拼音输入法实验报告

高敬越 计 92 2019011230

April 11, 2021

目录

1 算法的思路与实现

1.1 基本框架

1. 选用了字的三元模型，并与字的二元模型结合以提高算法的效率；
2. 采用面向对象的设计方式，建立拼音输入法类 Pinyin，并实现统计、加载、预测的方法；
3. 统计词语出现的频率，用频率估计概率，并储存在文件中作为训练的结果；
4. 计算出 $S = \arg\max_S P(S)$；在计算概率时采用了动态规划的算法。
5. 输出文件采用了 gbk 编码。

1.2 具体实现

1. 最初我在本次实验中采用了字的二元模型，但准确率遇到瓶颈。后来增加了三元模型的信息，通过统计的三元模型来增加上文的信息，以提高准确率。
2. 在训练时，首先统计实验考虑范围内的拼音、汉字，并建立它们之间的映射关系；再对所有的语料信息进行遍历统计，在提供的语料库中，分别统计每个字和长度为 2、3 的词语出现的频率，并将统计结果储存在 train 文件夹的数据中。

3. 在统计时，在句首、句尾增加特殊字符 $，并将各类不在汉字集内的符号、生僻字、数字等也变换为 $，并将 $ 当作一个正常的字符进行统计。这样在后来的预测时，也将第一个拼音设置为 $，从此开始计算概率。这样处理的好处有：
   (a) 首先，在处理时，统计到了不在汉字集合里的符号和汉字，防止本应分开的汉字由于删除了非法字符而连接在一起的情况，保证了词语统计的正确性；
   (b) 同时，在计算概率时从 $ 作为词首开始计算概率。由于拼音输入是计算整句或词语的概率，因此要考虑到一个字作为整个词语或整个句子开头的概率，而非只考虑到一个字出现的多少，这样考虑更全面，尤其对一些特有名词概率的计算更准确；
   (c) 最后，在计算证据的概率时，计算使得句尾的 $ 概率最大的句子。这样也考虑到拼音作为完整句子的概率, 计算整个句子、词语的概率更准确。
4. 在计算概率时，采用动态规划的算法，用 $f[i][cur][last]$ 代表第 i 位是字 cur，上一位是 last 的概率。为了防止最后计算的概率过小，在程序中用概率的负对数进行计算。此时概率最大对应负对数最小。
   (a) 首先用二元模型计算概率。在二元模型下，$f[i][cur][last] = \max\{f[i-1][last][k] \times p(cur|last)\}$。用统计得到的频率代替概率，可得：
   $$p(cur|last) = \frac{\#(cur + last)}{\#(last)}$$
   为了防止出现 0 的情况，采用光滑的方法，记
   $$p(cur|last) = reg \cdot \frac{\#(cur + last)}{\#(last)} + (1 - reg) \cdot \frac{\#(cur)}{total}$$
   以防止出现 0 的情况。特别的对于句首，采用另一个参数 $self\_reg$。
   (b) 计算出根据二元模型的概率后，选取出概率在前 num_select 的 last，利用三元模型进行更精准的估算。计算式
   $$f[i][cur][last] = \max(f[i-1][last][k] \times p(cur|last, k))$$
   其中为防止出现 0，取频率代替概率
   $$p(cur|last, k) = reg \cdot \frac{\#(cur + last + k)}{\#(last + k)} + (1 - reg) \cdot \frac{\#(cur)}{total}$$
   根据更新的概率，选取前 num_choose 个并储存用于下一个字计算。
   (c) 在计算到句尾 $ 后，取概率最大并反向即可求得概率最大的句子。

2 实现效果

从老师在群里发的野生测试集和自己想的一些句子中挑选了典型的例句：

2.1 效果好的句子

1. tai wan shi zhong guo bu ke fen ge de yi bu fen
   台湾是中国不可分割的一部分
2. ji qi xue xi ji qi ying yong
   机器学习及其应用
3. da jia neng bu neng bu yao shu ru zhe me qi guai de ju zi
   大家能不能不要输入这么奇怪的句子
4. gei a yi lai yi bei ka bu qi nuo
   给阿姨来一杯卡布奇诺

2.2 效果不够好的句子

1. te lang pu ti ming bu lu ye te chu ren mei guo neng yuan bu zhang
   特朗普提名布鲁野特出任美国能源部长
2. yong nv zi pang de ta lai ce shi yi xia
   用女子旁的他来测试一下
3. qu xiao huo ting zheng de she ji ge ren deng shi xiang de xing zheng shi ye xing shou fei bao kuo
   取消和停征的涉及个人等事项的行政事业性收费包括
4. zhong guo shi ren min min zhu zhuan zheng de she hui zhu yi guo jia
   中国诗人民民主专政的社会主义国家

2.3 策略的效果

1. 优点：在效果好的句子中，第一个句子是与新闻相关的句子，二元模型就可以正确的预测。后三个例句在二元模型中无法成功预测：第二个例句中的及其在二元模型中识别为机器，第三个例句中的句子在二元中会识别为车 (ju) 子。这样是由于二元模型中每个字只与前一个字有关。在增加了三元模型后，在推断时可以获取上下文信息，就可以推断出在句子中间出现及其的概率高于机器的概率。同样在二元中，由于“车”是多音字且出现的多，因此会替代“句”；在增加了上下文信息后就能正确推断。
2. 不足：尽管考虑到上下文的情况，但是还不能真正理解上下文的语义，只考虑到同时出现的概率。因此对于多音字、同音字仍然不能很好的区分。同时对于出现次数较少的人名、专有名词等预测能力同样不足。

3 参数选择

以在微信群中发布的数据集作为参数测试的例子。

3.1 reg 和 self_reg 的选择

在 num_choose 是 6 的情况下，对 reg 在 0.99，0.999，0.9999，0.99999 进行测试

	0.99	0.999	0.9999	0.99999
正确率（字）	90.5%	90.4%	90.4%	90.2%

表 1: 不同平滑参数 reg 时的正确率

考虑到此测试集的语料和新闻语料有所不同，因此采用 0.9999 作为 reg 的值
在 num_choose 为 6，reg 是 0.9999 时，对 self_reg 在 0.2，0.25，0.3 进行测试

	0.2	0.25	0.3
正确率（字）	90.39%	90.39%	90.40%

表 2: 不同初始平滑参数 init_reg 时的正确率

综合表现，选择 init_reg=0.3。

3.2 num_choose 的选择

	10	7	6	5	1
正确率（字）	90.6%	90.5%	90.4%	90.3%	88.4%
耗时	29.9s	19.0s	15.9s	13.2s	5.77s

表 3: 不同 num_chosse 时逐字正确率和运行时间

折中选择 6 作为最终 num_choose 的值。

4 总结和改进

1. 在实现了拼音输入法后，我对概率模型、马尔可夫过程有了更深入的了解；
2. 在这个拼音输入法中还有许多可以继续改进的地方：
   (a) 对于多音字来说，还没有办法对他们的读音进行区分，未对它们不同的读音单独进行统计频率，这样就会使得多音字出现的概率总会大于只有一个读音的字，对结果产生影响。下一步将考虑如何加入对多音字的分析和处理。
   (b) 语料都来自于助教提供的新闻稿，类型较为单一。可以下一步增加增多、各种类型的语言。
   (c) 对每个字可以在语音的基础上增加在语句中的词性，以防止名词做谓语这类错误的产生。
   (d) 可以在字的基础上考虑到词的模型，可能需要进行分词再统计的操作。

5 附: 运行方式

1. 运行需要的 python 文件位于 bin 文件中，包含 main.py 和 train_3.py。在运行时需要在 bin 文件夹中直接运行 main.py，否则会找不到预先保存的统计值。
2. 统计值位于 train 文件夹中；src 文件夹中除 bin 的文件外还包括 train_2.py，这是纯二元模型的源代码。
3. 在运行时，需要输入文件的编码格式和 news 文件一样都是 gbk 格式；输出文件的编码格式也是 gbk。运行时可能比较慢，但经试验有很大一部分时间是在加载预先存储的数据。