

程序设计II 语音识别/合成部分

Python的依赖库中语音识别软件包

- apiai
- google-cloud-speech
- pocketsphinx
- SpeechRecognition
- watson-developer-cloud
- wit

其中SpeechRecognition专注于语音向文本的转换。而wit 和 apiai 提供了一些超出基本语音识别的内置功能，如识别讲话者意图的自然语言处理功能，甚至一个较为成熟的聊天交互平台。由于本实验只需实现将语音输入转为文本格式即可，因此选用SpeechRecognition库。同时，还需安装PyAudio库来获取麦克风输入。

SpeechRecognition

安装

使用 pip 命令从终端安装 SpeechRecognition：

```
pip install SpeechRecognition
```

```
C:\Users\86138>pip install SpeechRecognition
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.8.1-py2.py3-none-any.whl (32.8 MB)
    ----- 32.8/32.8 MB 6.1 MB/s eta 0:00:00
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.8.1
```

Recognizer API 主要目的是识别语音，每个 API 都有多种设置和功能来识别音频源的语音，分别是：

- recognize_bing(): Microsoft Bing Speech
- recognize_google(): Google Web Speech API
- recognize_google_cloud(): Google Cloud Speech - requires installation of the google-cloud-speech package
- recognize_houndify(): Houndify by SoundHound
- recognize_ibm(): IBM Speech to Text
- recognize_sphinx(): CMU Sphinx - requires installing PocketSphinx
- recognize_wit(): Wit.ai

以上七个中只有 recognition_sphinx () 可与CMU Sphinx 引擎脱机工作，其他六个都需要连接互联网。因此我们使用PocketSphinx库。

安装PocketSphinx库：

```
pip install PocketSphinx
```

英文文本转换

现在即可进行测试：

```
import speech_recognition as sr

# obtain audio from the microphone
r = sr.Recognizer()
harvard = sr.AudioFile(r"C:\Desktop\start.wav")
with harvard as source:
    audio = r.record(source)
# recognize speech using Sphinx
try:
    print("Sphinx thinks you said " + r.recognize_sphinx(audio))
except sr.UnknownValueError:
    print("Sphinx could not understand audio")
except sr.RequestError as e:
    print("Sphinx error; {0}".format(e))
```

中文文本转换

默认的r.recognize_sphinx(audio)是对输入的音频进行英语翻译，如果想要使用普通话版本的翻译，可到官方下载普通话语音包（<http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/>）

Portuguese	2019-08-23	16
Mandarin	2019-02-15	97
Archive	2019-02-15	34

1. 下载cmusphinx-zh-cn-5.2.tar.gz并解压
2. 在python安装目录下找到Lib\site-packages\speech_recognition
3. 点击进入pocketsphinx-data文件夹，并新建文件夹zh-CN。在这个文件夹中添加进入刚刚解压的文件，需要注意：把解压出来的zh_cn.cd_cont_5000文件夹重命名为acoustic-model、zh_cn.lm.bin命名为language-model.lm.bin、zh_cn.dic中dic改为dict格式。

PyAudio

PyAudio库简介

PyAudio是一个跨平台的音频处理工具包，使用该工具包可以在Python程序中播放和录制音频，也可以产生wav文件等。因此我们可利用PyAudio库通过麦克风输入音频并保存为wav格式文件，再调SpeechRecognition库进行语音-文本转换。

安装

```
pip install PyAudio
```

录制音频（实例程序）

```
import pyaudio
import wave

CHUNK = 1024
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
RECORD_SECONDS = 5
WAVE_OUTPUT_FILENAME = "output.wav"

p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=RATE,
                input=True,
                frames_per_buffer=CHUNK)

print("* recording")

frames = []

for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    frames.append(data)

print("* done recording")

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(CHANNELS)
wf.setsampwidth(p.get_sample_size(FORMAT))
wf.setframerate(RATE)
wf.writeframes(b''.join(frames))
wf.close()
```

播放音频（实例程序）

```
import pyaudio
import wave
import sys

CHUNK = 1024

if len(sys.argv) < 2:
    print("Plays a wave file.\n\nUsage: %s filename.wav" % sys.argv[0])
    sys.exit(-1)

wf = wave.open(sys.argv[1], 'rb')
```

```
p = pyaudio.PyAudio()

stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                channels=wf.getnchannels(),
                rate=wf.getframerate(),
                output=True)

data = wf.readframes(CHUNK)

while data != b'':
    stream.write(data)
    data = wf.readframes(CHUNK)

stream.stop_stream()
stream.close()

p.terminate()
```