Python JARVIS Speech Recognition

Contents

Time required: 60 minutes

This series of tutorials were inspired by https://www.freecodecamp.org/news/python-project-how-to-build-your-own-jarvis-using-python/

NOTE: You may need to run this program from the command line for it to work.

The JARVIS Project

Speech Recognition

What good is an Iron Man suit if we can't communicate with it? Until the next release of Python, JARVIS can't read your mind . . . . yet.

The Python SpeechRecognition library performs speech recognition. It has support for several engines and APIs, online and offline.

https://pypi.org/project/SpeechRecognition/

PyAudio

To use your microphone, we need to install the PyAudio library. PyAudio is a cross-platform audio input/output stream library.

1. Go to a command prompt

```
pip install SpeechRecognition
pip install setuptools
pip install pyaudio
```

Tutorial Part 1

With all the JARVIS tutorials, we will start from the bare bones, and build to an OOP version. This code is based on sample code from the SpeechRecognition library GitHub.

https://github.com/Uberi/speech_recognition/blob/master/examples/microphone_recognition.py

```
"""
    Name: jarvis_speech_recognition_1.py
    Author:
    Created:
    Purpose: Voice recognition from Google
    Sample code using Google Speech Recognition
    from the SpeechRecognition library sample code.
    You can use the Google Speech API for free 50 times a day
"""
# We have to install pyaudio, we do not have to import it
# SpeechRecognition uses pyaudio directly
# pip install pyaudio
# pip install setuptools
# pip install SpeechRecognition
import speech_recognition as sr

# Create SpeechRecognition recognizer object
r = sr.Recognizer()
```

```python
# With your local microphone as the source
with sr.Microphone() as source:
    print('Listening . . . .')
    audio = r.listen(source)

    try:
        print('Recognizing . . .')
        # Capture the recognized word in a variable
        recognized_words = r.recognize_google(
            audio,
            language='en-US',
            show_all=True
        )

        # Google Speech Recognition returns a list of dictionaries
        # Pull only the transcript with the highest confidence
        recognized_words = recognized_words.get(
            'alternative'
        )[0].get(
            'transcript'
        )
        print(f"You may have said: {recognized_words}")

    except sr.UnknownValueError:
        print("Google Speech Recognition could not understand audio")

    except sr.RequestError as e:
        # If there was an error communicating with Google Speech
        print(f"Google Speech did not respond: {e}")
```

Tutorial Part 2

We can recognize speech. Time to start controlling our computer with our voice. This is an OOP example that allows you to quit the program by saying quit. This leads to many possibilities for a voice-controlled AI Python program.

```python
"""
    Name: jarvis_speech_recognition_2.py
    Author:
    Created:
    Purpose: Voice recognition from Google Speech API OOP
    You can use the Google Speech API for free 50 times a day
"""
# We have to install pyaudio, we do not have to import it
# SpeechRecognition uses pyaudio directly
# pip install pyaudio
# pip install setuptools
# pip install SpeechRecognition
import speech_recognition as sr
from sys import exit


Codiumate: Options | Test this class
class Jarvis:
    Codiumate: Options | Test this method
    def __init__(self) -> None:
        # Create SpeechRecognition recognizer object
        self.r = sr.Recognizer()
```

```python
# ---------------------- USER INPUT ----------------------------------------#
    Codiumate: Options | Test this method
    def user_input(self):
        """Recognizes user voice input using
            Speech Recognition module, converts it to text
        """
        # Your local microphone as the source
        with sr.Microphone() as source:
            print('Listening....')

            # Start listening for speech
            audio = self.r.listen(source)

            try:
                print('Recognizing . . .')
                # Capture the recognized word in a string variable
                recognized_words = self.r.recognize_google(
                    audio,
                    language='en-US',
                    show_all=True
                )

                # Google Speech Recognition returns a list of dictionaries
                # Pull only the transcript with the highest confidence
                self.query = recognized_words['alternative'][0]['transcript']
                print(self.query)
                # If you say quit, the program will exit

            except sr.UnknownValueError:
                print("Google Speech Recognition could not understand audio")

            except sr.RequestError as e:
                # If there was an error communicating with Google Speech
                print(f"Google Speech did not respond: {e}")
```

```python
# ---------------------- VOICE COMMANDS ---------------------------------- #
    Codiumate: Options | Test this method
    def voice_commands(self):
        if "quit" in self.query:
            print("Goodbye!")
            exit()
```

```
63  # Create a jarvis program object
64  jarvis = Jarvis()
65  while True:
66      jarvis.user_input()
67      jarvis.voice_commands()
```

Assignment: The JARVIS Project

It is now up to you. The world is under attack, aliens are everywhere, the space time continuum is falling apart, where is Iron Man?

Combine the text to speech and the speech recognition program into a program named **jarvis.py**

1. When the words you are speaking are recognized:
    a. They are displayed on the screen.
    b. JARVIS repeats the words.
    c. Add another command for JARVIS to do something. For example, you could have Jarvis print your name.

```
Listening....
Recognizing...
hello Jarvis
Listening....
Recognizing...
how are you this evening
Listening....
Recognizing...
quit
Goodbye!
```

Assignment Submission

1. Attach code files for both programs.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.