拼音输入法实验报告

计 53
张正彦
2014010515

1 算法思路

拼音输入法问题定义 求使得 $P(S) = \prod_{i=1}^{n} P(w_i|w_{i-1})$ 最大的句子。

字的二元模型 在字的二元模型中，我们通过语料去学习 $P(w_i|w_{i-1})$，这里 w_i 代表着一个汉字，$i = 1, 2, ..., n$，n 的大小为一二级汉字表的大小。但是我们输入的是一串拼音，对于长度为 L 的拼音串，可以表示为 $V_1, V_2, ..., V_L$，其中 V_i 代表一个音节，在确定汉字表的情况下，我们的拼音表的大小 m 也是确定的。对于每个字来说，它只会对应一个或者多个拼音。通过这样的分析，我们便可以把字的二元模型转换为一个一阶隐马尔科夫模型。

一阶隐马尔科夫模型 在一阶隐马尔科夫模型中，用 a_{ij} 来表示由状态 i 转移到状态 j 的概率，也就是我们的 $P(w_i|w_{i-1})$。与此同时，用 b_{jk} 来表示在状态 j 时表现为可使特征 k 的概率，但通过语料库我们是无法得到字与拼音之间的概率关系的。所以在这里为了简化问题，我假设只要拼音表中没有这种对应则 b_{jk} 就为 0，有对应则为 1，暂时不做对于多音字的处理。

求解最大概率的隐状态 如果暴力枚举，再选取最大，这里的时间复杂度为 O($n^T T$) 这里 n 是汉字的个数，T 为序列的长度，这个时间复杂度是不可接受的。所以这里我使用了 viterbi 算法，它巧妙的利用了一阶马尔可夫模型下一个状态只和当前状态有关的特性，将时间复杂度降到了 O($n^2 T$)。算法的基本思路就是从 T=1 时开始计算若以隐状态 w_i 结尾时，生成 $V_1, V_2, ..., V_j$ 的最大概率，这里 j 为当前计算序列的长度。这样算 $V_1, V_2, ..., V_j$ 概率只要知道 $V_1, V_2, ..., V_{j-1}$ 概率就行了。

实现过程

- 根据词表对汉字进行编号，并且与汉字的读音文件结合，构建一个新的 lookup 文件，便于之后的使用
- 对于文本中的二元关系进行计数，保存为 $n * n$ 的矩阵形式
- 读取统计的数据后，使用 laplace 平滑，避免出现一些次数为 0 的二元组影响结果
- 在算最大关系序列时使用了 viterbi 算法，较快完成了计算并进行预测

2 实验效果

正确长句

- 系统对于人们来说并不是一个陌生的名词
- 全国人民代表大会在北京人民大会堂隆重召开
- 深度学习技术推动了人工智能的发展
- 特朗普希望不久和中国国家主席面对面会晤
- 为了方便大家测试自己的拼音输入法的性能

这些是我从输出结果中挑选的长句，可以看出在一些长句上，输入法也能有比较好的表现。但也能注意到，这些长句中的主要组成部分是长的固定词汇。

错误案例

- 恰似那朵莲花不胜凉风的娇羞
- （错）卡斯那朵莲花不乘凉风的脚臭
- 智能技术与系统国家重点实验室
- （错）智能技术语系统国家重点实验室
- 数学分析本身就是难学的
- （错）术学分析本身就是难学的
- 她是我的母亲
- （错）他是我的母亲

第一个错误例子很明显地体现了多音字对于程序输出的结果，事实上一些读音的出现概率是很低的，但是我们在学习的时候是无法区分的。第二句展现了二元语法的局限性，术与还是术语，对于二元语法而言，程序必然更加倾向于后者。第三句在一定程度上体现对于开头汉字处理的难度，因为事实上第一个字按照纯随机变量的概率选取是不合适的，因为第一个字并不是独立的，与后面的内容也存在着极大的联系。另外，还存在着如第四种句子这样的错误，因为人的想法也不一定就是那个概率最高的组合，这种问题就算是非常成熟的输入法也无法解决，这类型的差错几乎可以说是无法消除。

3 对比参数选择

如果使用 laplace 平滑的话，就不存在参数选择的问题，只是简单地给每一个项加一。在这里，我在 laplace 平滑的基础上，加入 λ 的参数，有公式 $P = (1-\lambda)P(w_i|w_{i-1}) + \lambda P(w_i)$. 下面就是参数与实验结果的表格。

λ	正确字数	字正确率	正确句数	句正确率
0	4646	0.750808015514	173	0.260542168675
0.01	4650	0.751454427925	173	0.260542168675
0.02	4646	0.750808015514	172	0.259036144578
0.05	4645	0.750646412411	175	0.263554216867
0.1	4641	0.75	173	0.260542168675
0.2	4608	0.744667097608	166	0.25
0.5	4496	0.726567550097	139	0.209337349398

可以看出加入 λ 之后，效果的提升不是很大，并且在参数变大的过程中，效果开始下降。

4 改进尝试

我认为预料对于实验结果有比较的影响，也就是说，如果训练语料与测试句子的风格较为一致的话，我们就能获得更好的效果。因此我编写了一个爬虫程序，对于知乎上的部分回答内容进行的抓取，加入语料库进行训练。最后训练的结果获得了显著的提升，它很好地解释了我的假设。具体的结果如下：

正确字数	字正确率	正确句数	句正确率
4872	0.787330316742	215	0.323795180723

对于新增正确例子进行分析：

- 金庸的武侠小说非常精彩
- 请大家选择你觉得可以的时间
- 就是用鸭嘴笔把这个水弄到里面
- 以妖怪之死来说太长了
- 但愿死者耐心包容我逐渐衰退的记忆
- 吹啊吹啊我的骄傲放纵

很让人惊讶，对于一些流行的句子和晦涩的句子，程序也能够很好地完成任务。

5 总结收获

总结 这次实验是对第一章节搜索内容的一次实践，对于人工智能的大多数问题，都是一个优化问题。并且在大多时候，这个优化问题都能被转化为一个求解最短路径的问题，第一章介绍的各种算法就是对于这种问题求解的方法。对于拼音输入法来说，问题的本质也是一样。为了能比较好的解决这个问题，我查阅了许多相关的资料，也收获了许多。从贝叶斯置信网络，再到它的特立隐马尔科夫模型，我对于这种统计机器学习有了新的认识。最后在求解问题时，我使用了 viterbi 算法进行求解，里面运用了动态规划的思想，很好地实践了理论，并且也取得了不错的效果。

改进方向 我认为多音字在本次实验中对于结果产生了极大的影响，所以首先我认为我们对于预料也需要有一些标注，借助预料中的拼音，能够构建一个更好的 HMM 模型。其次，语料库也应该扩展到足够大，这里的大一方面是指数量上的大，越大的语料就越能放映客观的规律；另一方面，大应该是对于各种各样的语料都能囊括，只有新闻或者只有知乎的回答都是不完全的。最后就是可以应用多元的模型来进行计算，并且可以尝试正着句子的顺序构建模型进行计算，然后再反着构建模型进行计算，综合两次的计算结果进行取舍，这样可能可以解决开头首字出错概率高的问题。