GROUP-2 (D6AD)

Artificial Intelligence and Data Science

PYTHON MINI-PROJECT

MEMBERS

Om Bhatia (6)

Arunim Chakraborty(7)

Rupesh Dhirwani(10)

TITLE: Speech Recognition System

DESCRIPTION:

Speech recognition is a machine's ability to listen to spoken words and identify them. We can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. We can even program some devices to respond to these spoken words. In this mini-project speech recognition converts speech into text and then opens up an application.

PROGRAM FLOW:

Installation required:

Python Speech Recognition module:

```
pip install speechrecognition
```

PyAudio:

Windows users can install PyAudio by executing the following command in a terminal

```
pip install pyaudio
```

Python pyttsx3 module:

pip install pyttsx3

Speech Input Using a Microphone and Translation of Speech to Text

Allow Adjusting for Ambient Noise: Since the surrounding noise varies, we must allow the program a second or too to adjust the energy threshold of recording so it is adjusted according to the external noise level.

Speech to text translation: This is done with the help of Google Speech Recognition. This requires an active internet connection to work. However, there are certain offline Recognition systems such as PocketSphinx, but have a very rigorous installation process that requires several dependencies. Google Speech Recognition is one of the easiest to use.

Translation of Speech to Text:

First, we need to import the library and then initialize it using init() function. This function may take 2 arguments.

init(driverName string, debug bool)

drivername: [Name of available driver] sapi5 on Windows | nsss on MacOS

debug: to enable or disable debug output

After initialization, we will make the program speak the text using say() function.

This method may also take 2 arguments.

say(text unicode, name string)

text: Any text you wish to hear.

name: To set a name for this speech. (optional)

Finally, to run the speech we use runAndWait() All the say() texts won't be said unless the interpreter encounters runAndWait().

Below is the implementation.

Packages used:

webbrowser:

The package used in the project is web-browser package that can be used for opening the browser that is present in our machine to open an application that we commanded our machine to open

speech_recognition as sr:

This is the second package that we use in our machine to recognize the speech that is taken from our microphone of our laptop/computer to analyze the sound.

pygame:

The third package that we used in our project is pygame that serves the purpose of mixing the sound and helps to remove the unwanted noise, that was taken at the time of voice listening.

time:

The time package is the package that is imported from python library for the time given to the user to speak in front of the microphone in the machine. If an inactivity occurs for a certain period of time then the following message will be displayed as “Sorry could not recognize what you said” or if the voice doesn’t matches with the proper command.

Os:

The os package creates a temporary file in the machine that is created at the time of voice that has been provided by the user into the machine and saves as .mp3 file in a folder after the execution is carried out it deletes that created folder automatically.

Program:-

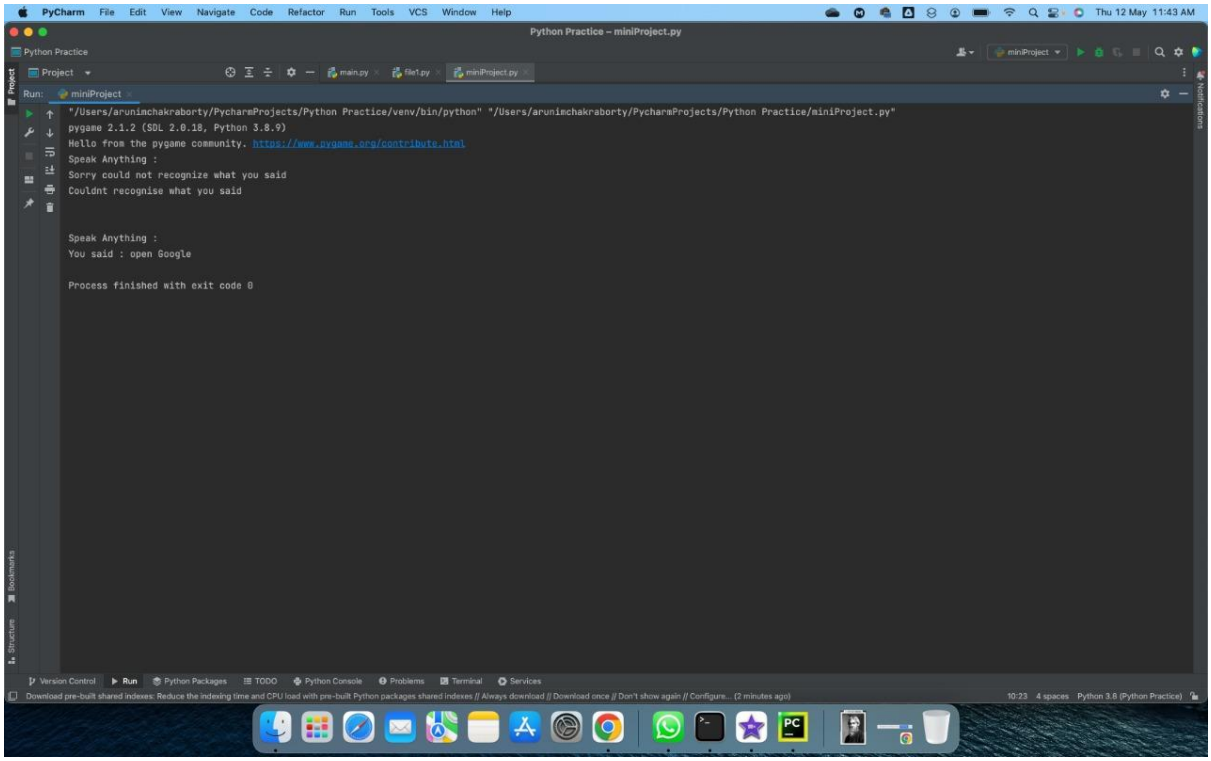
```
import webbrowser
import speech_recognition as sr
from gtts import gTTS
import os
import pygame, time


def speak(maintext):


    if os.path.exists("welcome.mp3"):
        os.remove("welcome.mp3")


    # Language in which you want to convert
    language = 'en'
```

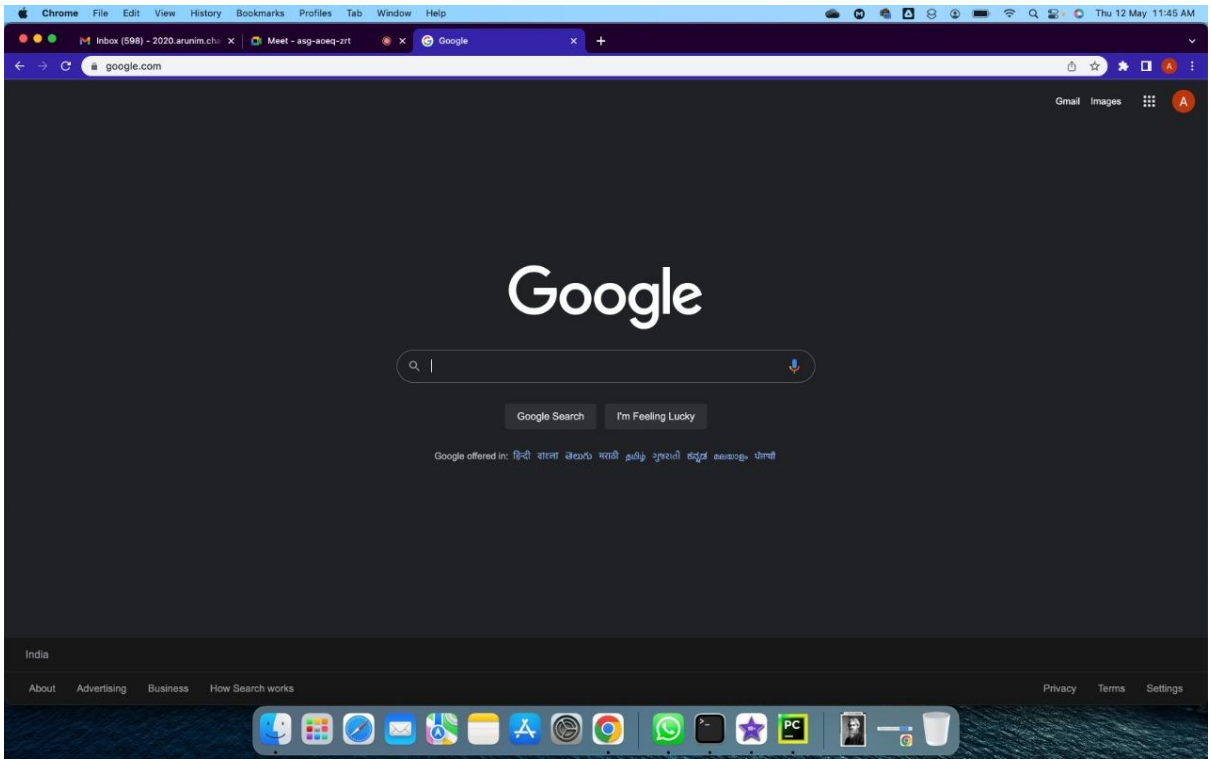
```
    myobj = gTTS(text=maintext, lang=language, slow=False)

    # Saving the converted audio in a mp3 file named
    # welcome
    myobj.save("welcome.mp3")

    pygame.init()
    pygame.mixer.music.load('welcome.mp3')
    pygame.mixer.music.play()
    time.sleep(2)
    pygame.mixer.music.unload()


    # Playing the converted file
    # os.system("mpg321 welcome.mp3")
    # os.remove("welcome.mp3")


def func():

    r = sr.Recognizer()
    with sr.Microphone() as source:
        speak("Speak Anything")
        print("Speak Anything :")
        audio = r.listen(source)
        s = ""
```

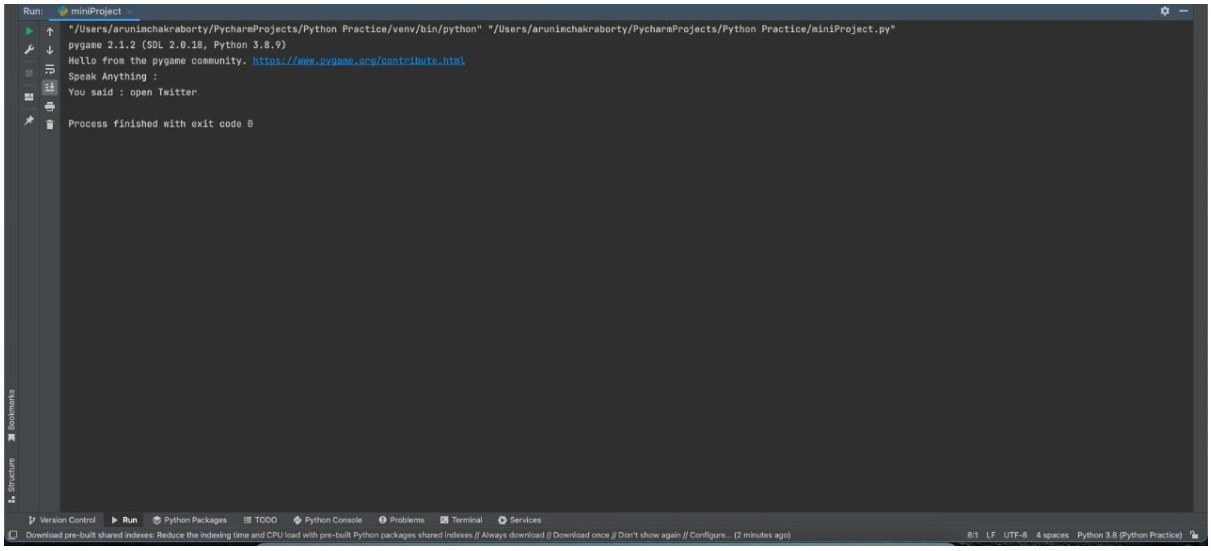
```
        try:
            text = r.recognize_google(audio)
            s = text
            print("You said : {}".format(text))
        except:
            print("Sorry could not recognize what you said")
            speak("Sorry could not recognize what you said")

        # getting path
        chrome_path = r"C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe"

        # First registers the new browser
        webbrowser.register('chrome', None,
                            webbrowser.BackgroundBrowser(chrome_path))

        if s.find("Google") != -1:
            speak("Opening Google")
            webbrowser.get('chrome').open("http://google.com")
        elif s.find('YouTube') != -1:
            speak("Opening Youtube")
            webbrowser.get('chrome').open("http://youtube.com")
        elif s == s.find("spotify") != -1:
            speak("Opening Spotify")
            webbrowser.get('chrome').open("http://spotify.com")
        elif s == s.find("Facebook") != -1:
            speak("Opening Facebook")
```

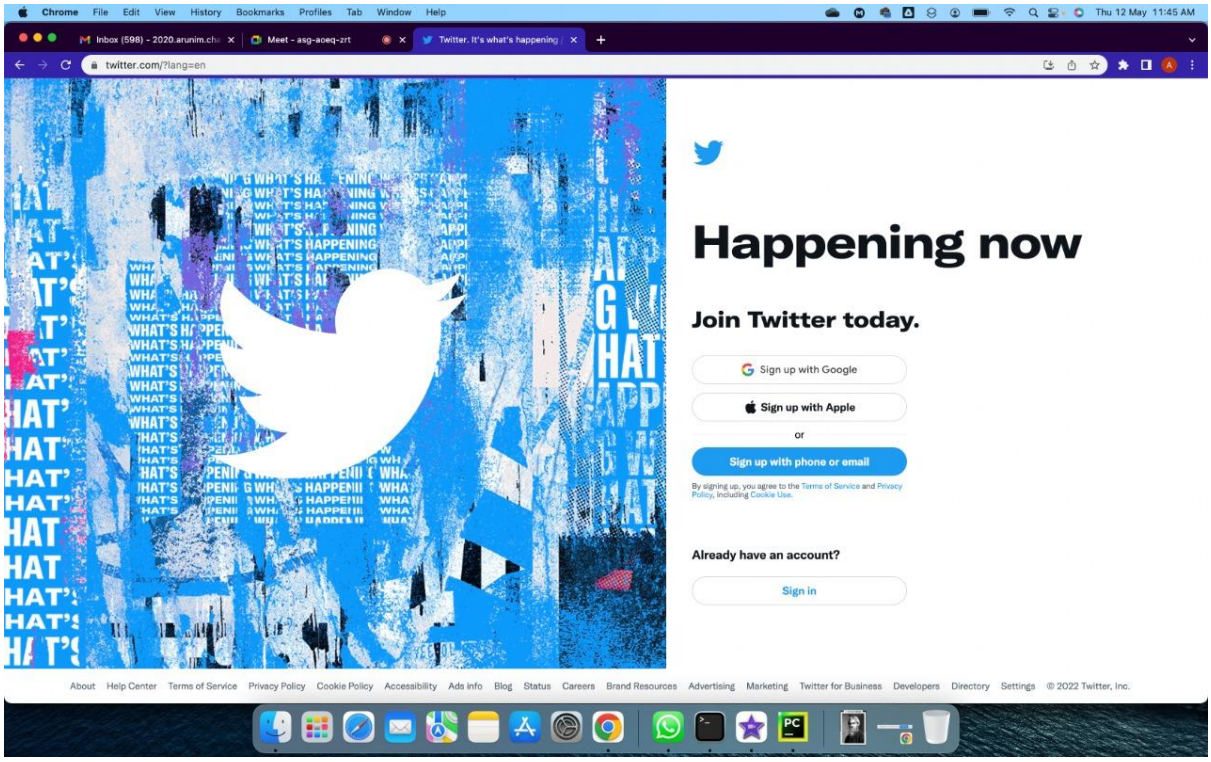
```
            webbrowser.get('chrome').open("http://facebook.com")
        elif s.find("Twitter") != -1:
            speak("Opening Twitter")
            webbrowser.get('chrome').open("http://twitter.com")
        elif s.find("Instagram") != -1:
            speak("Opening Spotify")
            webbrowser.get('chrome').open("http://instagram.com")
        else:
            print("Couldnt recognise what you said\n\n")
            speak("Couldnt recognise what you said")
            func()



func()
```

OUTPUT:


PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help
Thu 12 May 11:43 AM
Python Practice – miniProject.py
Python Practice
miniProject
Project
main.py
File1.py
miniProject.py
Run: miniProject
"/Users/arunimchakraborty/PycharmProjects/Python Practice/venv/bin/python" "/Users/arunimchakraborty/PycharmProjects/Python Practice/miniProject.py"
pygame 2.1.2 (SDL 2.0.18, Python 3.8.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
Speak Anything :
Sorry could not recognize what you said
Couldnt recognise what you said
Speak Anything :
You said : open Google
Process finished with exit code 0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (2 minutes ago)
10:23
4 spaces
Python 3.8 (Python Practice)


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 12 May 11:45 AM
Inbox (598) - 2020.arunim.ch
Meet - asg-aoeq-zrt
Google
google.com
Gmail
Images
Google
Google Search
I'm Feeling Lucky
Google offered in: हिंदी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
India
About
Advertising
Business
How Search works
Privacy
Terms
Settings





CONCLUSION:

1. Many speech recognition applications and devices are available, but the more advanced solutions use AI and machine learning. They integrate grammar, syntax, structure, and composition of audio and voice signals to understand and process human speech. Ideally, they learn as they go — evolving responses with each interaction.

II. Applications of speech recognition technology

Search for reports or documents on your computer.
Create a graph or tables using data.

Dictate the information you want to be incorporated into a document.
Print documents on request.
Start video conferences.
Schedule meetings.
Record minutes.
Make travel arrangements.

----------THANK-YOU----------