# 离线端到端语音识别引擎设计概要说明

## 1.概述

## 1.1简介

### 1.1.1引擎功能介绍

中科院声学所端到端语音识别解码器可以完成非流式场景下的语音识别任务。引擎实现方案为：对输入的语音识别特征序列（例如mfcc-hires），通过神经网络完成前向计算（例如Transformer作为语音识别模型，CE/CTC Loss作为联合训练准则）并输出字符的后验概率，通过帧同步的方式，以静态加权有限状态转移器的形式，将语音转写为文字。

满足如下功能及性能需求：

1）支持大词表连续语音识别；

2）支持实时语音流识别；

3）不依赖于语种相关的发音字典；

4）具有较低的字错率，容忍一定的口音。

### 1.1.2引擎输入介绍

（1）资源文件，包括：

* 配置文件onepass.cfg
* 端到端语音识别模型
* 静态加权有限状态转移器网络

（2）语音数据

* 语音识别特征（例如mfcc-hires）

### 1.1.3引擎输出介绍

语音识别结果，包括：识别文字和对应的时间点。

## 1.2 读者对象

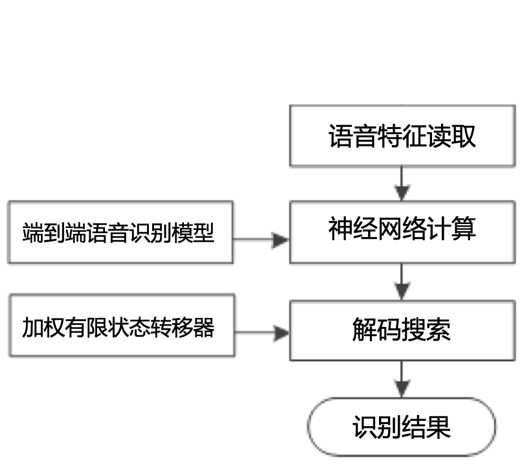
本文档的读者对象为引擎算法研究，开发人员和测试人员。读者通过阅读该文档了解设计的主要细节，可进行具体代码实现和测试。

## 1.3 参考资料

1. Miao, Y., Gowayyed M., Metze F.: ‘EESEN: End-to-End Speech Recognition Using Deep RNN Models and WFST-based Decoding’, ASRU, 2015, p. 167-174.
2. Vaswani, A., Shazeer, N., Parmar, N., *et al*: ‘Attention is All you Need’, *NIPS*, 2017, p. 5998-6008.
3. Watanabe, S., Hori, T., Karita, S., *et al*: ‘ESPnet: End-to-End Speech Processing Toolkit’, *Interspeech*, 2018, p. 2207-2211.
4. Karita, S., Soplin, N. E. Y. and Watanabe, S., *et al*: ‘Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration’, *Interspeech*, 2019, p. 1408-1412.
5. Miao, H., Cheng, G., Gao, C., *et al*: ‘Transformer-Based Online CTC/Attention End-To-End Speech Recognition Architecture’, *ICASSP*, 2020, p. 6084-6088.
6. Miao, H., Cheng, G., Zhang, P., *et al*: ‘Online Hybrid CTC/Attention End- to-End Automatic Speech Recognition Architecture’, *IEEE ACM Trans. Audio Speech Lang. Process.*, 2020, 28, p. 1452-1465.

# 2.总体设计

端到端语音识别引擎的基本框架图如下：



语音数据送入引擎后，经过语音特征读取、神经网络计算和解码搜索3个模块，最终得到语音识别的结果。各模块概述如下：

1. 语音特征读取：输入是语音识别任务中常见的语音特征，例如40维的mfcc-hires特征，80维的filter-bank特征，语音特征将被送入声学概率计算模块。
2. 神经网络计算：在引擎初始化阶段加载端到端语音识别模型，例如Transformer模型，构建端到端语音识别神经网络；将特征送入神经网络中，计算出字符后验概率。
3. 解码搜索：在引擎初始化阶段加载加权有限状态转移器，构建静态网络，负责将字符的后验概率序列转换为字符序列，将后验概率送入静态网络，通过集束搜索输出识别结果。

# 3.可靠性设计

识别解码器内核设计考虑了上层调用的复杂性和输入的多样性的情况，为了支持不同应用场景中高性能的工作，内核中的基础模型采用了多领域较数据量进行训练，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；

识别内核支持多种不同格式的语音特征作为输入（需要相关的声学模型支持），支持高并发同步处理方式，对内部各个模块尽可能减少耦合。

识别解码器内核按照严格的7\*24小时压力测试标准进行测试，同时也针对所有异常数据进行异常情况测试，保证解码器的稳定性和可靠性。

# 4.扩展性设计

识别解码器内核支持面向对象编程，支持多个实例多路并发，保证了不同cpu数和内存大小的机器拓展需求。内核支持对于不同领域采用不同的字典，语言模型以及声学模型的设计，可以根据实际需求选取更匹配的模型达到最优性能。

# 5.维护性设计

## 5.1 主要配置参数说明

### 5.1.1.解码器配置文件参数说明

|  |  |  |  |
| --- | --- | --- | --- |
|  | **参数名** | **说明** | **默认配置** |
| **引擎相关** | EngineType | 语音识别引擎类别 | e2e |
| useWFSTCTC | 是否使用3.2方式解码 | true |
| **模型相关** | ModelFile | 端到端模型路径 | default/modelFile.bin |
| wfst | WFST模型路径 | default/TLG.int |
| key\_wfst | 热词WFST模型路径 | default/TLG\_key.int |
| insym | WFST的输入列表 | default/tokens.txt |
| outsym | WFST的输出列表 | Default/words.txt |
| VocabularyFile | 端到端模型的输出列表 | default/vocabularyFile.txt |
| LanguageModelFile | 语言模型路径 | default/lstmlmFile.bin |
| filter1 | 端到端模型第1层CNN卷积核数 | 320 |
| filter2 | 端到端模型第2层CNN卷积核数 | 320 |
| **解码相关** | factor | 分数缩放因子 | 10 |
| acwt | 声学分数系数 | 1.0 |
| lmwt | 语言学分数系数 | 0.3 |
| mainBeam | 分数剪枝参数 | 20.0 |
| maxHyps | 直方图剪枝参数 | 30 |
| blkScale | <blank>概率的惩罚系数 | 0.4 |
| isword | 输出格式（1为分词结果） | 0 |
| beamSize | 节点剪枝参数 | 10 |
| ctcBeamSize | 节点剪枝参数 | 15 |
| maxDepth | 最大识别字数参数 | 5000 |
| toleration | 终止搜索参数 | 3 |
| dEnd | 终止搜索参数 | -3.0 |
| ctcWeight | CTC分数系数 | 0.5 |
| lmWeight | 语言学分数系数 | 0.0 |
| forceEnd | 强制扩展<eos>参数 | false |
| speechSpeed | 限制识别字数参 | 0.1 |
| timeReduction | 语音特征采样倍数参数 | 4 |
| forceAlign | 强制对齐参数 | true |
| forceAlignBlank | 强制对齐参数 | true |
| keywt | 热词功能权重参数 | 0 |
| Nbest | 多候选结果数目 | 1 |
| silThreshold | 强制对齐平滑参数 | -1e-4 |

1. 上表中标红的参数为经常需要改动的参数，其余参数基本不用改动。

# 6.易用性设计

识别解码器内核采用c/c++接口，接口方便灵活，内部运行高效，支持c/c++调用，只需掌握基础的语言学习知识和基本的语音应用调用流程，即可自主设计功能模块和对新增功能的实现。