

Report_Lab2

Automatic Speech Recognition

2051840-梁厚

Overall Configuration

I configured this project in my virtualbox ubuntu22.04. Basic configuration is as follows (I have had python3.10.6 before):

- Install PocketSphinx

In terminal, run:

```
1 sudo apt-get install libpulse-dev
2
3 sudo apt-get install libasound2-dev
4
5 sudo apt install swig
6
7 # In final step, there was a error 'pip3:command not found', so I had to
  install a pip3
8 sudo apt-get install python3-pip
9
10 sudo pip3 install pocketsphin
```

- Install PyAudio

```
1 sudo apt-get install python-pyaudio python3-pyaudio
```

- Install SpeechRecognition

```
1 cd ~/lab2_asr
2
3 sudo pip install SpeechRecognition
```

- Run asr.py

```
1 cd ~/lab2_asr
2
3 # check
4 ls -l
5
6 chmod u+x asr.py
7
8 #run
9 python3 asr.py
```

Activities Unknown Apr 8 15:46

river@river-VirtualBox: ~/lab2_asr

Voice Assistant

Hi! How can I

You can:

1. Enjoy music by saying "Play music"
2. Take some notes by saying "Open Notepad"

```
2017.4.17 in /usr/lib/python3/dist-packages (from requests>=2.26.0->S
2.5 in /usr/lib/python3/dist-packages (from requests>=2.26.0->SpeechR
.27,>=1.21.1 in /usr/lib/python3/dist-packages (from requests>=2.26.0
cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (199 kB)
199.3/199.3 KB 1.7 MB/s eta 0:00:00
normalizer, requests, SpeechRecognition
s 2.25.1
o/python3/dist-packages, outside environment /usr
were found to uninstall.
-3.10.0 charset-normalizer-3.1.0 requests-2.28.2
can result in broken permissions and conflicting behaviour with the s
to use a virtual environment instead: https://pip.pypa.io/warnings/v
pip install PyQt5
/usr/lib/python3/dist-packages (5.15.6)
<13,>=12.8 in /usr/lib/python3/dist-packages (from PyQt5) (12.9.1)
can result in broken permissions and conflicting behaviour with the s
to use a virtual environment instead: https://pip.pypa.io/warnings/v

2019 asrInterface.py
-rw-rw-r-- 1 river river 451 Apr 11 2019 asr.py
-rw-rw-r-- 1 river river 861454 Mar 25 2019 filcapae.wav
-rw-rw-r-- 1 river river 4383 Mar 27 2019 guessTheWord.py
drwxrwxr-x 2 river river 4096 Apr 11 2019 icon
-rwxrwx--- 1 river river 8298385 Apr 2 10:57 lab2-asr.zip
drwxrwxr-x 2 river river 4096 Apr 11 2019 __pycache__
-rw-rw-r-- 1 river river 382 Mar 26 2019 test.py
drwxrwxr-x 5 river river 4096 Mar 26 2019 venv
river@river-VirtualBox: ~/lab2_asr$ chmod u+x asr.py
river@river-VirtualBox: ~/lab2_asr$ ls -l
total 8980
-rw-rw-r-- 1 river river 3589 Apr 11 2019 asrInterface.py
-rwxrw-r-- 1 river river 451 Apr 11 2019 asr.py
-rw-rw-r-- 1 river river 861454 Mar 25 2019 filcapae.wav
-rw-rw-r-- 1 river river 4383 Mar 27 2019 guessTheWord.py
drwxrwxr-x 2 river river 4096 Apr 11 2019 icon
-rwxrwx--- 1 river river 8298385 Apr 2 10:57 lab2-asr.zip
drwxrwxr-x 2 river river 4096 Apr 11 2019 __pycache__
-rw-rw-r-- 1 river river 382 Mar 26 2019 test.py
drwxrwxr-x 5 river river 4096 Mar 26 2019 venv
river@river-VirtualBox: ~/lab2_asr$ ./asr.py
./asr.py: line 1: from: command not found
./asr.py: line 2: $'\r': command not found
./asr.py: line 3: from: command not found
./asr.py: line 4: import: command not found
./asr.py: line 5: $'\r': command not found
./asr.py: line 6: import: command not found
./asr.py: line 7: $'\r': command not found
./asr.py: line 8: syntax error near unexpected token `('
./asr.py: line 8: `class myWindow(QtWidgets.QMainWindow):
river@river-VirtualBox: ~/lab2_asr$ python3 asr.py
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
river@river-VirtualBox: ~/lab2_asr$ python3 --version
Python 3.10.6
river@river-VirtualBox: ~/lab2_asr$ python3 asr.py
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
```

Run test.py and guessTheWord.py

test.py

Run

In terminal, run:

```
1 chmod u+x test.py
2
3 python3 test.py
```

Result

The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'LAB2_ASR' with files like 'test.py', 'asr.py', and 'guessTheWord.py'. The main editor displays the content of 'test.py', which imports 'speech_recognition' and uses 'sr.Recognizer()' to process an audio file 'filecapae.wav'. It also includes a section for working with microphones, using 'sr.Microphone()' and 'r.listen()' to capture audio. The bottom panel shows the 'TERMINAL' output, which displays the results of running the script. The output shows the recognized text: 'john resigned accountancy john designed to cap joan designed to catch today john i'm the captain john descending path and john murray the captain of john designed to catch and john designed the cab today pope john designed to cap john resigned accountanc'. Below the terminal output, there are tabs for 'python3' and 'bash'.

Analysis

In test.py, there are two major program segments: working with audio files and working with Microphones.

In the first program segment, it reads the content in filecapae.py, and turns toward the words. Like:

```
john resigned accountancy john designed to cap joan designed to catch today john i'm
the captain john descending path and john murray the captain of john designed to
catch and john designed the cab today pope john designed to cap john resigned
accountanc
```

In the second program segment, it reads from microphones and recognizes the content of the voice.

guessTheWord.py

Run

In terminal, run:

```
1 chmod u+x guessTheWord.py
2
3 python3 guessTheWord.py
```

Result

```
I'm thinking of one of these words:
apple, banana, grape, orange, mango, lemon
You have 3 tries to guess which one.
```

```
Guess 1. Speak!
```

```
You said: one two
Incorrect. Try again.
```

```
Guess 2. Speak!
```

```
You said: oh yeah ruling
Incorrect. Try again.
```

```
Guess 3. Speak!
```

```
You said: evolution at home
Sorry, you lose!
I was thinking of 'apple'.
```

Analysis

For some unknown reason, the results of recognition are totally incorrect.

More functions for the program (asr.py)

Modifications to the codes

To accomplish the assignment, I have added three functions to myWindows class, including `play_music()`, `open_file()` and `test()`. What is more, I introduced `recognize_speech_from_mic` function in `guessTheWord.py` to help recognize user's speech. (`from guessTheWord import recognize_speech_from_mic`)

- `play_music()`: opens a music doc (`light.wav`) when the user says 'play music'. I use a playsound packet to open the `.wav` doc. So I imported this packet like: `from playsound import playsound`. And the code of opening music doc like: `playsound('./light.wav')`.

The overall codes of the function:

```

1 def play_music(self):
2     # create recognizer and mic instances
3     recognizer = sr.Recognizer()
4     microphone = sr.Microphone()
5
6     recommand = recognize_speech_from_mic(recognizer, microphone)
7     # if there was an error, stop the game
8     if recommand["error"]:
9         print("ERROR: {}".format(recommand["error"]))
10
11     # show the user the transcription
12     print("You said: {}".format(recommand["transcription"]))
13
14     if recommand["transcription"] == 'play music':
15         playsound('./light.wav')
16     else:
17         print("Sorry, I don't understand!\n")

```

- `open_file()`: opens a txt doc (`river.txt`) when the user says 'open a file'. I used the open function in python to open this file, and printed the content of the text at the terminal.

The overall codes of the function:

```

1 def open_file(self):
2     # create recognizer and mic instances
3     recognizer = sr.Recognizer()
4     microphone = sr.Microphone()
5
6     recommand = recognize_speech_from_mic(recognizer, microphone)
7     # if there was an error, stop the game
8     if recommand["error"]:
9         print("ERROR: {}".format(recommand["error"]))
10
11     # show the user the transcription
12     print("You said: {}".format(recommand["transcription"]))
13
14     if recommand["transcription"] == 'open a file':
15         f0 = open('river.txt')
16         print(f0.read())
17     else:
18         print("Sorry, I don't understand!\n")

```


- `test()` : instead of operating the command from user's speech, it recognizes a `.wav` doc (`filcapae.wav`), if it matches to a pre-text, the programme will play a piece of music (`light.wav`)

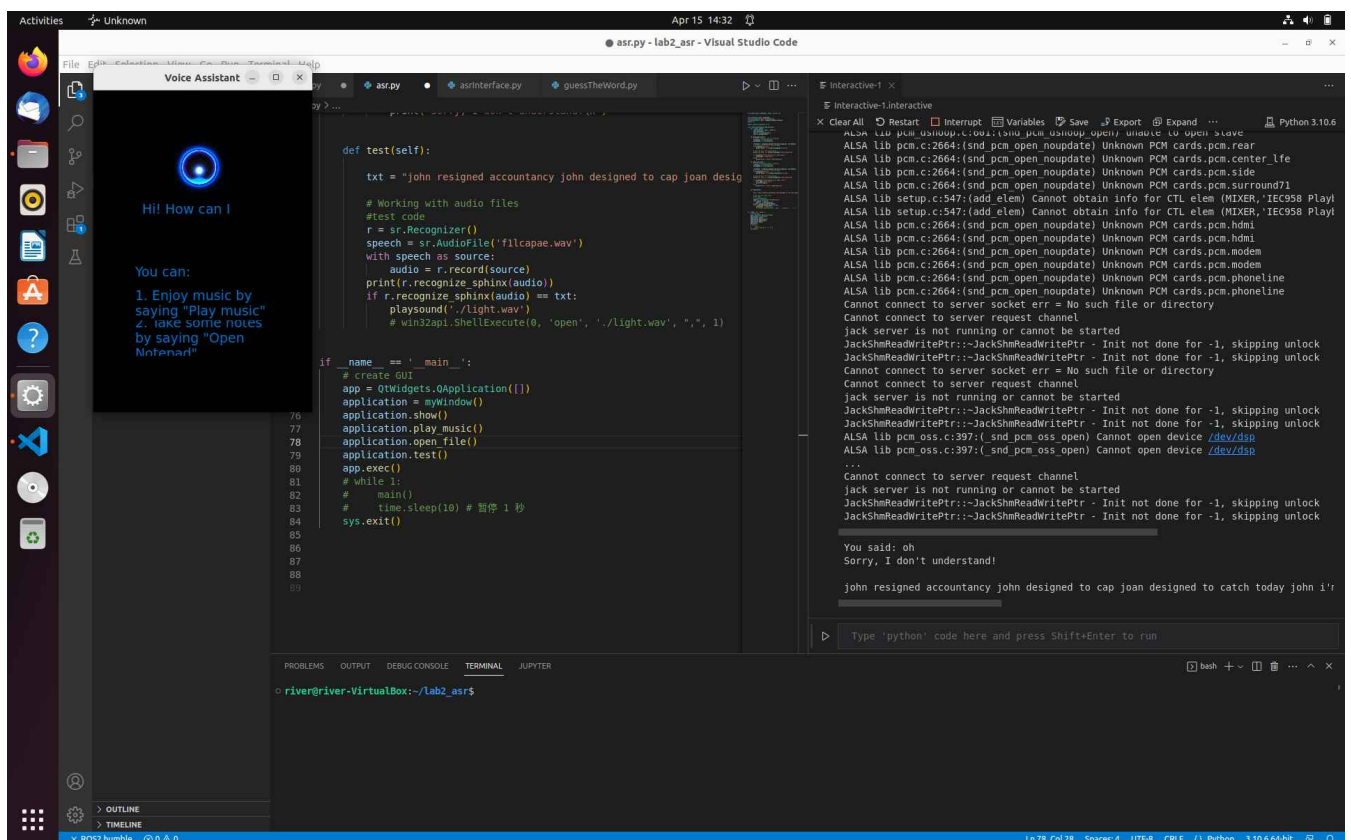
The overall codes of the function:

```

1 def test(self):
2
3     txt = "john resigned accountancy john designed to cap joan
        designed to catch today john i'm the captain john descending path and
        john murray the captain of john designed to catch and john designed the
        cab today pope john designed to cap john resigned accountancy"
4
5     # Working with audio files
6     #test code
7     r = sr.Recognizer()
8     speech = sr.AudioFile('filcapae.wav')
9     with speech as source:
10         audio = r.record(source)
11         print(r.recognize_sphinx(audio))
12         if r.recognize_sphinx(audio) == txt:
13             playsound('./light.wav')

```

The overall running result is as follows:



The accuracy of speech recognition

According to the overall running result, the accuracy is totally low. It has extremely high requirements for voice resources and sounding environment. For example, when I use the sample (`f11capae.wav`), the recognized text is basically correct, as this sample is real clear without any interference. However, when I tried to use the voice resource recording by myself via my mobilephone, the recognition failed.

On the other hand, when it comes to working with Microphones, the recognized texts are basically incorrect (do not match to the original voice's content). This situation does not occur in my modifications, but also in `test.py` and `guessTheWorld.py`

The viable solutions may include:

- Improve the quality of speech (choose a quiet environment or use more advanced equipment to record the voice).
- Optimize the algorithm of speech recognition (or use another speech recognition interface in our programme).