# 语音转文字

### 安装配置

|  |
| --- |
| 1. **安装依赖库libfftw3**   sudo apt-get install libfftw3-dev libfftw3-single3   1. **安装依赖库libopenblas**   sudo apt-get install libopenblas-dev   1. **安装python环境**   sudo apt-get install python3 python3-dev   1. **下载和更新训练模型**   **创建一个文件夹:**  mkdir ./models/paraformer\_cli  cd ./models/paraformer\_cli  **从modelscope官网下载预训练模型**  wget --user-agent="Mozilla/5.0" -c "https://www.modelscope.cn/api/v1/models/damo/speech\_paraformer-large\_asr\_nat-zh-cn-16k-common-vocab8404-pytorch/repo?Revision=v1.0.4&FilePath=model.pb"  mv repo\?Revision\=v1.0.4\&FilePath\=model.pb model.pb  **将python的模型转化为C++的**  ../scripts/paraformer\_convert.py model.pb |

### 程序需要链接的库(否则无法编译/运行)

|  |
| --- |
| 1. 需要依赖fftw3 所以需要链接 fftw3提供的库   如果没有指定目录安装,每个机器可能有一些不同  cd /  sudo find -name \*fftw3\*  查找到对应的静态库/动态库,添加到编写的Makefile中   1. 需要依赖 openblas 所以需要链接 openblas 提供的库   cd /  sudo find -name \*blas\*  查找对应的静态库/动态库,添加到编写的Makefile中   1. 需要依赖 fastasr静态库 2. 需要依赖 webrtcvad 静态库   **Makefile示例**  flags:= -I /home/chen/FastASR\_copy/include/  flags+= -L /home/chen/FastASR\_copy/build/src/lib -lfastasr -L/usr/lib/x86\_64-linux-gnu -lfftw3 -lfftw3f -lblas  flags+= -L /home/chen/FastASR\_copy/build/third\_party/webrtc/ -lwebrtcvad  src\_cpp=$(wildcard ./\*.cpp)  debug:      g++ -g $(src\_cpp) -omain $(flags) -std=c++11    **需要提供训练集**  **训练集合提取准备好(程序默认是用**paraformer**)**  **放到**models中 |

### 使用示例

|  |
| --- |
| #include <iostream>  #include <win\_func.h>  #include <Audio.h>  #include <Model.h>  #include <string.h>  using namespace std;  bool externContext(const char\* src, const char\* dst)  {      Audio audio(0);      audio.loadwav(src);      audio.disp();      Model \*mm = create\_model("/home/chen/FastASR/models/paraformer\_cli", 3);      audio.split();        setbuf(stdout, NULL);      cout << "Result: \"";      float \*buff;      int len;      int flag;      char buf[1024];        FILE\* fp = NULL;      fp = fopen(dst, "w+");      while (audio.fetch(buff, len, flag) > 0) {          mm->reset();          string msg = mm->forward(buff, len, flag);          cout << msg;          memset(buf, 0, sizeof(buf));          snprintf(buf, sizeof(buf), "%s", msg.c\_str());          fseek(fp, 0 ,SEEK\_END);          fprintf(fp, "%s", buf);          fflush(fp);      }      cout << endl;        delete mm;  }  int main()  {        externContext("/home/chen/FastASR\_copy/zh.wav", "/home/chen/FastASR\_copy/context.txt");      return 0;  } |

### 支持的文件格式

|  |
| --- |
| 只支持wav 如果想运行提取其它格式请先转换格式 **具体支持的格式为：**  温馨提示：  不是wav格式都可以识别出来，需要单声道，采样率为16000Hz。否则识别出来的准确概率为0%  方法:  ffmpeg -i video\_input.XXX video\_output.wav **多通道变单通道** ffmpeg -i input.wav -ac 1 output.wav **修改音频文件采样率：** ffmpeg -i input.wav -ar 16000 output.wav |

### 支持的语音

|  |
| --- |
| 英文+普通话 |

### 准确率

|  |
| --- |
| 根据训练模型进行特定训练，训练完之后的准确率99% |

### 效率

|  |  |  |
| --- | --- | --- |
| 5s | 5min | 30min |
| 0.695871s | 29.141491s | 186.848961s |