# NLP第一次作业报告

### 班级：AI2103 学号：U202115092 姓名：郭子毅

## 任务目标

利用language model语言模型和channel model通道模型对testdata.txt的1000行文本数据进行改正，并将改正结果输出到result.txt中，利用eval.py文件对输出结果进行评分，输出准确率。本项目在利用eval评分的同时将错误文本改正输出到data\_ans\_result.txt中，方便对程序进行改进。

## 数据准备及代码架构

测试数据存储在textdata.txt中，程序输出的更正文本存储在result.txt,正确答案存储在ans.txt,本次程序文本的词典存储在vocab.txt,spell-singal-error.txt记录了正确单词错写为错误单词的类型和次数，channelprob.json记录了channel model计算通道模型的数据缓存文件，主要是每个单词对应各种类型错误（deletion/insertion/substitution/transposition)的概率。Languagemodel.json记录了language model基于bigram语言模型计算前后各一个词确定的情况，计算中间是某个词的概率。channel model和language model的源代码都在model.py中。main.py为主程序，执行全部功能。nltk文件夹为路透社新闻语料库训练数据，该训练数据用于语言模型统计概率。

## 算法原理

更正非词错误： 对于每个文本句子检测是否有非词错误，即检测句子的每个元素是否在vocab.txt中，如果有至少一个元素不在vocab中，即为非词错误。利用正则化表达式将句子的每个元素分开，包括所有的标点符号成为单个元素（如’s将被分为‘’’和‘s’）。对于不在字典之中的元素，利用编辑距离函数generate\_candinates（编辑距离为1）和函数generate\_edit\_two（编辑距离为2）生成候选词序列，编辑距离函数对输入元素执行插入、删除、替换单个大小写字母。通过通道模型和语言模型计算每个候选词的备选概率，选择概率最高的候选词改正。



P（x|w）是channel模型输出的概率，代表将正确词w错写为错误词x的概率，利用spell-singal-error.txt统计错写类型的次数，利用model.py中函数edit\_distance计算单词转换错误类型，搜索对应错写类型的次数，然后利用vocab.txt统计正确词改写类型（正确词中的对应字符串）总次数，两者相除输出即为P（x|w），P（w）是language 模型输出的概率，代表w词出现在当前位置的概率，在bigram语言模型中，利用w的前后各一个词来统计出现次数来计算w出现概率P（w）。利用log取值，每个候选词概率可表示为logP（x|w）+logP（w），生成候选词概率序列，选取最大候选概率词，改正句子并将其输出到result.txt文件中。

更正真词错误：对比非词错误改正，利用函数chosen判定句子改正前各元素是否均出现在字典中，如果均为字典中元素，则该句子被判定为真词错误，后续对句子的每个元素进行改正，生成候选词列表，按照非词错误相同步骤计算概率，选取最大概率候选词改正句子。

### 准确率报告及当前缺点

准确率报告：94.6



缺点：如果一个句子中同时出现了非词错误和真词错误，则只能改正非词错误，无法改正真词错误。在data-ans-result.txt文件中包含了未改正的句子序列。

改进方法：在更正完所有错误后，将更正完的文本所有元素再一次按照真词错误进行改错，但是此方法可能会造成准确率下降和程序运行复杂度大大增加。