoone as source
        print("Tell me what you want....")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing.....")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:
        # print(e)
        print(" Say that again please.....")
        return "None"
    return query
```

Over View of Main function, coding logic of peter :-

1. First we will call the wish me function to greet the user .
2. We have a infinite while loop for taking continuous instructions from the user until the user says ‘quit’.
3. The voice will be converted into string, stored in the query.
4. Now , we will analyse the user requirement according to the specific keyword in string query .

We are using the following keywords :-

- who is
- what is
- whatsapp
- speech to text
- open youtube
- play music
- time
- send email
- quit
- shutdown the system

To search something on Wikipedia

```
142     while True:
143         query = takeCommand().lower()
144         # Logics for executing tasks based on query
145         if 'who is' in query:
146             speak('Wait few seconds')
147             query = query.replace("who is", "")
148             results = wikipedia.summary(query, sentences=2)
149             speak("According to Wikipedia")
150             print(results)
151             speak(results)
152             time.sleep(1)
153             speak("what else i can search for you")
154 
155         elif 'what is' in query:
156             speak('Wait few seconds')
157             query = query.replace("what is", "")
158             results = wikipedia.summary(query, sentences=2)
159             speak("According to Wikipedia")
160             print(results)
161             speak(results)
162             time.sleep(1)
163             speak("what else i can search for you")
164 
```

In the above code, we have used an if statement to check whether Wikipedia is in the user's search query or not. If Wikipedia is found in the user's search query, then two sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

To send message through WhatsApp

```
165         elif 'whatsapp' in query:  # send message to rishu verma on whatsapp
166             message = takeMessage()
167             while message == "None":
168                 message = takeMessage()
169 
170             name = takeName().lower()
171             if thisdict.get(name) == None:
172                 while thisdict.get(name) == None:
173                     name = takeName().lower()
174             if thisdict.get(name) != None:
175                 webbrowser.open(f"https://wa.me/91{thisdict.get(name)}")
176 
177             time.sleep(7)
178             pag.typewrite(message)
179             pag.press("enter")
```

```
134     thisdict = {
135         "saurabh": "9821808096",
136         "saurav": "9821808096",
137         "sourav": "9821808096",
138         "rishu": "8447068664",
139         "saroj": "9210556669"
140     }
```

Speech to Text, Play Music, Opne Google Chrome and Open You Tube functions :

```
189         elif 'speech to text' in query:
190             f = open("myfile.txt", "x")
191             txt = takeText()
192             f = open("myfile.txt", "w")
193             f.write(txt)
194             f.close()
```

```
196         elif 'open youtube' in query:
197             webbrowser.open("youtube.com")
198
199         elif 'open google' in query:
200             webbrowser.open("google.com")
```

```
202         elif 'play music' in query:
203             music_dir = 'C:\\Users\\rishu\\Music\\Playlist'
204             songs = os.listdir(music_dir)
205             print(songs)
206             os.startfile(os.path.join(music_dir, songs[0]))
207
208         elif 'time' in query:
209             strTime = datetime.datetime.now().strftime("%H:%M:%S")
210             speak(f"Sir, the time is {strTime}")
```

To Quit and or to ShutDown the System

```
223         elif 'quit' in query:
224             speak("ok sir, have a nice day")
225             break
226 
227         elif "shutdown" in query:
228             speak("Are You sure you want to shutdown")
229             print("Do you wish to shutdown your computer? (yes/no)")
230             shutdown = (takeYN().lower())
231             print(shutdown)
232             if 'yes' in shutdown:
233                 os.system("shutdown /s /t 1")
234             elif 'no' in shutdown:
235                 speak("ok sir as you wish")
236 
237         else:
238             speak("try again...")
```

Send Email function :

We will create a sendEmail() function, which will help us send emails to one or more than one recipient.

In the above code, we are using the SMTP module, which we have already discussed above.

Note: Do not forget to 'enable the less secure apps' feature in your Gmail account. Otherwise, the sendEmail function will not work properly.

Calling sendEmail() function inside the main() function:

We are using the try and except block to handle any possible error while sending emails

```
212         elif 'send email' in query:
213             try:
214                 speak("What should I say to Mr. Rishu ?")
215                 content = takeCommand()
216                 to = "rishuyadav5349@gmail.com"
217                 sendEmail(to, content)
218                 speak("Email has been sent Syccessfully!")
219             except Exception as e:
220                 print(e)
221                 speak("Sorry Sir, I am not able to send this email. You may try again.")
```

```
133 def sendEmail(to, content):
134     server = smtplib.SMTP('smtp.gmail.com', 587)
135     server.ehlo()
136     server.starttls()
137     server.login('rishuyadav_ee20a14_56@dtu.ac.in', password())
138     server.sendmail('sauravkumarmehar97@gmail.com', to, content)
139     server.close()
```

Future Applications

This technology is still largely in its infancy, but with the theory of “hyper adoption”, by which consumers tend to adopt new technologies more quickly than they did in the past, it’s likely this technology is going to grow and improve rapidly.At this stage of the technology’s life cycle, having a clear idea of its potential and its likelihood to become a commonality in our daily lives in the near future is paramount. Businesses should be proactive in their approach to introducing or incorporating speech recognition technology into their digital marketing strategy and budget, while individuals should continue to explore the benefits of speech recognition in their daily activities. As accuracy rates improve and consumer buy-in increases, industries can expect to encounter a need to adapt to be more speech centric, and in turn more human, than may have seemed possible in previous years.

Thank You