拼音输入法

算法基本思路与实现

本次大作业一共实现了两种模型，分别是基于字的二元和三元模型，两种模型的基本思路都是使用贝叶斯公式计算概率，最后通过动态规划求得最优解，其具体分析如下

1.基于字的二元模型

思路

a）二元模型使用如下所示的贝叶斯公式，其中 $P(w_i|w_{i-1})$ 是字 w_{i-1} 后面接字 w_i 的概率，而 $Q(w_i)$ 则是指和 w_i 同拼音中的字 w_i 出现的概率

$$\prod_{i=1}^{n} P(w_i|w_{i-1}) Q(w_i)$$

b）所求的最优解即为使得上式取最大的句子，通过动态规划即可求得。令 F_{ij} 表示第 i 个拼音取第 j 个汉字的概率，则状态转移方程为：

$F_{i,j} = \max(F_{i-1,k} * P(w_j|w_k) * P(w_k))$

实现过程

a）使用所给定的拼音汉字对照表，生成拼音字典 mydic.json，主要记录每个汉字及对应的读音，为之后统计频数提供条件

b）统计每个拼音对应的汉字出现的频数。使用语料库，针对每行数据，过滤掉特殊字符后，通过拼音字典 mydic 获取到其对应的拼音，之后统计频数，存放到文件 final_pinyin_to_word.json 中。

c）统计每个字后面出现的下一个字的频数和概率，生成文件 final_word_to_char.json。依然是使用语料库，这次不过滤特殊字符，当前后两个连续的字符都是汉字的时
候，就认为它们是相连的汉字，计入字典中。
之后即可使用动态规划编写程序，详见 c2.py

2.基于字的三元模型

思路

a） 三元模型使用如下所示的改进公式，与二元类似，其中 $Q(w_{i-1}*w_{i-2})$ 是和 $w_{i-1}*w_{i-2}$ 同拼音的词中 $w_{i-1}*w_{i-2}$ 出现的概率，$P(w_i|w_{i-1}*w_{i-2})$ 是出现 w_{i-1} 和 w_{i-2} 后面接 w_i 的概率。

$$\prod_{i=1}^{n} P(w_i|w_{i-1} * w_{i-2}) Q(w_i * w_{i-2})$$

b）状态转移方程也与二元模型类似，之后通过动态规划即可求出最优解，在此不再赘述。

实现过程

a）统计两个拼音对应的汉字出现的频数，同二元模型类似，统计结果保存在 final_pinyin2_to_word2.json 文件中

b）统计所有两个字后面出现的下一个字及其频数，依旧使用语料库，当连续的三个字符都是汉字时，就认为是相连的三元组而计入到统计结果中，最后生成文件 final_word2_to_char.json

之后即可编写程序，详见 c3.py

3.语料库

大部分来源于作业中提供的 sina 新闻语料；除此之外，自己另找了 wiki 百科的一些词条数据

实验效果

1.二元模型

效果较好

```
please input the pinyin: jue sheng quan mian jian cheng xiao kang she hui
决胜全面建成小康社会
please input the pinyin: shen du xue xi tui dong le ren gong zhi neng de fa zhan
深度学习推动了人工智能的发展
please input the pinyin: quan guo ren min dai biao da hui
全国人民代表大会
```

效果一般

```
please input the pinyin: qing bu yao shu ru tai duo qi guai de yu ju
情不要输入太多奇怪的育局
please input the pinyin: jin yong de wu xia xiao shuo hen hao kan
仅用的武侠小说很好看
```

说明

由于所用语料库为主要为新闻数据和 wiki 词条，因此对于比较正式的句子，程序的效果较好，而像“金庸“小说，则会翻译成更常见的“仅用”。

同时，对于处于句尾的拼音，因为后面没有了通过动态规划再来修正的可能，所以当正确的词语在语料中出现的概率并不高时，就容易翻译错误，如上图中的“育局”应为“语句”

2.三元模型

效果较好

```
please input the pinyin: zhong guo guo jia dui zhan sheng han guo guo jia dui
中国国家队战胜韩国国家队
please input the pinyin: zhi neng ji shu yu xi tong guo jia zhong dian shi yan shi
智能技术与系统国家重点实验室
please input the pinyin: te lang pu xi wang bu jiu he zhong guo guo jia zhu xi mian dui mian hui wu
特朗普希望不久和中国国家主席面对面会晤
```

效果一般

```
please input the pinyin: ni de shi jie hui bian de geng jing cai
拟的世界会变得更精彩
please input the pinyin: zhong guo shi ren min dang jia zuo zhu de she hui zhu yi guo jia
中国市人民当家作主的社会主义国家
please input the pinyin: shu xue fen xi tai nan le
数学分析台难了
```

相比于二元模型，三元模型的正确率确实有所提升，一个比较明显的例子就是“金庸的武侠小说很好看”，在三元模型中翻译完全正确，而在二元模型中，最后的翻译结果为“仅用的武侠小说很好看”

性能表现

		测试集 1	测试集 2
二元	逐字	89.0%	76.7%
	整句	39.3%	26.6%
三元	逐字	91.3%	79.9%
	整句	44.1%	26.6%

其中，测试集 1 为两篇新闻稿，测试规模不大，为一百行左右，且由于语料较符合训练数据，因此正确率较高。测试集 2 为老师在微信群中所发的测试数据

平滑参数的选择

尝试了 λ 平滑方法，但是效果不好，正确率反而有所降低。于是就直接简单粗暴遇到不存在的字使用 1 代替其频数。

总结

这次实验是对第一章节搜索内容的一次实践，人工智能的大多数问题，个人感觉来讲，都像是一种优化问题，并且在大多时候，都能被转化为求解最短/优路径。拼音输入即是如此，第一章介绍的各种算法就是关于此类问题求解方法。

总的来说，我觉得对于此次作业来讲，提早规划比较重要。从一开始就要想清楚实现模型需要统计哪些数据，以及该以怎样的格式储存。当然，其实只要搞懂了隐马尔科夫过程，这些问题的答案都跃然纸上。不过为了完成此次作业，也需要查阅许多资料，而我也因此收获了许多。从贝叶斯置信网络，再到隐马尔科夫过程，我对人工智能与机器学习等有了新的认识

至于改进的方向，我觉得有以下几点：(1) 对出现在结尾的字处理的还不够完善。很多时候一整句话可能就错在结尾的几个字上，这部分可以完善；(2) 语料库过于单一。虽然之前添加了一些其它语料，但仍效果有限，需要继续添加，扩展语料库。这里所说的扩展语料库，一方面是指数量上的扩大，越大的语料就越能放映客观的规律；另一方面，则是指对于各种各样的语料都能囊括，只有新闻或者 wiki 词条的语料库谈不上是完全的。