# 大作业-拼音输入法

计 63 黄冰鉴 2016011296

2018/4/21

## 算法基本思路和实现过程：

### 1. 基于字的二元模型

基本思路：$\prod_{i=1}^{n} P(w_i \mid w_{i-1}) CF(w_i)$

（1） 二元模型使用贝叶斯公式，其中 CF(wi)是和 wi 同拼音中 wi 出现的概率，P(wi|wi-1)是出现 wi-1 后面接 wi 的概率。

（2） 通过动态规划就可以得到最优解。表示第 i 个拼音取第 j 个读音的概率。状态转移方程： $F_{ij} = max(F_{i-1,k} * P(w_j|w_k) * P(w_k))$

实现过程：

（1） 生成字和对应的拼音的文件 char_to_pinyin.json。因为 python 的 utf-8 编码中汉字的范围为 u'\u4e00'到 u'\u9fa5'，调用 python 的开源库 pypinyin 生成字对应的拼音，然后存到文件中。

格式：{'汉字'：对应拼音的 list}

{"一": ["yi"], "丁": ["ding", "zheng"], "丐": ["kao", "qiao", "yu"], "七": ["qi"],

（2） 统计每一个拼音出现的汉字对应的频数和概率，生成 pinyin_to_char3.json。使用语料库，将每行的文字过滤掉特殊符号后放入 Pypinyin 中得到对应的拼音 list，然后统计频数，存到文件中。

格式：{'拼音'：{'总频数'：cnt，汉字：list，频数：list}

"nu": {"cnt": 120524, "chs": ["仅", "傲", "傩", "努", "呶", "奴", "弩", "伎", "迈", "陋"], "cnts": [143071, 0, 9017, 11, 0, 0, 0, 0, 0, 0]},

（3） 统计每个字后面出现的下一个字的频数和概率，生成文件 char_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。

格式：{'汉字'：{'cnt'：总频数，'char'：{'汉字'：频数}}}

{"中": {"cnt": 4198123, "char": {"国": 1139052, "央": 257384,

（4） 在有了以上三个文件之后，就可以开始编写动态规划程序 dp.py。

（5） 小插曲：在实现过程中曾经被同学告知过另一种贝叶斯中的 P(wi|wi-1)的定义，即在前一个字是 wi-1 且下一个字的读音是 wi 的这些字中，wi 出现的频数。但是在实现过程中发现这种定义理论上不够合理，并且实测准确率确实降低了，所以还是使用原来的定义方法。

### 2. 基于字的三元模型

基本思路：$\prod_{i=1}^{n} P(w_i|w_{i-1} * w_{i-2}) * P(w_{i-1} * w_{i-2})$

（1） 三元模型使用改进的贝叶斯公式，其中 P(wi-1*wi-2)是和 wi-1*wi-2 同拼音的词中 wi-1*wi-2 出现的概率，P(wi|wi-1*wi-2)是出现 wi-1 和 wi-2 后面接 wi 的概率。

（2） 通过动态规划就可以得到最优解。表示第 i 个拼音取第 j 个读音的概率。状态转移方程： $F_{ij} = max(F_{i-1,k} * P(w_j|w_k) * P(w_k))$ 这里的 wk 表示前两个字取第 k 种取法。

实现过程：

（1） 统计两个拼音出现的两个汉字对应的频数和概率，生成 py2_to_ch2.json。因为调用 pypinyin 生成对应拼音的速度非常慢，这里可以使用之前二元模型中的 char_to_char.json 来生成。

格式：{'拼音 2'：{'总频数'：cnt，汉字 2：list，频数：list}

{"zhong guo": {"cnt": 1146536, "chs": ["中国", "中过", "中果", "中裹",

（2） 统计每两个字后面出现的下一个字的频数和概率，生成文件 ch2_to_ch.json。依然是使用语料库，当连续三个字符都是汉字的时候，就认为它们是相连的三元组，计入字典中。

格式：{'汉字 2'：{'cnt'：总频数，'char'：{'汉字'：频数}}}

{"中国": {"cnt": 955677, "char": {"天": 4779, "气": 2406, "国": 21237, "驻": 8714,

（3） 在有了以上两个文件之后，就可以开始编写动态规划程序 dp.py。

3. **语料库**

大部分来源于老师给的 sina 新闻，另外自己附加了一部分，有古诗，新闻，小说，微博等。

## 实验效果：

### 1. 二元模型：

效果较好的例子

```
PS D:\2018spring\IntroAI\pysrf> python .\dpdirect.py
input something: duo qu xin shi dai zhong guo te se she hui zhu yi wei da sheng li
夺取新时代中国特色社会主义伟大胜利
input something: jue sheng quan mian jian cheng xiao kang she hui
决胜全面建成小康社会
input something: zi zhu chuang xin tui jin wang luo qiang guo jian she
自主创新推进网络强国建设
input something: xi jin ping zai quan guo wang luo an quan he xin xi hua gong zuo hui yi shang qiang diao
习近平在全国网络安全和信息化工作会议上强调
```

从这个例子可以看出，因为主要语料库还是 sina 新闻，所以对正式的句子效果较好。

效果较差的例子

```
5363 7169 0.748082
者要是农不死你我也是美者了
5372 7182 0.747981
大家能不能不要输入这么多奇怪的育局
```

第一句是“这要是弄不死你我也是没辙了”，非常口语化的句子，可能因为小说在语料中占比太小，所以无法正确翻译。

第二句的最后两个字是“语句”，这个词看起来很正常，但是实际上在语料中出现的概率都不高，只有 828/160704 = 0.5%，而且因为是在结尾，后面没有通过动态规划修正的可能。

### 2. 三元模型：

效果较好的例子

```
今天晚上有好看的电影
562 712 0.789326
北京奥运会开幕式非常精彩
574 724 0.792818
全国人民代表大会在北京人民大会堂隆重召开
594 744 0.798387
金庸的武侠小说非常精彩
605 755 0.801325
你的世界会变得更精彩
615 765 0.803922
深度学习技术推动了人工智能的发展
631 781 0.807939
```

三元模型可以更好地翻译出一些较为口语化的句子。因为二元前缀的可能种类大大增加，一个频数不多的二元组可能在它的拼音组中占了较大比例，因此可能会被输出出来。

效果较差的例子：

```
沸腾的条件是大道费店兵持续吸热
5208 6702 0.777081
我艾学习学习时我快乐
5216 6712 0.777116
人间四月芳菲尽
5223 6719 0.777348
电量数字人生
5227 6725 0.777249
```

第一句中间是“达到沸点”，在语料库范围之外，所以要翻错一般是直接连续翻错很多个字……第二句和第四句虽然翻错了，但是从语义上是可以接受的，“学习时我快乐”和“学习使我快乐”都有道理。

## 对比参数选择和性能分析：

1. **平滑参数：**

尝试了两种平滑方法，一是λ平滑，但是效果不好，正确率降低了非常多，没有找到一个比较合适的λ；而是如果不存在的话，直接用 1 代替频数，效果反而还可以。

**2. 性能分析：**

我使用了三个不同的测试集。

（1） 二元模型字正确率：对句子比较混乱的测试集能达到 74.8%，对句子较为规整正式的测试集能达到 78.0%。

（2） 三元模型字正确率：对句子比较混乱的测试集能达到 77.9%，对句子较为规整正式的测试集能达到 81.7%。

（3） 三元模型句正确率：对句子比较混乱的测试集能达到 32.0%，对句子较为规整正式的测试集能达到 54.4%。

## 总结和改进：

总的来说，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，以怎样的格式储存。而考虑这些问题其实最好倒过来想，从动态规划需要哪些数据和怎样的格式来考虑，而不是统计完之后再考虑动态规划怎么调用已经固定格式的数据。这样可以更好地规划好整个设计，而不是在中途遇到问题之后推倒重来。

在实现模型过程之后不能好高骛远，操之过急。我是先从二元写起，在实现并完善了二元模型之后，发现二元准确率的极限就是 75%左右，不能再上升了，这才考虑了使用三元模型；而在实现三元模型的过程中，通过复用二元模型的统计数据和代码，只用了几个小时就完成了二元到三元的转变，比直接三元快了不少。在之前没有写过输入法的情况下，这样做性价比最高，收获也最大。

至于改进的话，大概可以分为四点：(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聊，但是因为比例太小，效果一般，需要继续添加语料库，使概率更加准确；(3) 平滑方法有待改进。目前只使用了常数和线性的平滑方法，可以尝试其它平滑方法看是否有促进效果；(4) 三元模型中贝叶斯公式还有另外一种形式，可以一试。

黄冰鉴
2018/4/22