拼音输入法作业报告

计 81 包涵 2018011289

2020 年 4 月 12 日

1 算法简介

实现了字的二元、三元和四元模型，以及词的二元模型。

1.1 n 元字模型

n 元字模型是基于 n-1 阶马尔科夫模型的统计模型，具体实现如下

- 统计: 通过语料学习 $P(w_i|w_{i-1}w_{i-2}...w_{i-n+1})$ 的概率，其中 w_i 是注音表中的单个汉字。
- 平滑: 对于高阶的字模型，由于字的 n 元组在语料中出现比例低，需要进行平滑处理，避免概率为 0 的情况。我使用了 absolute discounting 的方法，将样本中每个 n 元组的出现频次减去 0.75 作为对总体频次的估计，把剩余的概率平均分给未出现的 n 元组，公式如下：

$$P(w_i|w_{i-1}w_{i-2}...w_{i-n+1}) = \frac{c(w_iw_{i-1}...w_{i-n+1}) - 0.75}{c(w_{i-1}...w_{i-n+1})} + \lambda * P(w_{i-1}|w_{i-1}w_{i-3}...w_{i-n+2}) \quad (1)$$

其中，

$$\lambda = \frac{0.75 * N(w_{i-1}w_{i-2}...w_{i-n+1})}{c(w_{i-1}...w_{i-n+1})}$$

$N(w_{i-1}w_{i-2}...w_{i-n+1})$ = 对于给定的 $w_{i-1}w_{i-2}...w_{i-n+1}$，不同的 w_i 的总数

如果 $w_{i-1}w_{i-2}...w_{i-n+1}$ 没有在语料中出现，那么 $c(w_{i-1}...w_{i-n+1})$ 和 $N(w_{i-1}w_{i-2}...w_{i-n+1})$ 都为 0，概率退化为

$$P(w_i|w_{i-1}w_{i-2}...w_{i-n+1}) = P(w_i|w_{i-1}w_{i-2}...w_{i-n+2}) \tag{2}$$

- 求解：使用 viterbi 算法，计算能使 $P(w_iw_{i-1}w_{i-2}...w_{i-n+1})$ 最大的汉字序列。
- 多音字的问题：区分多音字时，对语料先用 pypinyin 进行注音，再用带注音的字作为基本单元进行统计；不区分多音字时直接使用单字作为基本单元统计。但是由于 pypinyin 的注音和我们的使用习惯有所区别，对准确率的提升不明显。

1.2 2 元词模型

- 统计：利用 jieba 对语料进行分词，之后学习 $P(w_i|w_{i-1})$ 的概率，这里 w_i 代表单个词语，并建立拼音串到词语的字典。
- 拼音划分：对于输入的拼音串 $p_1p_2...p_m$，将其分段为 $p_1...p_{k_1}, p_{k_1+1}...p_{k_2}, ..., p_{k_{n-1}+1}...p_m$，每一段作为一个词语对待。由于不知道高效的划分方法，这里使用暴力枚举，考虑词语长度不超过 MAX_PHRASE_LEN 的所有划分方法，复杂度接近 $O(2^n)$。因此即便在翻译时使用 4 个进程，对于长度为 15 的串进行翻译也需要近 3 分钟。MAX_PHRASE_LEN 和最大进程数 MAX_PROCESS_NUM 可以在 src/config.py 中进行设置。
- 平滑：对于词模型，概率矩阵更加稀疏，但是统计结果的置信程度更高，所以仍然采用了上一部分字模型的平滑方法，对于退化情况的处理也完全一样。
- 求解：由于词模型的字典过于稀疏，大多数拼音段都找不到对应的词语，所以要结合字模型进行计算。对于已经翻译好的词语 w_{i-1} 和其之后紧邻的拼音段 $p_{k_i+1}...pk_{i+1}$，如果在拼音到词语的字典中有对应项，那么将对应的汉语词作为备选；否则调用字模型，计算该拼音段可能对应的词语作为备选，在把字模型和词模型的概率进行结合时，会给字模型得到的概率乘上系数 λ，对该系数也在参数调整一节中进行了简单的探究。得到备选词和对应概率之后，使用 viterbi 算法计算最大似然解即可。

2 实验效果

基于实现的功能，测试了如下模型的表现：

1. 2 元字模型
2. 考虑多音字的 2 元字模型
3. 3 元字模型
4. 考虑多音字的 3 元字模型
5. 2 元词模型 +2 元字模型
6. 2 元词模型 +3 元字模型

使用同学们贡献的测试文件（去掉了长度大于 14 的句子，共 292 句，2824 字），对以上模型进行了测试，准确率如下表：

模型	句准确率	字准确率
2 元字模型	28.08%	75.59%
2 元多音字模型	28.77%	77.05%
3 元字模型	48.97%	84.64%
3 元多音字模型	48.97%	84.95%
2 元词模型 +2 元多音字模型	45.55%	73.65%
2 元词模型 +3 元多音字模型	50.00%	75.50%

可见更高阶的字模型对准确率有较大的提高，而且考虑多音字对于准确率有一定的提高，但是随着字模型阶数的增加，多音字的影响会变小。词模型的引入对于整句的准确率有一定提高，但是由于 jieba 分词的局限性和语料的不足，词模型的准确程度并不高，在测试集上的表现也不是非常理想。下面展示一些例子：（模型 1-6 分别对应上面提到的 6 种设置）

- ru guo pin yin shou zi mu da xie hui zen me yang 如果拼音首字母大写会怎么样
    1. 如果品饮收资模大协会怎么样
    2. 如果拼音首字幕大协会怎么样
    3. 如果拼音首字母大协会怎么样

4. 如果拼音首字母大协会怎么样
5. 如果拼音首字母大写会怎么样
6. 如果拼音首字母大写会怎么样

这句话用 2 元模型翻译，因为多音字的问题，根本翻译不出来，用 2 元多音字模型稍好，但是无法将 “拼音首字母” 几个字完整拼出来。而三元模型可以拼出更长的词，但是对词于词之间的关系缺乏理解，所以会把 “xie hui” 认成高频出现的 “协会” 从而导致错误。词模型就可以准确翻译。

- yin wei wo mei tian dou zai ku 因为我每天都在哭

1. 因为我每天都仔裤
2. 因为我每天都在库
3. 因为我每天都在哭
4. 因为我每天都在哭
5. 因为我每天都再苦
6. 因为我每天都再苦

这个例子里，“zai ku” 两个字词模型翻译错了，可能是因为对语料分词时，“再苦” 被 jieba 当做一个词出现的次数有些高，所以模型优先使用了 “再苦”。

- ren gong zhi neng dao lun hen you yi si 人工智能导论很有意思

1. 人工智能导论很有意思
2. 人工智能导论很有意思
3. 人工智能岛论很有意思
4. 人工智能岛论很有意思
5. 人工智能到伦很有意思
6. 人工智能到论很有意思

这个例子与上一个例子相似，3 元模型可能反而因为 “智能岛” 在文本中多次出现，从而翻译出错；而词模型由于把 “能到” 认为是一个词，也翻译出错。

- shang hai zi lai shui lai zi hai shang 上海自来水来自海上

  1. 上孩子来说来自海上
  2. 上孩子来水来自海上
  3. 上孩子来说来自海上
  4. 上孩子来水来自海上
  5. 上海自来水来自海上
  6. 上海自来水来自海上

  这个例子显示出了词模型的优点，当句子由常用词组成时，词模型通过对拼音串进行划分，可以找到看起来最正常的解。

3 参数调整

1. Absolute discounting 的值 (D_VALUE)
   该值是样本中短语出现次数与总体中短语出现次数的差值，通过调整该值，可以微调出现次数少的短语的置信程度以及未出现短语的平滑概率。此处以带注音的 3 元字模型为例，探究 D_VALUE 对模型在测试集上表现的影响。

D_VALUE	句准确率	字准确率
0.5	48.29%	84.52%
0.75	48.97%	84.95%
0.9	49.32%	85.31%

   可以看出 absolute discounting 的方法中 D_VLAUE 的选择对于准确率的影响不是很大，部分原因是调整的空间太小，之后可以选用调整空间更大、更精确的 laplace 平滑来提高模型的表现。

2. 词模型中字模型概率的 discount factor(λ)
   由于测试太费时间，此处举例说明 λ 对结果的影响。

	ni de li jie shi dui de
ground truth	你的理解是对的
$\lambda = 0.9$	你的理解是对的
$\lambda = 0.95$	你的理解是对的
$\lambda = 1.0$	你的理解是对德

如果 λ 越小，那么字模型对词模型的影响就越大，此处“对德”在词模型里是一个出现频率不高的词，当字模型的比重稍调高时，可以得到正确的结果，但是此项调整的最终效果难以判断，因为词模型统计的语料较少，字模型又不如词模型可靠，难以找到一个合适的平衡。

4 总结

- 模型实现过程中对内存的高效利用一直是一个问题。如果可以对字和词做 hash，把概率用 8 位或 16 位表示，就可以统计更多的语料并进行计算。
- 语料库对于模型的能力有决定性的影响。我没有使用其他的语料库，但是可以预料到，如果语料库的语言习惯越接近测试集，那么结果会越准确。
- 还可以使用更加精确的平滑方法，比如将语料库分为两部分，用第一部分计算概率值，用第二部分计算平滑所用的系数。在测试集与语料库相似的前提下，应该会取得更好的效果。
- 词模型的运算速度问题。应当可以使用贪心等局部最优的方法，加速拼音划分，同时实现较高的准确率。