ITN引擎源代码概要设计说明

# 概述

## 1.1 简介

### 1.1.1引擎功能介绍

ITN引擎可以实现字符串之间的转译，如数字转阿拉伯数字等。

适用场景及性能：语音识别结果后处理，将数字转换为阿拉伯数字，具体可以根据需求进行定制开发。

### 1.1.2引擎输入介绍

（1）资源文件，包括：

语法规则文件：example.far

（2）识别结果

语音识别的转写结果作为输入

### 1.1.3引擎输出介绍

ITN引擎会对识别结果进行处理，按照语法规则文件对识别结果进行重写。

## 1.2 读者对象

本文档对ITN引擎进行总体设计，包括使用的算法和主要实现代码，主要面向引擎开发和维护人员。

1）有一定的C和C++语言基础；

2）了解相关算法。

## 1.3 术语定义

ITN(Inverse Text Normalization)：逆文本正则化，将汉字的数字转写为阿拉伯数字。

## 1.4 参考资料

Paper：

A Mostly Data-driven Approach to Inverse Text Normalization

The OpenGram Open-source finite-state grammar.pdf

<http://www.openfst.org/twiki/bin/view/GRM/Thrax>

# 2. 总体设计

ITN处理流程如下：



图2.1 ITN处理流程图

ITN引擎主要实现识别结果后处理，包括语法文件的构建和程序设计模块。

## 2.1 语法文件的构建

首先我们人为构建需要的语法文件，将语法文件通过thrax和WFST进行编译，构建语法文件的网络空间，实现识别结果后处理；

## 2.2 程序设计

1） 对识别结果进行处理，去空格

2） 识别结果后处理，对数字进行处理，输出处理后的结果

3） 对输出结果进行处理，还原输入的分词形式，以及时间点的合并标点引擎主要利用条件随机场算法，并对引擎接口进行设计实现，下面进行简要介绍。

# 3. 可靠性设计

1）函数返回值规范，对应程序里的异常（如：文件打开失败、内存分配、释放异常等），通过函数返回值的形式逐层向上反馈，避免直接退出引擎；

2）支持多线程异步调用，加快处理速度。

# 4. 扩展性设计

## 4.1 支持跨平台

目前引擎支持intel、ARM平台上的编译运行。

# 5. 安全性设计

引擎对各类异常进行判断，有一定的容错机制。

# 6. 易用性设计

引擎接口设计简洁，方便开发者快速熟悉引擎接口调用流程，快速完成接口集成；

引擎同时提供了接口测试程序，可供开发者参考。