

## 任务二 语音转文字

### 【任务描述】

语音识别技术需要强大的算力支撑，所以很难离线部署到本地进行应用开发，但是科大讯飞、百度等企业的开放平台提供了API免费供开发者使用。本任务用Python语言调用百度API将语音文件转换成中文。

### 【任务实施】

**步骤1** 进入百度 AI 官网[cloud.baidu.com](https://cloud.baidu.com)，注册账号和语音识别服务，创建语音识别应用。

- 进入右上角进行注册。
- 着打开左侧产品服务，选择“人工智能-语音技术”，进入页面选择创建应用。



应用

用量

请选择时间段2020-12-09 - 2020-12-09

已建应用: 1 个

管理应用

创建应用

API	调用量	调用失败	失败率	详细统计
短语音识别-中文普通话	8	0	0%	<a href="#">查看</a>
短语音识别-英语	0	0	0%	<a href="#">查看</a>
短语音识别-粤语	0	0	0%	<a href="#">查看</a>
短语音识别-普通话	0	0	0%	<a href="#">查看</a>

- 获取应用的 ID 和 Key，调用API的时候需要用到。

应用列表					
<a href="#">+ 创建应用</a>					
应用名称	AppID	API Key	Secret Key	创建时间	操作
1 Test				2020-12-06 11:07:24	<a href="#">报表</a> <a href="#">管理</a> <a href="#">删除</a>

- 领取语音识别和语音合成的免费额度。

语音识别和语音合成都要领取								
短语音识别	API	状态	调用量限制	开发限制	开通付费	购买次数包	购买开发叠加包	模型训练优化
中文普通话		● 待开通付费	总量5万次赠送	2	开通付费	购买	购买   配额详情	立即训练
英语		● 待开通付费	--	2	开通付费	购买	购买   配额详情	--
粤语		● 待开通付费	总量2万次赠送	2	开通付费	购买	购买   配额详情	--
四川话		● 待开通付费	--	2	开通付费	购买	购买   配额详情	--

## 步骤2 安装python依赖包

任务1已经安装过wave和pyaudion库了，在这个任务我们就可以直接引用，但还需要安装baiduaip和SpeechReconition包。

```
pip install baidu-aip
pip install SpeechRecognition
```

## 步骤3 学习百度的官方文档，根据官方文档简介和接口说明编写代码。

- 利用步骤1中申请的APP\_ID、APP\_KEY和SECRET\_KEY新建AipSpeech对象。

cloud.baidu.com/doc/SPEECH/s/Bk4o0bmt3

所有文档

语音技术

安装使用Python SDK有如下方式:

- 如果已安装pip, 执行pip install baidu-aip即可。
- 如果已安装setuptools, 执行python setup.py install即可。

- 读取保存的语音文件并调用asr函数实现将语音文件转换为中文。

asr函数中包括6个参数，前3个是必须要填写的，后3个选填。参考官方文档信息，第1个参数是本地读取的语音文件（格式可以是pcm、wav、amr中的一种），第2个参数第一个参数中的语音文件格式（字符串格式）、第3个参数是语音文件采样率（只能是16000或8000，知道为什么我们在任务1中采集声音用16000的采样率了吗？）。除了这3个必须要填的参数，一般我们要转换成简体中文还需要添加dev\_pid参数，参考示例添加方式为{'dev\_pid': 1537,}。

语音技术

搜索本产品文档

语音识别

导航

产品价格

短语音识别标准版

短语音识别API

短语音识别-Android-SDK

短语音识别-iOS-SDK

语音识别-LinuxC++SDK

REST-API-PHPSDK

REST-API-C++SDK

REST-API-C#SDK

REST-API-NodeSDK

REST-API-PythonSDK

简介

快速入门

接口说明

错误信息

REST-API-JavaSDK

短语音识别极速版

实时语音识别

音频文件转写

开发工具

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例，要对段保存有一段语音的语音文件进行识别：

```
# 读取文件
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 识别本地文件
client.asr(get_file_content('audio.pcm'), 'pcm', 16000, {
    'dev_pid': 1537,
})
```

参数	类型	描述	是否必须
speech	Buffer	建立包含语音内容的Buffer对象，语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见本节开头的表格	否
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文(zh)。中文=zh、粤语=ct、英文=en，不区分大小写。	否

- 编写代码，将任务1的录音功能封装到rec（）函数，导入步骤2安装的库，创建AipSpeech对象（client），调用方法asr方法，将录音转换成文本存储到result。

```
from aip import AipSpeech
import speech_recognition as sr
from playsound import playsound
import time

time = time.perf_counter
# 填写自己申请的语音识别应用ID、KEY
APP_ID = '*****'
API_KEY = '*****'
SECRET_KEY = '*****'
file_Path = 'recording.wav'
client = AipSpeech(APP_ID,API_KEY,SECRET_KEY) #新建一个AipSpeech对象

# 录音
def rec():
    rec_audio = sr.Recognizer()
    with sr.Microphone(sample_rate=16000) as source:
        print('开始录音')
        audio = rec_audio.record(source,duration=10)
    with open(file_Path,'wb') as fw:
        fw.write(audio.get_wav_data())
        print('结束录音')

# 读取文件
def get_file_content(file_Path):
    with open(file_Path,'rb') as f:
        return f.read()

rec() #开始录音
# 识别本地语音文件
result = client.asr(get_file_content(file_Path),'wav',16000,{'dev_pid':1537})
```

**步骤4** 从result中提取转换后的文本并输出到屏幕。

返回样例的信息是数据字典类型，其中key为“result”中存储了语音转换后的文本内容，其中“result”的数据类型是列表，所以result[‘result’][0]就是我们语音对应的文本内容。

返回样例：

```
// 成功返回
{
    "err_no": 0,
    "err_msg": "success.",
    "corpus_no": "15984125203285346378",
    "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
    "result": ["北京天气"]
}

// 失败返回
{
    "err_no": 2000,
    "err_msg": "data empty.",
    "sn": null
}
```

```
print(result['result'][0])
```

**步骤5** 在输出后文本后，播放录音进行校对。

支持python3的playsound模块可以很方便的播放wav、mp3等格式的音频文件，安装和import playsound模块后，利用playsound函数就可以了。

```
playsound(file_Path)
```

完成以上步骤后，电脑就可以实现让输入声音，输出你说话的内容文本和录音了，完整代码如下。

```
from aip import AIPSpeech
import speech_recognition as sr
from playsound import playsound
import time

time = time.perf_counter
# 填写自己申请的语音识别应用ID、KEY
APP_ID = '*****'
API_KEY = '*****'
SECRET_KEY = '*****'
file_Path = 'recording.wav'
client = AipSpeech(APP_ID,API_KEY,SECRET_KEY) #新建一个AipSpeech对象

# 录音
def rec():
    rec_audio = sr.Recognizer()
    with sr.Microphone(sample_rate=16000) as source:
        print('开始录音')
        audio = rec_audio.record(source,duration=10)
    with open(file_Path,'wb') as fw:
        fw.write(audio.get_wav_data())
    print('结束录音')

# 读取文件
def get_file_content(file_Path):
    with open(file_Path,'rb') as f:
```

```
        return f.read()

rec() #开始录音
# 识别本地语音文件
result = client.asr(get_file_content(file_Path), 'wav', 16000, {'dev_pid': 1537})
print(result['result'][0])
playsound(file_Path)
```

本任务实战代码如下,位于/xm1/rw2.ipynb 同学们来运行一下吧

【任务拓展】

任务2中我们完成了普通话的文本转换功能。那么英语、四川话、粤语是否可以转换呢？我们一起来查看百度语音识别技术中的接口说明里面的dev\_pid参数说明。1537对应普通话，1637对应粤语，还可以对远场录音进行普通话文本转换。

dev\_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(大陆普通话)	标准普通话模型	有标点	大陆普通话模型

