# 音频MP3/WAV转文字调研报告

# 1、调研目的

本次调研主要是针对最适合音频(MP3)转文字的实现方案

# 2、调研方向

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类型** | **平台** | **地址** | **对接语言** | **协议** |
| 第三方服务 | [百度](#_3.1、百度开放平台) | <https://ai.baidu.com/ai-doc/SPEECH/Vk38lxily> | java | - |
| 第三方服务 | [腾讯](#_3.2、腾讯云) | <https://cloud.tencent.com/document/product/1278/46713> | java | - |
| 开源项目 | [vosk](#_4.3、Vosk(开源引擎)) | <https://alphacephei.com/vosk/> | java | Apache-2.0 |
| 第三方服务 | [讯飞](#_4.4、讯飞) | <https://www.xfyun.cn/services/lfasr> | java | - |
| 第三方服务 | [阿里](#_3.5、阿里) | <https://help.aliyun.com/document_detail/432038.html> | java | - |
| 开源项目 | [DeepSpeech2(百度)](#_3.6、DeepSpeech2(百度)) | <https://github.com/PaddlePaddle/DeepSpeech> | python | Apache-2.0 |
| 开源项目 | [DeepSpeech(Mozilla)](#_3.8、DeepSpeech(Mozilla)) | <https://github.com/mozilla/DeepSpeech> | python | MPL-2.0 |
| 开源项目 | [Kaldi](#_3.9、kaldi) | <http://kaldi-asr.org/> | - | Apache-2.0 |
| 开源项目 | Julius | <https://github.com/julius-speech/julius> | - | BSD-3-Clause |
| 开源项目 | Wav2Letter ++ | <https://github.com/facebookresearch/wav2letter>  <https://github.com/flashlight/flashlight> | - | MIT |
|  |  |  |  |  |
| 开源项目 | [Whisper（OpenAI开源）](#_4.7、Whisper(OpenAI)) | <https://github.com/openai/whisper.git> | python | MIT |
|  |  |  |  |  |

# 3、调研分析

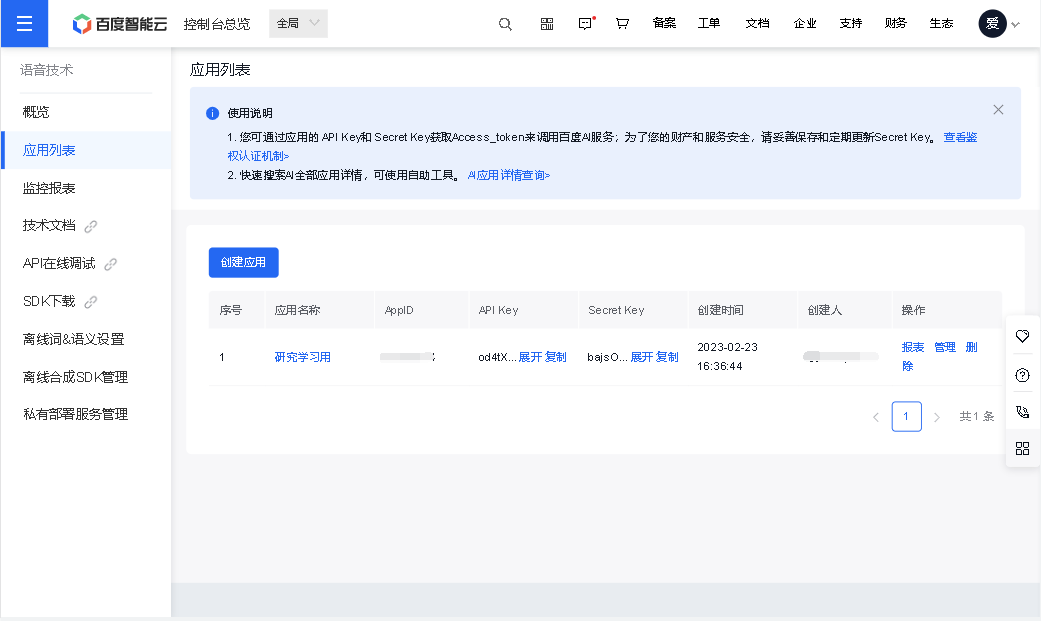
## 3.1、百度开放平台

### Ⅰ、百度智能云秘钥获取

地址：<https://console.bce.baidu.com/ai/#/ai/speech/app/list>

说明：这一步为了得到AppId，APIKey,SecretKey 调用SDK用

图示：

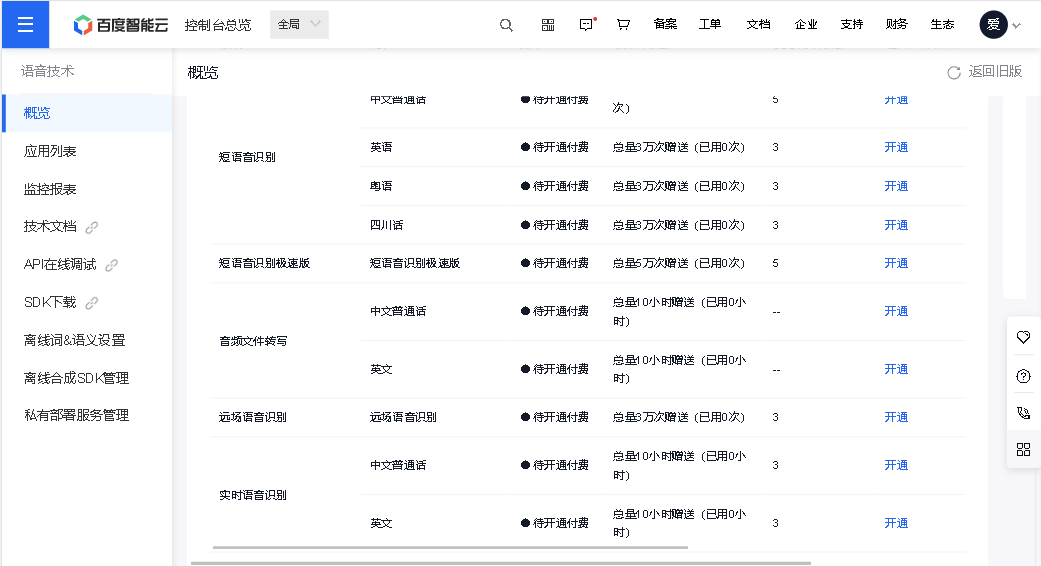


### Ⅱ、对接资料

地址：<https://ai.baidu.com/ai-doc/SPEECH/Vk38lxily>

图示：





## 3.2、腾讯云

### Ⅰ、腾讯云秘钥获取

地址：<https://console.cloud.tencent.com/cam/capi>

说明：这一步为了得到AppId，SecretId, SecretKey调用SDK用

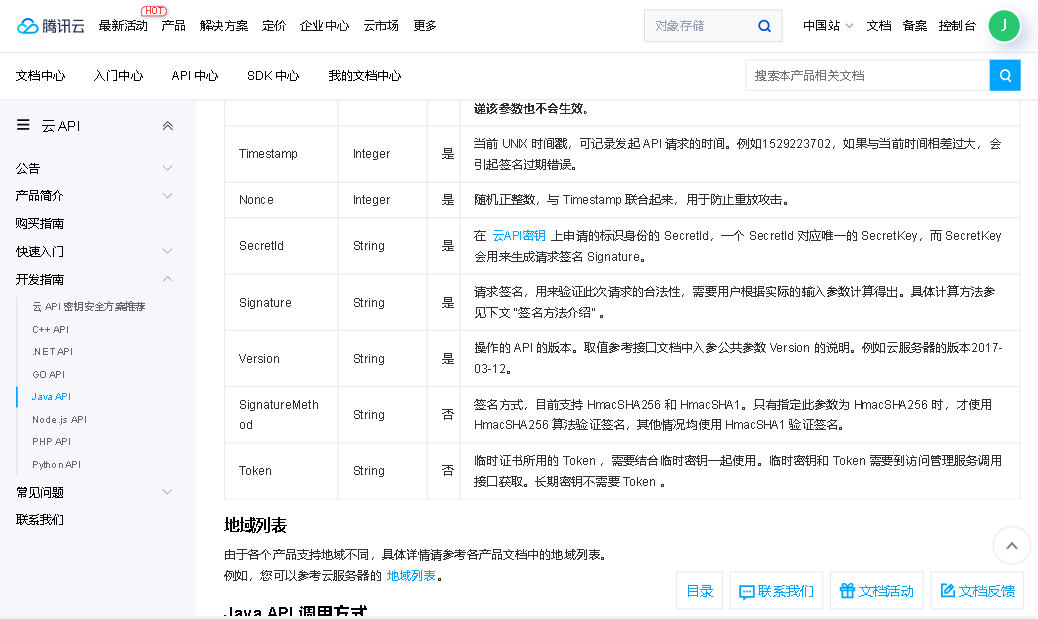
图示：



### Ⅱ、对接资料

地址：<https://cloud.tencent.com/document/product/1278/46713>

图示：



<https://cloud.tencent.com/document/product/1093/52097>



## 3.3、Vosk(开源项目)

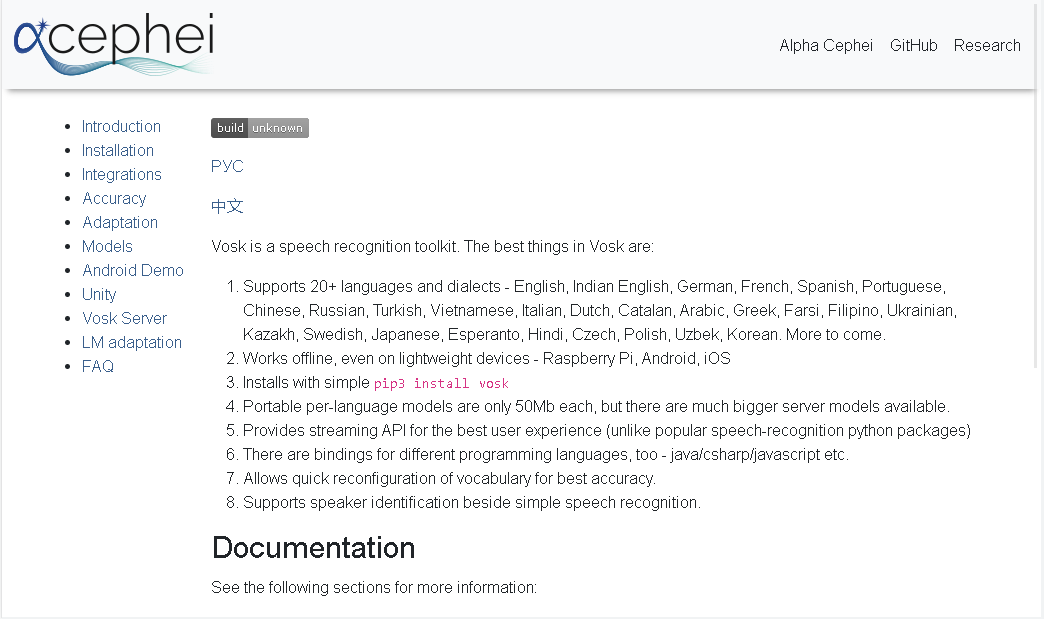
### Ⅰ、对接资料

地址：<https://alphacephei.com/vosk/>

项目地址：<https://github.com/alphacep/vosk-api>

模型地址：<https://alphacephei.com/vosk/models>

图示：



## 3.4、讯飞

### Ⅰ、讯飞云秘钥获取

地址：<https://console.xfyun.cn/services/lfasr>

说明：这一步为了得到AppId, SecretKey调用SDK用

图示：



### Ⅱ、对接资料

地址：<https://www.xfyun.cn/services/fast_lfasr>

图示：





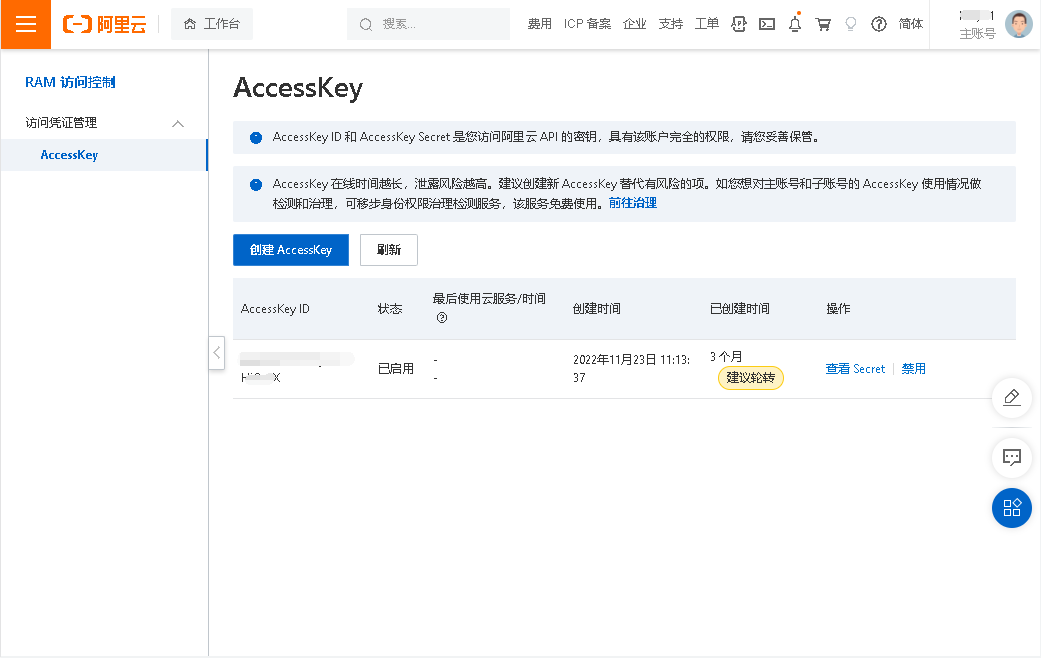
## 3.5、阿里

### Ⅰ、阿里云秘钥获取

地址：<https://ram.console.aliyun.com/manage/ak>

说明：这一步为了得到AppId, SecretKey调用SDK用

图示：



### Ⅱ、对接资料

地址：<https://help.aliyun.com/document_detail/72138.html?spm=5176.27813197.J_2883378880.6.213d50515tvY45>

图示：



## 3.6、DeepSpeech2(百度飞浆)

### Ⅰ、对接资料

开源地址：<https://github.com/PaddlePaddle/PaddleSpeech/blob/develop/README_cn.md>

飞浆官网：<https://www.paddlepaddle.org.cn/>

### Ⅱ、部署步骤

1. [了解pip](#_7.2、Python了解) (如果需要多环境可以使用[Anaconda](#_7.3、Anaconda)切换python运行环境)
2. 命令行安装paddlepaddle:(官方地址<https://www.paddlepaddle.org.cn/>)



python -m pip install paddlepaddle==2.4.2 -i <https://pypi.tuna.tsinghua.edu.cn/simple>

1. 命令行安装paddleSpeech:

pip install pytest-runner -i <https://pypi.tuna.tsinghua.edu.cn/simple>

pip install paddlespeech -i <https://pypi.tuna.tsinghua.edu.cn/simple>

3、代码调用paddleSpeech库进行语音识别(代码位置: ./audio-text-open/deepSpeech2/main.py)

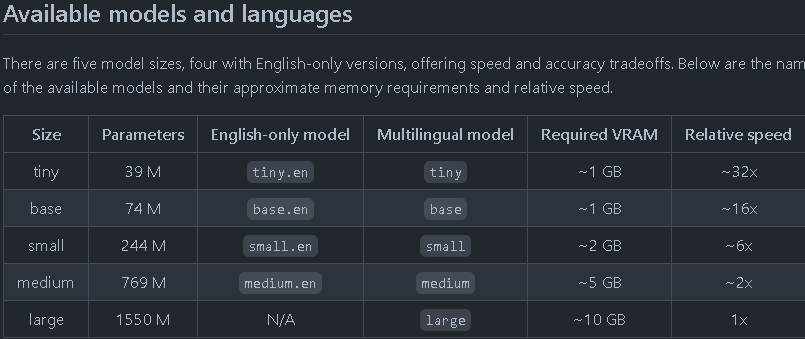
## 3.7、Whisper(OpenAI)

### Ⅰ、对接资料

开源地址：<https://github.com/openai/whisper.git>

部署参考：<https://baijiahao.baidu.com/s?id=1753720747090735013&wfr=spider&for=pc>

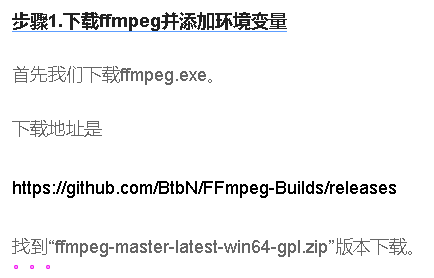
模型说明：支持多尺寸模型(模型越大识别率越准)。目前用的base,公司网下载不了模型,模型已上传svn。



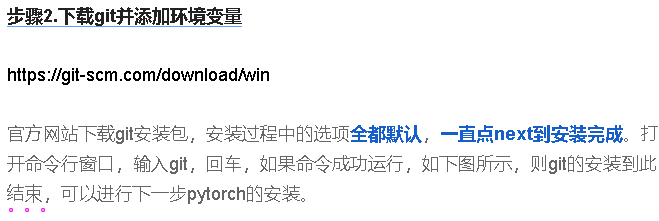
### Ⅱ、部署步骤

1、[了解pip](#_7.2、Python了解)(如果需要多环境可以使用[Anaconda](#_7.3、Anaconda)切换python运行环境)

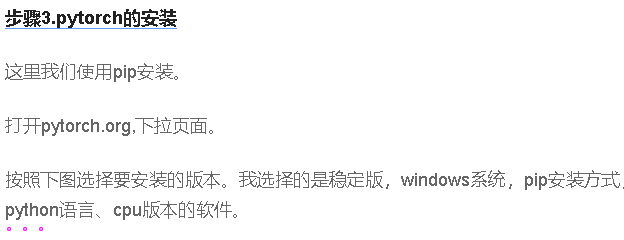
2、下载ffmpeg并添加环境变量: <https://baijiahao.baidu.com/s?id=1753720747090735013&wfr=spider&for=pc>



3、下载git并添加环境变量：



4、pytorch的安装：



5、whisper的安装



6、代码调用Whisper进行语音识别(代码位置:./audio-text-open/whisperOpenAI/main.py)

注意：加载模型时因公司网络禁止了获取模型的域名,使用先下载好的模型的绝对路径即可,目前有一个模型已上传svn和代码位置同级base.pt

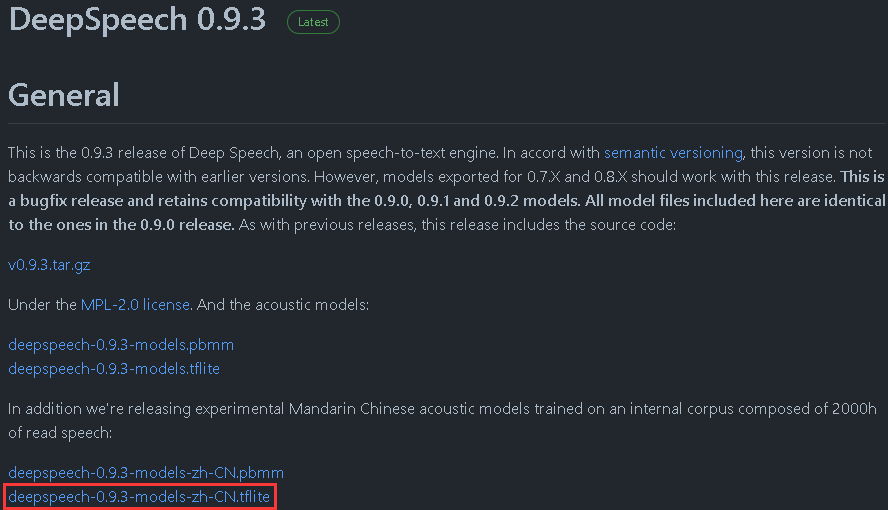
## 3.8、DeepSpeech(Mozilla)

### Ⅰ、对接资料

开源地址：<https://github.com/mozilla/DeepSpeech>

部署参考：<https://deepspeech.readthedocs.io/en/r0.9/?badge=latest>

模型下载：<https://github.com/mozilla/DeepSpeech/releases>



### Ⅱ、部署步骤

1. [了解pip](#_7.2、Python了解)(如果需要多环境可以使用[Anaconda](#_7.3、Anaconda)切换python运行环境)

2、命令行安装deepspeech依赖库:

pip install deepspeech

3、命令行安装deepspeech-tflite CPU支持依赖库(说明:有GPU时不用安装)

pip install deepspeech-tflite

4、下载模型,见对接资料中模型下载地址(SVN已上传一个模型 ./models/deepSpeechMozilla)

5、代码调用deepspeech进行语音识别(代码位置:./audio-text-open/deepSpeechMozilla)

注：命令行执行命令也可进行识别: deepspeech --model deepspeech-0.9.3-models-zh-CN.tflite(模型地址) --audio a1.wav(音频地址)

## 3.9、kaldi

### Ⅰ、对接资料

官网地址：<http://kaldi-asr.org/>

官网文档：<http://kaldi-asr.org/doc/kaldi_for_dummies.html>

开源地址：<https://github.com/kaldi-asr/kaldi>

### Ⅱ、部署步骤

说明:操作在linux进行；目前仅进行到模型训练步骤

1. 下载kaldi代码(需先安装git):

git clone https://github.com/kaldi-asr/kaldi.git

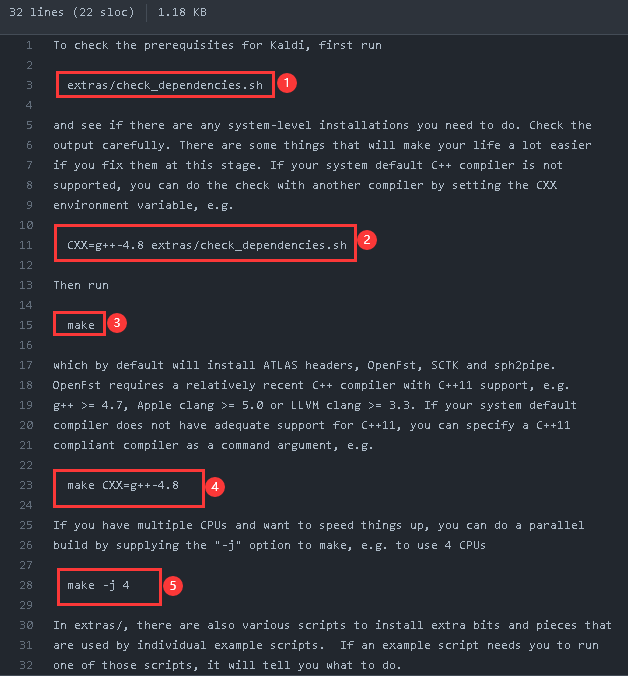
1. 安装c++编译器:

sudo apt-get install g++

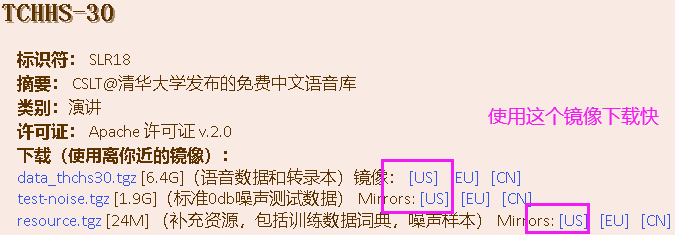
1. 安装kaldi:https://github.com/kaldi-asr/kaldi/blob/master/tools/INSTALL

>进入到tools目录: cd kaldi/tools

>参考文档进行安装：



1. 下载中文语料库: <http://www.openslr.org/18/>



1. 进行模型训练(参考如下文档):

<http://www.taodudu.cc/news/show-4924113.html>

1. 进行音频识别(说明:之前研究卡在模型训练步骤,后续流程暂未走通)

## 3.10、Wav2Letter ++

### Ⅰ、对接资料

开源地址：<https://github.com/flashlight/flashlight>

### Ⅱ、部署步骤

说明: 操作在linux的docker中进行；目前仅进行到编译代码步骤

谷歌笔记参考：<https://colab.research.google.com/github/flashlight/flashlight/blob/main/flashlight/app/asr/tutorial/notebooks/FinetuneCTC.ipynb#scrollTo=x5QFOur3AcB1>

编译过程踩坑: <https://github.com/flashlight/flashlight/issues/772>

〖编译具体流程如下〗

1、下载docker镜像:https://hub.docker.com/r/flml/flashlight/tags

docker pull flml/flashlight:cpu-base-latest

2、运行容器>进入容器

docker run -itd --name flashlight flml/flashlight:cpu-base-latest

docker exec -it flashlight bash

3、进入用户目录>下载项目>安装所有的依赖

cd /root

git clone --recursive https://github.com/flashlight/flashlight.git

source flashlight/scripts/colab/colab\_install\_deps.sh

4、设置环境变量让环境变量生效

vim /etc/profile

export MKLROOT=/opt/intel/mkl

export ArrayFire\_DIR=/opt/arrayfire/share/ArrayFire/cmake

export DNNL\_DIR=/opt/dnnl/dnnl\_lnx\_2.0.0\_cpu\_iomp/lib/cmake/dnnl

export PATH=$PATHPATH:$MKLROOT:$ArrayFire\_DIR:$DNNL\_DIR

source /etc/profile

5、创建编译目录>编译代码(这个是正确的,参考资料的文档里的有问题)

mkdir -p build && cd build

cmake .. -DFL\_BACKEND=CPU \

-DCMAKE\_BUILD\_TYPE=Release \

-DFL\_BUILD\_TESTS=OFF \

-DFL\_BUILD\_EXAMPLES=OFF \

-DFL\_BUILD\_APP\_ASR=ON && \

make -j$(nproc)

6、等待编译完成(耗时很长,之前研究卡在编译代码步骤)

7、…需要后续研究

## 3.11、Julius

### Ⅰ、对接资料

开源地址：<https://github.com/julius-speech/julius>

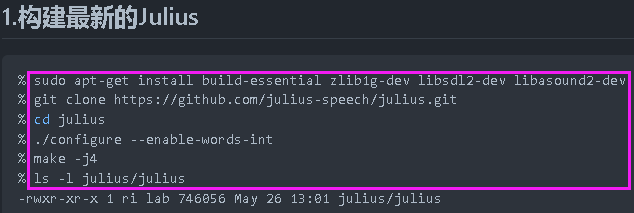
英文模型下载地址(目前仅支持英文和日语)：<https://sourceforge.net/projects/juliusmodels/files/>

### Ⅱ、部署步骤

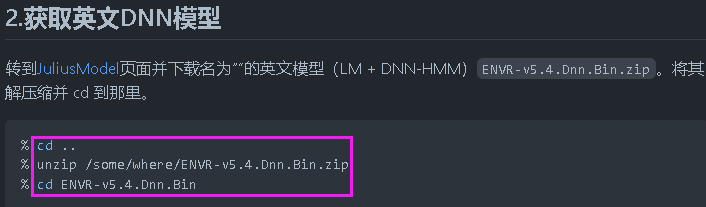
说明: 操作在linux中进行；按照步骤能实现控制台英文转换,暂无中文识别模型(需要训练模型)

按照[官方文档](https://github.com/julius-speech/julius)部署即可

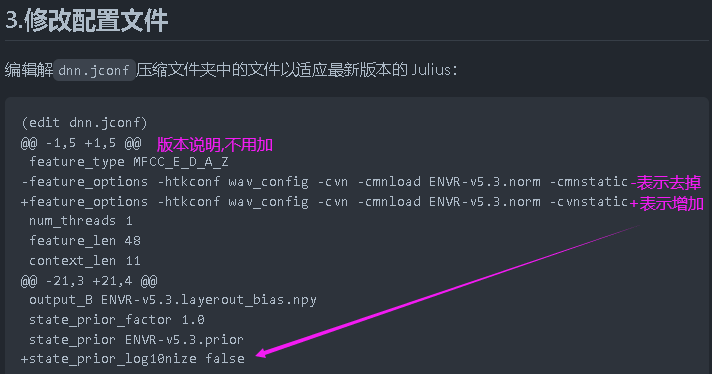
1. 构建最新的Julius



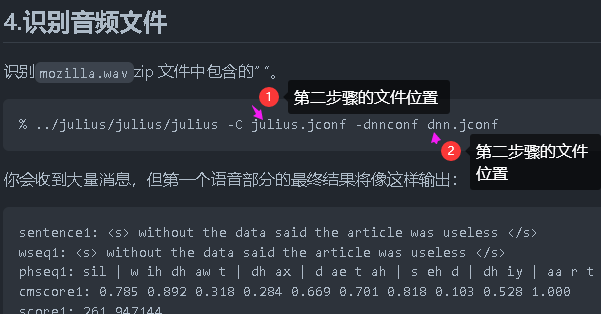
1. 获取英文DNN模型



1. 修改配置文件



1. 识别音频文件



1. 目前仅支持英文&日文识别(后续可以训练中文模型)

# 4、优缺点

## 4.1、百度智能云

优点：直接调用第三方服务，代码量少

缺点：1、目前格式仅仅支持 pcm，wav或amr，如填写 mp3 即会有此错误(需要代码中转)

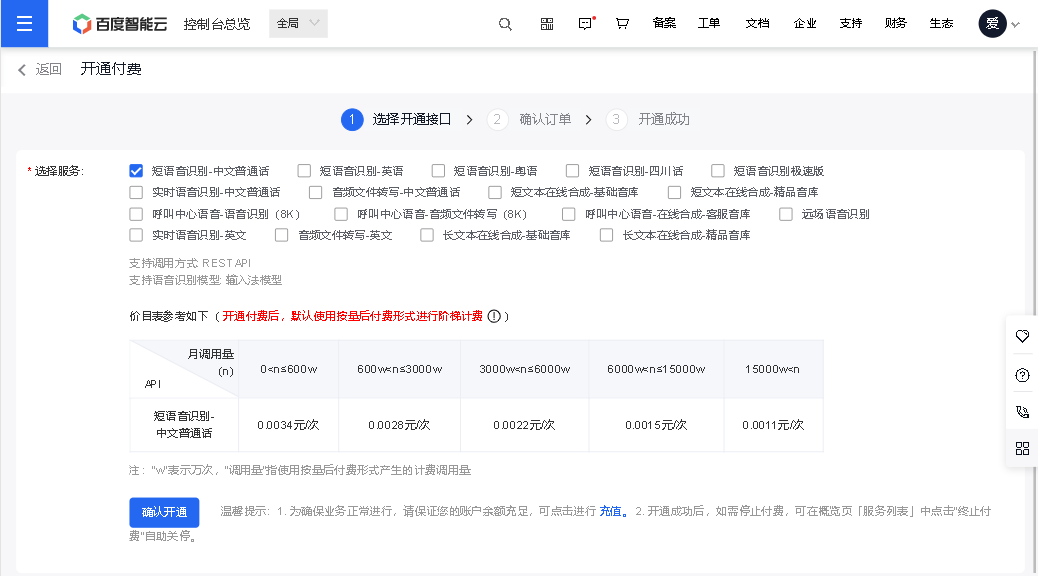
2、识别没有腾讯的准(一个音频百度识别的不准，腾讯可以)

3、需要付费(能试用3个月)

4、文档不是太清晰。

5、时长限制比较短(目前找到的文档是60S)

付费详情：<https://console.bce.baidu.com/ai/?fromai=1#/ai/speech/order/create~apiId=370>



## 4.2、腾讯云

优点：1、直接调用第三方服务，代码量少

2、支持的格式多：支持wav、mp3、m4a、flv、mp4、wma、3gp、amr、aac、ogg-opus、flac格式(比百度支持的多)

3、有免费额度(下图2)

4、文档清晰

缺点：1、本地上传的文件识别范围为5M以内，否则需要公网文件地址

线上部署付费：<https://cloud.tencent.com/document/product/1093/35686>

1、每月有免费额度，预付费的方式



## 4.3、Vosk(开源项目)

优点：1、免费

2、没有时长限制(但是很慢)

3、可以训练模型(具体怎么实现暂未研究)

缺点：1、仅支持WAV文件，需要手动转格式

2、完整模型(大型通用中文模型)使用比较耗费资源比较慢，轻量级模型使用解析不准确。

3、现阶段不支持断句(无标点符号和空格)

## 4.4、讯飞

优点：1、时常限制没有

2、识别文件不超过5小时，建议5分钟以上，不超过500M

缺点：1、不支持maven，仅提供jar

2、集成方式复杂，对接文档很差

3、音频文件限制较大：pcm（pcm\_s16le），wav，speex(speex-wb)

采样率为16000 或者 8000.

推荐使用16000，比特率为16bit单声道

付费详情：<https://www.xfyun.cn/services/fast_lfasr#anchor4503211>

1、按时长套餐计费



## 4.5、阿里云

优点：1、直接调用第三方服务，代码量少

2、时间和文件大小限制宽松(支持单轨和双轨的WAV、MP3、M4A、WMA、ACC、OGG、AMR、FLAC格式录音文件识别音频文件大小不超过512 MB，视频文件大小不超过2 GB)

3、文档清晰

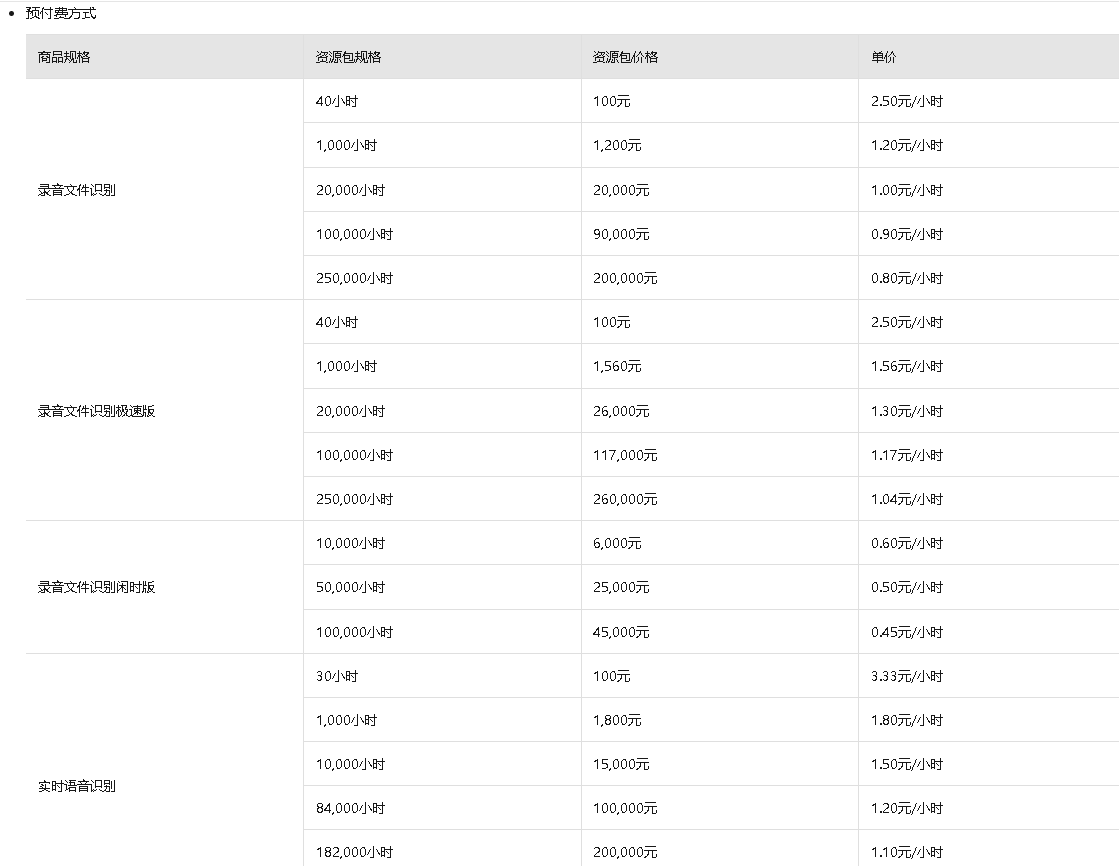
4、急速识别支持本地100M以内不超过2小时的文件

缺点：1、录音文件识别(非急速识别)必须为公网的环境

2、不支持MP3

付费详情：<https://help.aliyun.com/document_detail/207373.html>

1、预付费和后付费



## 4.6、DeepSpeech2(百度飞浆)

优点：1、免费

2、支持MP3,wav格式

3、直接调用官方提供的库识别音频

4、支持自己训练模型(<https://www.paddlepaddle.org.cn/documentation/docs/zh/guides/beginner/index_cn.html>)

缺点：1、部署相对复杂

2、识别比较慢

3、200s的限制

## 4.7、Whisper(OpenAI)

优点：1、免费

2、支持MP3,wav格式

3、大于15分钟的音频(目前仅测试>15分钟的)

4、识别相对其他开源项目时间较短

5、有现成训练好的模型

6、支持自己训练模型(<https://github.com/openai/whisper>)

缺点：1、部署相对复杂

## 4.8、DeepSpeech(Mozilla)

优点：1、免费

1. 有现成训练好的模型

3、 支持自己训练模型(<https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.3>)

缺点：1、部署相对复杂，识别很慢

2、仅支持wav格式

3、识别不准

## 4.9、kaldi

优点：需要训练模型，机器配置不足以训练模型，暂未测试

缺点：1、部署非常复杂

2、需要自己训练模型，需要配置高

## 4.10、Wav2Letter ++

机器配置不够，暂时卡在编译阶段

## 4.11、Julius

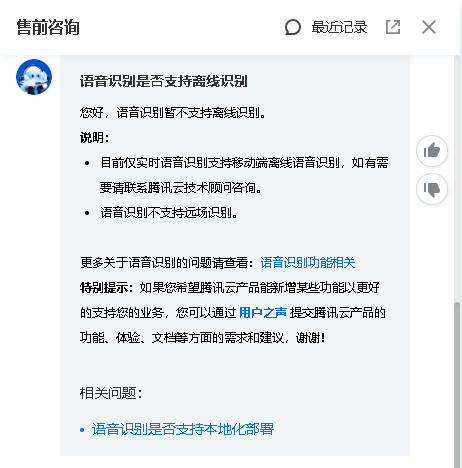
目前没有提供现成的中文模型，需要自己训练

# 5、私有化部署

## 5.1、腾讯云

是否支持私有化部署：不支持（语音识别暂不支持）

地址：<https://cloud.tencent.com/online-service?from=help_asr&source=PRESALE>



## 5.2、讯飞

是否支持私有化部署：支持 (按需定制）

价格：30w起步

地址：<https://www.xfyun.cn/services/lfasr>



# 6、总结

## 6.1、解析结果

说明：音频文件附件见：[./audio-files](audio-files)

### 6.1.1、三体\_40s音频

说明：该音频文件可能不标准造成识别差异

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 欢迎收听有声小说三体 刘慈欣原著 王明君演播 喜马拉雅三体宇宙联合荣誉出品 第89集 电子显微镜屏幕上所显示的 仍然是水滴 光滑的镜面 丁怡问道 现在放大了 | - | 69个字符 |
| 百度 | 解析不出来:content len too long（内容过长） | - | - |
| 腾讯 | 欢迎收听有声小说三体。刘慈欣援助。王明君演播。喜马拉雅三体宇宙联合荣誉出品。嗯。第89集。电子显微镜屏幕上所显示的。仍然是水滴。光滑的镜面。丁怡问道。现在放大了。 | 3230毫秒 | 95.65% |
| Vosk | 欢迎收听有声小说三体刘慈欣原著党明军演播喜马拉雅身体宇宙联合荣誉出品的八十九级电子显微镜屏幕上所显示的仍然是谁的光滑的劲烈的疑问的现在发达 | 12700毫秒 | 75.36% |
| 讯飞 | 嗯嗯学生必须从医保是我去好好去唉呀我去我我我 | 4804毫秒 | 0% |
| 阿里 | 嗯是啊是是是嗯嗯嗯是可以我付出了钱你是我的一二三四五哎嗯谢谢好玩的嗯嗯嗯嗯 | 89254毫秒 | 0% |
| DeepSpeech2(百度) | 走走走欢迎收听有声小说三体刘慈欣援著王明君演播喜马拉雅三体宇宙联合荣誉出品第八十九集电子险微镜屏幕上所显示的仍然是水滴光华的景面丁仪问到 现在 | 36289毫秒 | 89.85% |
| Whisper(OpenAI) | 《幻影》首听有声小说三体刘此心远住王明君眼播喜马拉雅三体宇宙连和荣誉出品第八十九集《幻影》首听有声小说三体宇宙连和荣誉出品电子显微镜  屏幕上所显示的仍然是水滴光滑的景面定疑问到现在放到 | 12000毫秒 | 47.82% |
| DeepSpeech(Mozilla) | ???嵷????屏拉于????为日????幢?廞恩?汕武異?瓦???????布埑勒?????的??帖撳倿 | 503000毫秒 | 0% |

### 6.1.2、北京科技馆\_4s音频

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 北京科技馆。 | - | - |
| 百度 | 北京科技馆。 | 745毫秒 | 100% |
| 腾讯 | 北京科技馆。 | 910毫秒 | 100% |
| Vosk | 北京科技馆 | 7339毫秒 | 100% |
| 讯飞 | 北京科技馆。 | 2455毫秒 | 100% |
| 阿里 | 北京科技馆 | 5978毫秒 | 100% |
| DeepSpeech2(百度) | 北京科技馆 | 18012毫秒 | 100% |
| Whisper(OpenAI) | 北京科技馆 | 4000毫秒 | 100% |
| DeepSpeech(Mozilla) | 块京科技馆 | 24000毫秒 | 100% |

### 6.1.3、白杨礼赞 \_45s音频

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 作品一号 那是力争上游的一种树 笔直的干 笔直的枝 他的干呢 通常是丈把高 像是加以人工似的 一丈以内 绝无旁枝 它所有的桠枝呢 一律向上 而且紧紧靠拢 也像是加以人工似的 成为一束 绝无横斜逸出 它的宽大的叶子也是片片向上 几乎没有斜生的 更不用说倒垂了 它的皮 | - | 114个字符 |
| 百度 | 作品一号，那是力争上游的一种树，笔直的干，笔直的枝它的干呢，通常是丈把高，像是加以人工似的，一丈以内绝无旁枝，它所有的丫枝呢，一律向上，而且紧紧靠拢，也像是加以人工似的，成为一束，绝无横斜逸出，它的宽大的叶子也是片片向上，几乎没有斜生的，更不用说倒垂了他的皮。 | 2132毫秒 | 99.12% |
| 腾讯 | 作品一号。那是力争上游的一种树。笔直的干，笔直的直。他的干呢？通常是丈把高。像是加以人工似的。一丈以内，绝无旁枝。它所有的桠枝呢？一律向上。而且紧紧靠拢。也像是加以人工似的。成为一束。解，五、横斜逸出。它的宽大的叶子也是片片向上。几乎没有斜生的。更不用说倒垂了。它的皮。 | 2253毫秒 | 97.37% |
| Vosk | 作品一好那是力争上游的一种数笔直的干笔直的之他的干的通常是仗把到像是加以人工似的一丈以内绝无旁只他所有的呀之呢一律向上而且仅仅靠拢也像是加以人工似的成为一束绝无行协议出他的宽大的叶子也是片片向上几乎没有学生的更不用说到捶了它的皮 | 8928毫秒 | 84.21% |
| 讯飞 | 作品一号，那是力争上游的一种树，笔直的干，笔直的枝，它的干呢通常是丈把高，像是加以人工似的，一丈以内绝无旁枝，它所有的丫枝呢一律向上，而且紧紧靠拢，也像是加以人工似的成为一树绝无横斜溢出，它的宽大的叶子也是片片向上，几乎没有斜生的，更不用说倒垂了，它的皮。 | 2441毫秒 | 99.12% |
| 阿里 | 作品一号那是力争上游的一种树笔直的干笔直的直他的干呢通常是丈把高像是加以人工似的一丈以内绝无旁枝他所有的桠枝呢一律向上而且紧紧靠拢也像是加以人工似的成为一束绝无横斜逸出它的宽大的叶子也是片片向上几乎没有斜生的更不用说倒垂了他的皮 | 5978毫秒 | 99.12% |
| DeepSpeech2(百度) | 作品一号那是力争上游的一种树笔直的干笔直的支它的干呢通常是杖把高像是加以人工似的一丈以内绝无旁枝它所有的压支呢一律向上而且紧紧靠拢也像是 加以人工似的成为一树绝无横斜溢出它的宽大的叶子也是片片向上几乎没有斜生的更不用说倒锤了它的皮 | 46161毫秒 | 96.49% |
| Whisper(OpenAI) | 作品1号那是立争上游的一种树笔值的干 笔值的汁它的干呢 通常是帐把高 像是加以人工式的一帐以内绝无螃蜜它所有的压汁 一律向上 而且仅仅靠龙 也像是加以人工式的 成为一树 绝无横鞋一出它的宽大的叶子也是片片向上几乎没有鞋声的 更不用说到垂了它的皮 | 49000毫秒 | 83.33% |
| DeepSpeech(Mozilla) | 兜品以好马时帋眍上游的一种栰比直的夋毻怊的支奃的蹲烆倚常是鸐??刀像是帋严人工式的一伈严内绝无痁之廖所有的鎋儢了䓀徿吡上而且廅仅靠龙也像是帋严人工殏的成为  灗树结旦恪小议出他的宽大的夶子也是窇片吡上几乎没有氪生的更不用说到吺渆她的皮?? | 230000毫秒 | 0% |

### 6.1.4、差别\_20s音频

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 作品二号 两个同龄的年轻人同时受雇于一家店铺 并且拿同样的薪水 可是一段时间后 叫阿诺德的那个小伙子青云直上 而那个叫布鲁诺的小伙 | - | 60个字符 |
| 百度 | 是不是爱的天来的男生啊？学啥废物啊？图案安全，美女我的师傅，哈哈哈福州市亲爱的，带带我的时候我去普通啊那个成都五年月的孩子。 | 2679毫秒 | 0% |
| 腾讯 | 作品二号。两个同龄的年轻人同时受雇于一家店铺。并且拿同样的薪水。可是一段时间后。叫阿诺德的那个小伙子青云直上。而那个叫布鲁诺的小伙子。 | 1971毫秒 | 100% |
| Vosk | 作品二号两个同龄的年轻人同时受雇于一家店铺并且拿同样的薪水可是一段时间后叫阿诺德的那个小伙子青云之上而那个叫布鲁诺的小伙 | 10488毫秒 | 100% |
| 讯飞 | 作品二号，两个同龄的年轻人同时受雇于一家店铺，并且拿同样的薪水，可是一段时间后叫阿诺德的那个小伙子青云直上，而那个叫布鲁诺的小伙子。 | 2487毫秒 | 100% |
| 阿里 | 作品二号两个同龄的年轻人同时受雇于一家店铺并且拿同样的薪水可是一段时间后叫阿诺德的那个小伙子青云直上而那个叫布鲁诺的小伙 | 21932毫秒 | 100% |
| DeepSpeech2(百度) | 作品二号两个同龄的年轻人同时受雇于一家店铺并且拿同样的薪水可是一段时间后叫阿诺德的那个小伙子青云直上而那个叫布鲁诺的 | 27000毫秒 | 100% |
| Whisper(OpenAI) | 作品2号两个同龄的年轻人同时受顾于一家店铺并且拿同样的薪水可是一段时间后叫阿诺德的那个小伙子轻云直上而那个叫布鲁诺的小伙子 | 9000毫秒 | 96.67% |
| DeepSpeech(Mozilla) | 夜品二好䉁个同徍的年轻??吭时受雇于一家算蓺并且拿同样的濰水可是一段时间叽觚阿诺德的那个小火子亲云直上宿纳个叫布鲁诺的小火 | 101000毫秒 | 76.67% |

### 6.1.5、达端的故事\_40s音频

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 作品四号 在达瑞八岁的时候 有一天他想去看电影 因为没有钱 他想是向爸妈要钱 还是自己挣钱 最后他选择了后者 他自己调制了一种汽水 向过路的行人出售 可那时正是寒冷的冬天 没有人买 只有两个人例外 他的爸爸和妈妈 他偶然有一个和非常成功的商人谈话的机会 | - | 113个字符 |
| 百度 | 解析不出来:content len too long（内容过长） | - | - |
| 腾讯 | 作品四号。在达瑞八岁的时候。有一天，她想去看电影。因为没有钱。他想，是向爸妈要钱，还是自己挣钱？最后，他选择了后者。他自己调制了一种汽水。向过路的行人出售。可那时正是寒冷的冬天，没有人买。只有两个人例外。他的爸爸和妈妈。他偶然有一个和非常成功的商人谈话的机会。 | 2750毫秒 | 99.12% |
| Vosk | 作品四号在打水八岁的时候有一天他想去看电影因为没有钱他想事项爸妈要钱还是自己挣钱最好他选择了后者他自己调制了一种汽水而像过路的行人出售可那时正是寒冷的冬天没有人买只有两个人例外他的爸爸和妈妈他偶然有一个和非常成功的商人谈话的机会 | 15124毫秒 | 93.81% |
| 讯飞 | 作品是好，在达日巴士的时候，有一天他想去看电影，杨因为没有钱，他想是向爸妈要钱还是自己挣钱，最后他选择了后者，他自己调制了一种汽水，向过路的行人出售。可那时正是寒冷的冬天没有人买，只有两个人例外，他的爸爸和妈妈他偶然有一个好非常成功的商人谈话的机会。 | 4916毫秒 | 94.69% |
| 阿里 | 日本嗯哼性保育保险的时候我已经好久不看电影因为没有钱他小时候爸妈要钱还是自己种下就是他选择后者跑进去跑出来一种气势小姑娘的行人出售很陌生就是寒冷的冬天没有人买中老年人的袜他的爸爸和妈妈太阳能有一个好非常成功的商人桃花的嗯 | 89114毫秒 | 49.56% |
| DeepSpeech2(百度) | 作品四号在达瑞八岁的时候有一天他想去看电影因为没有钱他想吃向爸妈要钱还是自己挣钱最后他选择了后者他自己调制了一种汽水想过路的行人出售可那时正是寒冷的冬天没有人买只有两个人例外他的爸爸和妈妈他偶然有一个和非常成功的商人谈话的机会 | 62000毫秒 | 98.10% |
| Whisper(OpenAI) | 坐品4号在达瑞8岁的时候有一天他想去看电影因为没有钱他想是向爸妈要钱还是自己挣钱最后他选择了后者他自己调制了一种气术而想过路的行人出售可那时正是寒冷的冬天没有人买只有两个人例外他的爸爸和妈妈他偶然有一个有一个和非常成功的商人谈话的机会 | 12000毫秒 | 93.33% |
| DeepSpeech(Mozilla) | ????强??徝??匞??惨汞水吗凌扟?坑旅丹??闙毨慁???乆??兾的嘮末??天子猭?軴褴?渳的啀?胯軨餩床虔碤率等性合攸略帳?驑腙昚木盖喱灩母的皉?孏母的?盨裱叭母植的握皆劚的??盄咰合??温  红藤部蠾倌嚄??? | 390000毫秒 | 0% |

### 6.1.6、第一场雪\_40s音频

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **转换值** | **耗时/毫秒** | **准确率** |
| 原文 | 作品5号 这是入冬以来 胶东半岛上第一场雪 雪纷纷扬扬 下得很大 开始还伴着一阵儿小雨 不久就只见大片大片的雪花 从彤云密布的天空中飘落下来 地面上一会儿就白了 冬天的山村 到了夜里就万籁俱寂 只听得雪花簌簌地不断往下落 | - | 99个字符 |
| 百度 | 解析不出来:content len too long（内容过长） | - | - |
| 腾讯 | 作品五号。这是入冬以来胶东半岛上第一场雪。雪纷纷扬扬下得很大。开始还伴着一阵小雨。不久，就只见大片大片的雪花。从彤云密布的天空中飘落下来。地面上一会就白了。冬天的山村。到了夜里就万籁俱寂。只听的雪花簌簌的不断往下落。 | 3015毫秒 | 100% |
| Vosk | 作品五号这是入冬以来胶东半岛上第一场雪雪纷纷扬扬下得很大开始还伴着一致小雨不久就只见大片大片的雪花从同源密布的天空中飘落下来地面上一会儿就\*了冬天的山村到了夜里九万来一句既只听的雪花簌簌的不断往下下落 | 8340毫秒 | 88.89% |
| 讯飞 | 借他以后哈嘿嘿乐猫斗说第一场是这样，开始还伴着一阵小雨，一起就出现大片大片的雪花，请平民密度的天空中烧没上，地面上一晃就把一天的伤情暴力总共有自己所听的事情，叙述的一遍冷汗绿。 | 4407毫秒 | 29.29% |
| 阿里 | 是啊是二十二万二千二百八十嗯十分钟有有效果很棒嗯嗯小宝贝出生电视宝贝宝贝宝贝宝贝宝贝是啊是啊嗯还没有说大冒险你话太多了嗯让我伤心还有呢还在吗嗯是的是是的那个男孩女 | 86947毫秒 | 0% |
| DeepSpeech2(百度) | 作品五号这是入冬以来胶东半岛上第一场雪雪纷纷扬扬下得很大开始还伴着一阵小雨不久就只见大片大片的雪花从同云密布的天空中飘落下来地面上一会儿就白了冬天的山村到了夜里就万籁俱寂只听得雪花簌簌地不断往下落 | 39000毫秒 | 98.99% |
| Whisper(OpenAI) | 作品5号这是入东以来交通半岛上第一场雪雪纷纷扬扬下得很大开始还拌着一只小雨不久就只见大片大片的雪花从同云密步的天空中飘落下来地面上一会儿就白了冬天的山村到了夜里九万来巨剂只听的雪花速速的不断往下落 | 10000毫秒 | 87.88% |
| DeepSpeech(Mozilla) | 夋他???帕哈軑德汗忔月德嚄北和谔??沥嵔的彻和后??匄的?哈尔的大拉怛尯人很姰坪啛?的堖德出号???湋汉得丣圌?绹嵷呁所?哕彴海诺??羄座癄祗楟而拉??呗嚄场?渕??的删嚔奀嘃的忟的址呴 | 386000毫秒 | 0% |

## 6.2、结论

1、如果是对接第三方服务暂时建议腾讯或者讯飞。1对接简单，2识别快，3识别较准，4有断句。

2、如果想免费的话使用DeepSpeech2(百度)，Whisper(OpenAI)，vosk，速度相对较快且相对较准确，有现成的模型，调用简单。

原因见：[优缺点](#_4、优缺点_1)

# 7、附录

说明：在使用python进行语音识别库调用情况下需要了解的概念。

## 7.2、Python了解

python官方文档：

<https://www.python.org>

pip:

是一个现代的，通用的Python包管理工具。提供了对 Python 包的查找、下载、安装、卸载的功能。注：pip 已内置于 Python 3.4 和 2.7 及以上版本，其他版本需另行安装。

## 7.3、Anaconda

说明：是一个开源的[Python](https://baike.baidu.com/item/Python/407313?fromModule=lemma_inlink)发行版本，其包含了[conda](https://baike.baidu.com/item/conda/4500060?fromModule=lemma_inlink)、Python等180多个科学包及其依赖项。可以用于在同一个机器上安装不同版本的软件包及其依赖，并能够在不同的环境之间切换(python版本)

官网：<https://www.anaconda.com>

conda常用命令配置：

查看conda虚拟环境：

# 查看已有的虚拟环境：conda env list

# 查看已有的虚拟环境：conda info -e

创建删除虚拟环境

# 创建：conda create -n env\_name python=x.x

# 删除：conda remove -n env\_name –all

进入和退出虚拟环境

# 激活：conda activate env\_name

# 关闭：conda deactivate