

中科院声学所识别引擎

概要设计说明

目录

[1.概述 5](#_Toc90302790)

[1.1简介 5](#_Toc90302791)

[1.2 读者对象 7](#_Toc90302792)

[1.3术语定义 7](#_Toc90302793)

[2.总体设计 8](#_Toc90302794)

[3.模块设计 10](#_Toc90302795)

[3.1 初始化模块 10](#_Toc90302796)

[3.1.1构建解码网络子模块 11](#_Toc90302797)

[3.2 端点检测模块 12](#_Toc90302798)

[3.2.1基于能量端点检测子模块 12](#_Toc90302799)

[3.2.2 基于谐波端点检测子模块 13](#_Toc90302800)

[3.3 特征提取模块 16](#_Toc90302801)

[3.3.1基于MFCC特征提取子模块 16](#_Toc90302802)

[3.3.2 基于PLP特征提取子模块 18](#_Toc90302803)

[3.4 解码模块 21](#_Toc90302804)

[3.5 结果后处理模块 24](#_Toc90302805)

[3.5.1音素转拼音子模块 24](#_Toc90302806)

[3.5.2 ITN子模块 24](#_Toc90302807)

[3.5.3 标点子模块 25](#_Toc90302808)

[3.5.4 热词子模块 26](#_Toc90302809)

[4.可靠性设计 27](#_Toc90302810)

[5.扩展性设计 28](#_Toc90302811)

[6.安全性设计 29](#_Toc90302812)

[7.维护性设计 30](#_Toc90302813)

[8.易用性设计 33](#_Toc90302814)

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 1.0 | 陈向东 | 龚云波  黎塔 | 2021/04/28 |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 1.概述

1

1.1简介

“好记性不如烂笔头”，反映出人们对记录的重视程度。从传统的  
纸文化进步到计算机时代，除了手写识别，当信息量更大的时候，人们  
会选择音频和视频。虽然人们不用担心存储容量不够用，但是今后如何  
有效的利用这些多媒体的记录，成为一个突出问题。

很多人为此倾向花高价邀请速录员。一来需要成本，二来在专业领  
域，培训和保密又成为新的问题。现在，用户有了新的选择-声学所识别引擎。该识别引擎不仅能保存语音，还能把语音转化成文字与之对应保存下来。有了它，今后无论何时用户想要调出所需的记录，只要通过查找相应文字并简单点击，相关的语音片断就找到了。同时，本引擎还提供了直接识别以往录音的功能，把以前宝贵的资料统统转化成文本，给用户的媒体档案库将来查询时提供最大便利。

中科院声学所中科信利连续语音识别引擎，针对连续音频流（即来自说话人直接录入的语音，或者广播电台、电视台或其他领域的音频信号）进行识别，将音频信息自动转化成文字。覆盖汉语中绝大多数词语，适用于说普通话的任何人群。输出的结果都是汉字，兼容数字。

在输入的声音中，检测出可靠的语音，排除噪声音乐等，实时送入语音识别解码器进行识别。识别引擎把音视频中提取出的语音分成 25 毫秒一帧，提取有用特征，然后识别出一些类似拼音的结果（声学模型），再根据汉语字词句之间的搭配概率（语言模型），综合考虑。当然，考虑的越多（beam路径越大），识别准确率相对提高，同时消耗的时间就增长了；所以我们有优化策略，及时排除不可能的结果，避免系统过慢。用户可以通过调节这些参数来平衡识别质量和速度，以满足实际的需要。

最后返回的识别结果，软件以汉语中基本词语为单位给出了包含对应时间点的词序列。

需要说明的是，对识别正确率以及识别速度来说，输入语音（普通话）的质量（录音过程）是很重要的。录音时应尽可能的排除噪声和音乐、增大语音，识别效果就会好得多，因此建议用户在录音的时候选用质量较好的麦克风。

另外，系统在检测是否有语音信号时可能会带来一定的时间滞后，用户在实时录入时在每句话结束后，需要略有停顿，这样可以提高系统语音检测的速度和精度。语音检测的越好。识别引擎的识别效果也就越好。

## 1.2 读者对象

本文档是对识别引擎的内部代码流程和相关技术点的整体说明，因此读者最好满足如下几点条件后再读此文档：

* 有一定的C和C++语言基础
* 了解语音识别过程和原理
* 了解声学所识别引擎实现原理

## 1.3术语定义

TBNR，Thinkit-Broadcast-News-Recongnize：中科信利识别引擎

CN，Confuse-Network：混淆网络

VAD，Speech Activity Detection：语音活动检测

WFST, Weighted Finite-State Transducer：加权有限状态转录即

ITN,Inverse –Text-Normalization：逆文本标准化

# 2.总体设计

2

声学所识别引擎的大体流程可以分为五步：

第一步：进行识别环境的初始化，构建WFST解码网络。

第二步：将输入的语音数据送入到声学所的识别引擎中进行分段处理，得到有效的分段语音数据。

第三步：将步骤一中得到的分段语音数据进行特征提取，得到分段语音对应的特征数据。

第四步：将步骤二中的特征数据进行解码，得到多个候选的识别结果。

第五步：识别结果进行相应的后处理。并通过回调函数进行返回。

识别引擎总体设计流程如图2.1所示：



图2.1 识别引擎总体设计流程

如上图所示，首先对输入的音频数据进行端点检测，获取逐句的有效语音数据，然后将其送入特征提取模块，接着对于提取的声学特征进行解码，解码过程采用transformer的decoder输出进行字符同步的束搜索算法，在搜索空间内寻找匹配概率最大的最优路径，得到识别结果。

# 3.模块设计

3

## 3.1 初始化模块

初始化模块主要任务是通过读取配置文件完成识别引擎环境的初始化，如初始化端点检测模块，初始化提取模块，初始化解码模块，初始化后处理功能模块。

初始化模块内容如图3.1所示：



图3.1 初始化流程图

### 3.1.1构建解码网络子模块

读取输入的模型，利用解码器加载相应的模型并构建解码网络。

## 3.2 端点检测模块

端点检测模块是对输入的语音数据进行分段处理，切分出有效的语音进行后面的特征提取和解码工作，该模块避免了噪音，静音等非有效片段对识别性能的影响，提高了识别结果的准确率，同时也提升了引擎实时率。

此识别引擎支持基于能量，谐波和深度学习三种方式的端点检测方法，使识别引擎在不同数据上具有很强的适用性，保证了端点检测的分段结果的可靠性。

### 3.2.1基于能量端点检测子模块

首先对输入的语音数据数据进行分帧处理，逐帧计算其时域能量，缓存一定的帧数利用自适应初始化噪音模型，重新计算各帧时域能量，当连续有效能量帧数超过设定阈值时确认出起点位置，当连续静音数据帧数超过设定阈值时确认出尾点位置。

基于能量端点检测流程图如图3.2.1所示：



### 3.2.2 基于谐波端点检测子模块

通过快速傅里叶变换FFT将语音在时域的信号变换到频域。一般正常的说话人语音可以在频谱图中看到明显的谐波，传真音和震铃这些类型的语音则没有明显的谐波结构，据此可通过分析信号在频域的谐波位置判断噪音和正常的说话人语音的起止点。谐波形式如图3.2.2.1所示

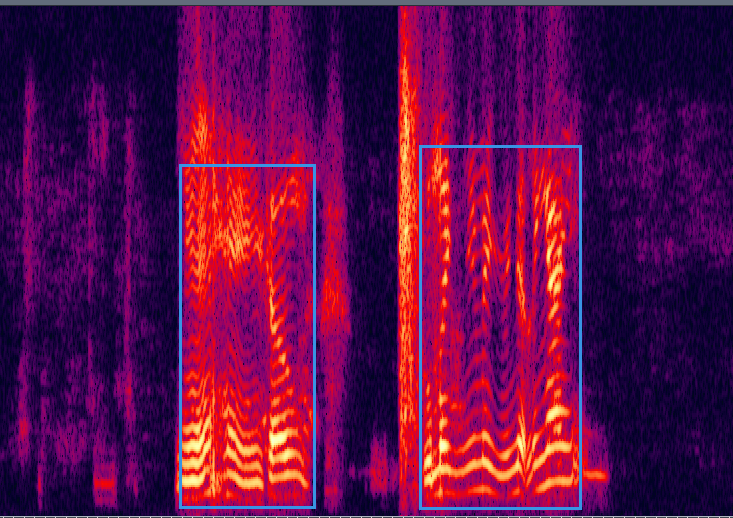


图3.2.2.1 谐波形式

基于谐波能量方式的端点检测方法的基本流程：

图3.2.2.2 端点检测流程图

## 3.3 特征提取模块

语音波形在时域上几乎没有描述能力，因此必须将波形作变换，即进行声学特征提取。特征提取模块负责计算语音的声学参数，并进行特征的计算，以便提取出反映信号特征的关键特征参数，以降低维数以便于后续处理。语音识别系统常用的特征参数有幅度、能量、过零率、线性预测系数（LPC）、LPC 倒谱系数（LPCC）、线谱对参数（LSP ）、短时频谱、共振峰频率、反映人耳听觉特征的Mel频率倒谱系数（MFCC）等。

识别模块支持感知线性预测（PLP）和Mel频率倒谱系数（MFCC）两种特征；

### 3.3.1基于MFCC特征提取子模块

梅尔频标倒谱系数(Mel Frequency Cepstrum Coefficient,MFCC)考虑了人耳的听觉特性，将频谱转化为Mel频标的非线性频谱，然后转换到倒谱域上。MFCC是采用滤波器组的方法计算出来的，这组滤波器组在频率的Mel坐标上是等宽的。这是因为人类对1000HZ以上的声音频率范围的感知不遵循线性关系而遵循在对数频率坐标上的近似线性关系。

梅尔频率倒谱特征的提取一般分为如下步骤：

* 预加重，目的是提升高频部分，使信号变得平坦，保持在低频到高频的整个频带中，能用同样的信噪比求频谱，并消除发声过程中的声带和嘴唇效应，突出高频的共振峰；
* 分帧，由于语音具有短时平稳特征，需将语音分成一个个小段，每一小段为一帧，帧长一般为25ms，相邻帧的重叠为10ms。
* 加窗，一般为方窗、汉明窗、汉宁窗等，目的是消除相邻帧两端可能造成的不连续性。
* 快速傅里叶变换FFT，将语音信号转换到频域，分析语音信号在频域的特性。
* 梅尔滤波器组，人耳对不同频率的敏感程度不同，且成非线性关系，因此我们将频谱按人耳敏感程度分为多个Mel滤波器组；在Mel刻度范围内，各个滤波器的中心频率是相等间隔的线性分布，但在频率范围不是相等间隔的，将能量谱通过一组Mel尺度的三角形滤波器组过滤后获得一个与滤波器个数相等的维数的特征。
* 离散余弦变换，大多数的自然信号（包括声音和图像）的能量都集中在离散余弦变换后的低频部分，离散余弦变换具有很强的"能量集中"特性，通过离散余弦变换对每帧数据在进行一次降维并获得音频的包络特征。

图3.2.1 MFCC特征流程

### 3.3.2 基于PLP特征提取子模块

PLP共60维度，包括1维能量，12维PLP，1维基本， 1维NCC（基频置信度），并进行三阶差分。具体实现步骤如下：

1、语音信号经过采样、加窗、离散傅立叶变换后,取短时语音频谱的实部和虚部的平方和, 得到短时功率谱P ( f ) = Rx [ X ( f ) ]2 + I m [ X ( f ) ]2

2、临界频带分析  
临界频带的划分反映了人耳听觉的掩蔽效应，是人耳听觉模型的体现。利用公式

Z( f ) = 6ln{ f / 600+ [ ( f / 600)2 + 1] 0.5 }  
将频谱P ( f ) 的频率轴f 映射到Bark 频率Z, 总共得到17 个频带。这17 个频带中每个频带内的能量谱与式( 3) 的加权系数相乘, 求和后得到临界带宽听觉谱θ( k) 。

3、等响度预加重

用模拟人耳大约40 dB 等响曲线E( f ) 对θ( k)进行等响度曲线预加重, 即

Γ( k) = E[f0(k)]θ( k) , ( k = 1, 2, ..., 17)

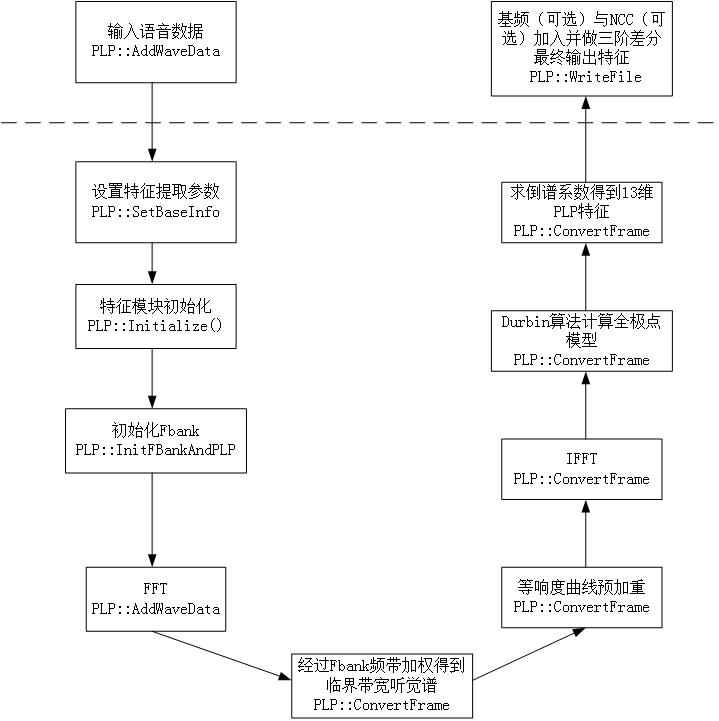
式中f0 ( k ) 表示第k 个临界带听觉谱的中心频率所对应的频率( 单位为Hz) 。其中

E[ f0(k)] =(f0(k)2 + 1. 44 \* 106 )f0(k)4/( f0(k)2 + 1.6\*105)2\*( f0( k)2 + 9.61\*109)

4、强度-响度转换

为了近似模拟声音的强度与人耳感受的响度间的非线性关系, 进行强度-响度转换θ(k) = Γ( k )0.33 经过离散傅里叶反变换后, 用德宾算法计算全极点模型, 并求出倒谱系数, 最终程序提取出0-12阶共计13维PLP特征。

5、之后可选择是否加入基频以及基频置信度（NCC），并做三阶差分，最终得到特征集。若选择加入基频以及NCC特征，则最终得到(13+2)\*4=60维特征集



## 3.4 解码模块

端到端声学模型神经网络推理介绍：

采用的是transformer结构构建的声学模型前向推理。输入的是归一化的声学特征，encoder输出是每一帧的声学embedding，声学embeding输入decoder，其中decoder是一个自回归的计算流程，最后通过decoder输出每一个character的概率。整个包括模型加载，模型分层计算，decoder自回归计算，模型定点化计算，向量矩阵计算加速，encoder-decoder计算优化，多线模型参数共享等部分。在线部分还包括截断点判决，中间隐含层结果缓存等模块。

解码中的令牌传递算法介绍：

有限状态转换机(WFST)提供了一个统一的形式来表示当前主流的大规模连续语音识别(LVCSR)系统的不同知识源(knowledge source)。表示不同知识源的多个WFST可以通过复合运算整合成一个WFST，这个WFST表示的搜索网络的输入是HMM状态。然后这个WFST可以通过各种优化运算来去掉其中的冗余部分而变成等价的但是更加紧凑高效的WFST来加速解码过程。具体包括 ：

* wfst网络确定化，最小化算法；
* 动态语言模型计算算法；
* 快速维特比搜索算法，基于空边和静音特殊处理算法；
* 动态结果回溯和词图构建算法。

基于热词网络的令牌传递算法：

在大规模连续语音识别中，解码过程一般采用加权有限状态转换器wfst，这里的热词解码算法即on-the-fly decoding就是在语音识别的帧同步算法中，其搜索空间包含基础模型的wfst网络和根据热词或领域词构建热词的wfst网络，解码过程中动态的结合利用基础网络和热词网络进行解码，其目的是有效的在一遍解码过程中融入领域热词的信息又不降低其他通用领域的信息权重，实现提高领域热词识别率的方法，领域热词网络权重可以根据任务更新，识别过程不用增加后处理耗时。具体包括热词网络的读取和构建，动态的热词网络解码两个部分。

调用流程如图3.4.4所示：



图3.4.4 解码函数调用流程图

## 3.5 结果后处理模块

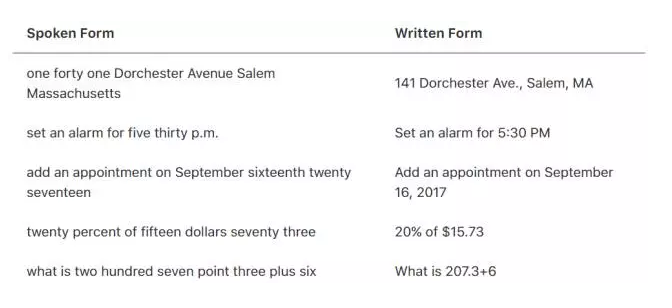
后处理模块是根据解码出来的识别结果，对识别结果进行一系列的处理；如音素转拼音，ITN转换，添加标点，热词替换等处理操作。

### 3.5.1音素转拼音子模块

解码后的识别结果是训练过程的音素单元，利用音素单元与汉语拼音之间的映射，进行相互转换。

### 3.5.2 ITN子模块

逆转文本标准化（ITN, Inverse Text Normalization）：在大多数语音识别系统中，核心语音识别器会生成语音形式的标志序列，这个序列随后通过 ITN 过程被转换成书写形式。ITN 包括数字、日期和地址等对象。图3.5.2.1 展示了输入为语音形式，输出为书写形式的例子。

  
图3.5.2.1

### 3.5.3 标点子模块

利用标点模型，对识别结果进行添加标点操作。标点模块主要采用CRF算法，用带有标点符号的文本进行训练，在模型的基础上再添加语义的方法，实现逗号、句号、感叹号、问号的添加，目前支持这四种中文标点符号。

CRF(Conditional Random Field)：条件随机场，一种机器学习技术（模型）。CRF由John Lafferty最早用于NLP技术领域，其在NLP技术领域中主要用于文本标注，并有多种应用场景，例如：分词（标注字的词位信息，由字构词）、词性标注（标注分词的词性，例如：名词，动词，助词）、命名实体识别（识别人名，地名，机构名，商品名等具有一定内在规律的实体名词）、标点添加等；

### 3.5.4 热词子模块

热词模块是针对一些特定领域的词语识别效果不好，用户可以将该特定领域的词语动态输入到识别引擎，引擎通过热词技术对识别出错的词语进行优化，提高这类词语识别的准确率。 声学所连续语音识别引擎 API 提供两种应用模式：

一种是在线实时录音，检测语音并进行识别。这种方法适合用于需要实时获得语音内容的场合。例如可以在智能导航系统、语音听写系统进行开发使用。

另一种是离线读入录音文件进行识别，可以充分的利用聚类、实时自适应等技术，把语音的内容充分分析，进一步提高识别准确度。

## 3.6多线程封装模块

引擎通过CycLine类来实现多线管理，用户层通过申请会话获取处于空闲状态的会话，申请成功后可以进行语音任务的发送，引擎内部将该路会话的语音任务进行前端处理后得到声学特征并将该声学特征送入对应的任务池LateralLink中等待解码线程获取并解码。

多线程解码流程如下图3.6所示：



图3.6

# 4.可靠性设计

4

识别引擎设计考虑了应用场景的复杂性和输入的多样性的情况，为了让识别引擎应用复杂的应用场景中高性能的工作，识别引擎中的模型采用了较全的多领域数据进行训练，并包含了该语种全集字词作为字典，保证了识别性能的可靠性；

识别引擎支持多种不同格式的语音数据作为输入，同时引擎支持高并发且异步的处理方式，保证了各个处理单元间的独立工作。

识别引擎内部对每一个操作和异常都做了判断以及日志记录，便于记录引擎工作流程和出现异常时的问题排查。并且识别引擎测试按照严格的7\*24小时压力测试标准进行测试，同事也针对所以异常数据进行异常情况测试，保证引擎的稳定性和可靠性。

# 5.扩展性设计

5

识别引擎支持多路并发，保证了不同cpu数和内存大小的机器拓展需求；识别引擎可以通过参数配置来实现不同语种的任务处理以及性能和功能上调节，目前支持汉，英，日，韩，维，粤，哈萨克，越南，藏，土耳其语种的语音数据进行语音识别。

# 6.安全性设计

6

*【详细说明引擎如何满足安全性，包括防止系统错误，如何进行保密，用户数据泄漏等】*

*【说明测试要点】*

# 7.维护性设计

7

识别引擎为了方便后期的引擎维护，识别引擎采用标准c/c++函数同时在较低编译器上进行编译。

识别引擎开放了大量内部参数，保证了引擎的可配性和可维护性。常见参数如下：

|  |  |  |
| --- | --- | --- |
| 参数名 | 取值 | 说明 |
| SampleRate | 8000或16000 | 声学模型训练语音的采样率。目前仅广电模型SampleRate=16000，其余模型下均为8000。 |
| isOnlineMode | true或fasle | 是否走在线语音流处理流程。true:在线；false:离线系统。 |
| NumberOfSession | >=1的整数 | 开启会话数。实际可通过TBNR\_Init()传参设定开启会话数，仅当参数<=0时, NumberOfSession才起作用。实际会话个数最多不能超过授权线数。 |
| NumberOfDecoder | >=1的整数 | 解码器个数。在线模式下，实际解码器个数=实际会话个数；离线模式下，decoderNumPerSession>0时，实际解码器个数= int (0.5 + decoderNumPerSession \* sessionNum)。实际TBNR解码器个数最大不能超过授权文件授权线数。 |
| NumberOfDecoderPerSession | >=1的整数 | 每个会话开启的解码器个数。仅对离线模式适用。 |
| CNFormat | DaLian或KWS | 引擎模式。仅做纯识别还是要做关键词检索。 |
| IndexWithRec | true或  false | 是否建索引。  CNFormat与IndexWithRec一般如下搭配使用：   * CNFormat=DaLian,IndexWithRec=false:纯识别模式 * CNFormat=KWS, IndexWithRec=false:纯关键词检索模式 * CNFormat=KWS, IndexWithRec=true:识别+检索模式 |
| isSaveData | true或  false | 是否保存分段语音。isSaveData=true,保存分段语音到参数“FeatureRootDir”所指目录下（一般位于model/ElvaData/目录下）。 |
| KeepAllSpeech | true或  false | 是否保留无效语音(如，静音等)。  KeepAllSpeech=false,无效语音不送解码。  KeepAllSpeech=true,无效语音送解码。 |
| isDoEPD | true或  false | 是否做分段（端点检测）。 |
| VadFormat | oldEPD或BC | 端点检测算法。BC:广电引擎端点检测；oldEPD:EPD端点检测。 |
| DetectRingNum | >=0的整数 | 彩铃检测开关。DetectRingNum=0时，不做彩铃检测。DetectRingNum>0时，做彩铃检测。 |
| PrintSegTimeOFWord | true或  false | PrintSegTimeOFWord =true: 转写候选结果中词的时间是相对于整条语音的绝对时间；  PrintSegTimeOFWord =false：词的时间是相对于句的时间。 |
| PrintPunctuation | true或false | 是否输出标点。如果PrintPunctuation=true，则必须设置PrintSegTimeOFWord=true。 |
| PrintPinyin | true或false | 识别结果带拼音输出。如果PrintPinyin=true，必须设置字典WordPhonemePinyinDict。且只有当解码器版本支持拼音输出时，最终才能够输出拼音。 |
| DictFileName |  | 识别用字典，含词语及其对应音素发言。该字典跟模型一一对应，更换模型时，注意更换相应字典。 |
| WordPhonemePinyinDict |  | 识别结果输出拼音时所用字典，含词语、对应音素及拼音。 |
| MonoListFileName |  | 音素字典。 |
| SyllableWordDict |  | 检索用字典，含拼音、词语、音素。 |
| bfsmFileName |  | 语言模型。 |
| LatticeDir | LatticeDir=./lat  或  #LatticeDir=./lat | CN结果存放路径。  纯识别时：#LatticeDir=./lat  做检索时：LatticeDir=./lat |
| FeatureRootDir | ../ElvaData | 索引文件存放路径。 |
| StateProbFile |  | 声学模型：.arpa文件。 |
| NNModel |  | 声学模型：.bin文件。 |
| VocabFile | ../syllvocab.list | 拼音字典。关键词检索时需要。 |
| MultiWordFile | ../multiword.vocab | 拼音、音素对应字典。关键词检索时需要。 |
| LogLevel | 0 | log级别值越大输出越详细[0-2] |

# 8.易用性设计

8

识别引擎采用c/c++接口，接口方便灵活，支持c/c++，java，python等变成语言调用，只需掌握基础的语言学习知识，即可自主设计功能模块和功能的实现。