Lab1: Automatic Speech Recognition

1854116
Mingzhi Zhu

Xiaosi Voice Assistant
May 14, 2021

Contents

1 Modifications

- After two weeks of structural analysis of asr.py/asrinterface.py, the user interface was partially changed in this lab, and Qt Designer was used to layout the interface.
- Now the new program provides two ways to wake up the voice assistant,one is clicking the Siri iron and the other is saying something when Siri iron is moving.
- This program uses Baidu speech recognition API instead of PocketSphinx.

1.1 GUI

1.1.1 Window Icon and Name

I changed the name and icon of the windows to make them more beautiful and conform to the overall style of Xiaosi voice assistant.The picture is shown in **figure1**.

Figure 1: Window Icon and Name

1.1.2 Main Interface

Main interface is also startup interface.The top-down composition of the user interface is described below.

- **Windows Name:**Define the window name of app.
- **Title:**Explain to the user that this is a helper program.
- **Clickable Button:**A gif file called *siri-ianzhao.gif* in folder *assets*.
- **Text Echo:**Show state of the assistant. Such as listening,pardon or what user said.
- **Tips:**Prompt the user to use the voice assistant by saying the following command.

The specific interface is shown in **4**.

1.1.3 Wake-up Logic

You have two ways to activate the assistant.

- **Click Button:**Click the Siri icon on the interface to wake up voice assistant.The iron is shown in **figure2**.

Figure 2: Button Icon

- **Say Directly**Say someting to wake up voice assistant,for example you can say "soushuozhong-wen" and the result is shown in **figure3**.

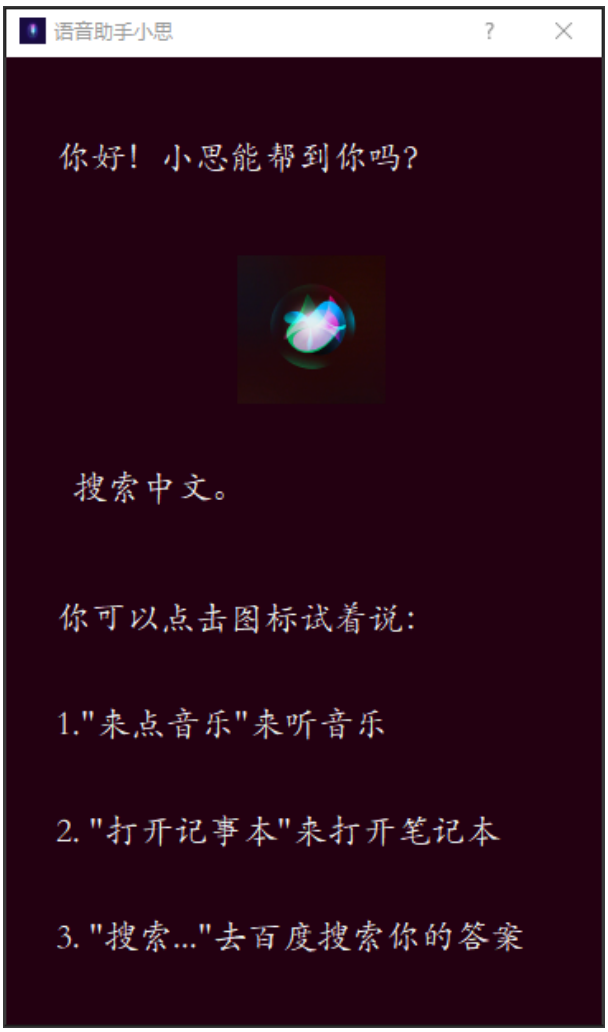

Figure 3: Say Something

1.2 Code

1.2.1 Interface code

- **QLabel**

```
def retranslateUi(self, Dialog):
    _translate = QtCore.QCoreApplication.translate
    Dialog.setWindowTitle(_translate("Dialog", "语音助手小思"))
    self.label_4.setText(_translate("Dialog", "你好！小思能帮到你吗？"))
    self.label_3.setText(_translate("Dialog", "你可以点击图标试着说："))
    self.label.setText(_translate("Dialog", "1.\"来点音乐\"来听音乐"))
    self.label_2.setText(_translate("Dialog", "2. \"打开记事本\"来打开笔记本"))
    self.label_5.setText(_translate("Dialog", "3. \"搜索...\"去百度搜索你的答案"))
```

- **QIron**

```
Dialog.setWindowIcon(QIcon('assets/siri.gif'))
```

- **QMovie**

```
self.movie = QtGui.QMovie("assets/siri-ianzhao.gif")
self.movie.frameChanged.connect(lambda: self.voiceFig.setIcon(
    QtGui.QIcon(self.movie.currentPixmap())))
```

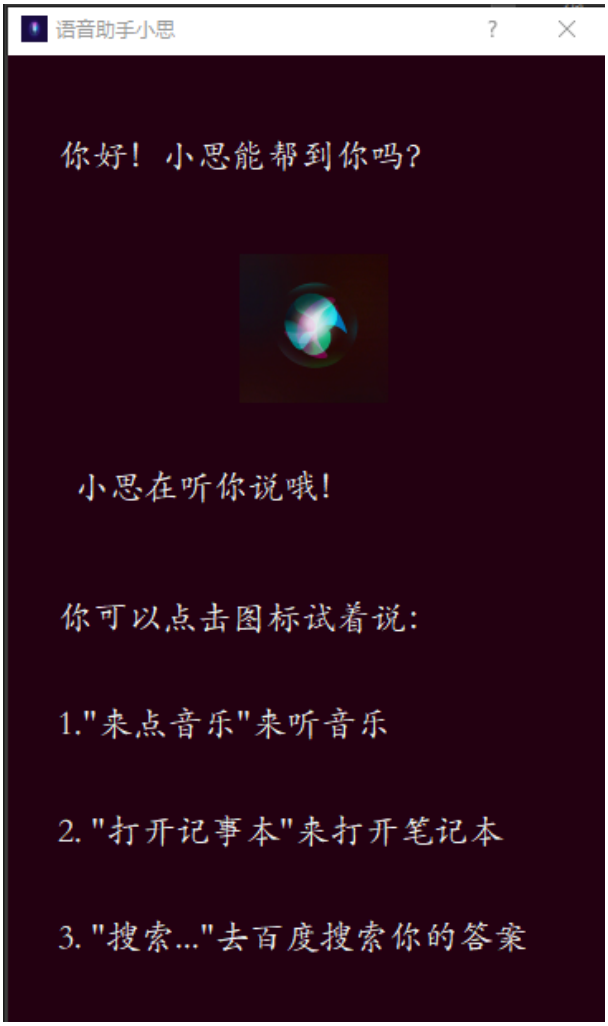

Figure 4: Main Interface

1.2.2 Speech Recognition

- **Voice Recording:**Use function in SpeechRecognition package to monitor to user's microphone. It will stop automatically when no further detection.The core code is shown below, and the detailed implementation is shown in the code file recorder.py.

```
for i in range(self.rate * time_sec // self.CHUNK):
    data = stream.read(self.CHUNK)
    frames.append(data)
```

```
stream.stop_stream()
stream.close()

frames_data = b''.join(frames)

if save_file != None:
    with wave.open(save_file, 'wb') as wf:
        wf.setnchannels(self.channels)
        wf.setsampwidth(self.pa.get_sample_size(self.format))
        wf.setframerate(self.rate)
        wf.writeframes(frames_data)

return frames_data
```

- **Voice Recognition:**Use BaiduAip to recognize the voice.Register an account on Baidu AI console, apply for a key for speech recognition, and call the interface provided by Baidu locally.The core code is shown below, and the detailed implementation is shown in the code file asr.py.

```
APP_ID = '24156401'
API_KEY = 'E0IoR5KgwaxNfBdLWsimRi34'
SECRET_KEY = 'ZfmpKut46SnGap4VgvXozy5u5TdnAnQB'
client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
def baidu_asr(wave, format, rate):
        return client.asr(wave, format, rate, {'dev_pid': 1537, })
```

1.2.3 Command Execution

- **Keyword Retrieval:**Retrieve the keywords in the text returned by Baidu API. If the keywords are consistent with the keywords in the command, execute the command.The core code is shown below, and the detailed implementation is shown in the code file app.py.

```
try:
    if '音乐' in text or '来点' in text:
        play_music()
    elif '文本' in text or '记事本' in text:
        open_file()
    elif '浏览器' in text:
        open_browser()
    elif '搜索' in text:
        keyword = text[text.index('搜索') + 2:]
        search(keyword)
    else:
        self.ui.label_6.setText('小思没听清能再说一遍嘛？')
```

- **Not Continuous Execution:**After recognizing the voice and retrieving the voice for 2S, the next voice command execution starts. The purpose of this operation is to give users enough time to get the text feedback on the interface.The core code is shown below, and the detailed implementation is shown in the code file app.py.

```
self.timer.setInterval(2000)
self.timer.start()
```

1.2.4 Assistant Functions

I add four functions to the program,they are:

- Play Music
- Open File
- Search on Baidu
- Open Browser

The core code is shown below, and the detailed implementation is shown in the code file app.py.

```
import win32api

def play_music():
    win32api.ShellExecute(0, 'open', '林海 - 流动的城市.mp3', '', 'assets', 1)

def open_file():
    win32api.ShellExecute(0, 'open', 'notepad.exe', '', '', 1)

def search(keyword):
    win32api.ShellExecute(0, 'open', f'https://www.baidu.com/s?wd={keyword}', '', '', 1)

def open_browser():
    win32api.ShellExecute(0, 'open', "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe", '', '', 1)
```

2 Discussion

2.1 Accuracy of Speech Recognition

The speech recognition effect of Sphinx interface is very poor. Although it supports Chinese recognition, it can hardly recognize the user's command accurately, so I use Baidu speech recognition API instead.The advantage of Baidu API is that it has complete documentation to teach you how to call their interfaces in Python programs. And the recognition accuracy is 95% according to the official documents. In the actual experience, the API recognition result is very accurate.Baidu API recognition speed is also very fast,I can hardly feel that this is an online interface.

But Baidu API also has its shortcomings,Baidu API only has limited call opportunities, and Sphinx is a completely free api.Baidu API is an online interface, once the computer does not access to the Internet, speech recognition programs can not run.In particular, Baidu API can not be used when the computer uses proxy IP address.

The console interface of Baidu API is shown in **figure5**.

Figure 5: Console of Baidu API

2.2 Audio Recording Quality

Pyaudio package supports multi sampling rate and multi-channel audio format, while SpeechRecognition package only supports mono audio.At the same time, SpeechRecognition package supports silent automatic stop, while PyAudio needs to set recording time manually.

So I use the **adjust_for_ambient_noise** function in PyAudio called every 0.5 seconds to ensure the audio recording quality.In addition, I set the sampling rate to 16000 while mono recording.

References

[1] Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th edition,Ben Shneiderman,Catherine Plaisant,Maxine Cohen

[2] Baidu Speech Recognition Technology Document,`https://cloud.baidu.com/doc/SPEECH/index.html`