拼音输入法 实验报告

计 76 陈之杨 2017011377

1 实验内容

实现一个汉语拼音输入法，能够将输入拼音转化为汉字。以新浪新闻为模型训练集语料库。

2 模型建立

使用马尔可夫模型对输入法建模，即输入每一个汉字时，只考虑之前输入的汉字。

具体地，设 $p_1p_2\dots p_n$ 为输入的拼音串，$q_1q_2\dots q_n$ 为一种可能的汉字串，我们即是要最大化

$$\mathbb{P}(p_1p_2\dots p_n = q_1q_2\dots q_n) = \prod_{i=1}^{n} \mathbb{P}(p_i = q_i | p_1 = q_1, \dots, p_{i-1} = q_{i-1}).$$

考虑如何计算 $\mathbb{P}(p_i = q_i | p_1 = q_1, \dots, p_{i-1} = q_{i-1})$。当 i 较大时，该条件概率的条件空间是指数级增长的，这对于参数量的需求是巨大的。为了易于计算，采用 n-gram 模型，也即只考虑每个字之前出现的 n 个字：

$$\mathbb{P}(p_i = q_i | p_1 = q_1, \dots, p_{i-1} = q_{i-1}) \approx \mathbb{P}(p_i = q_i | p_{i-1} = q_{i-1} \dots, p_{i-n+1} = q_{i-n+1}).$$

本实验中笔者实现了基于字的二元和三元模型，即 $n = 2, 3$。

考虑如何计算 $\mathbb{P}(p_i = q_i | p_{i-1} = q_{i-1}, \dots, p_{i-n+1} = q_{i-n+1})$。假设语料库是从汉语词句集合中均匀采样的结果，那么我们只须统计出语料库中所有 n 元组出现的次数，其频率就是对应概率：

$$\begin{aligned}\mathbb{P}(p_i = q_i | p_{i-1} = q_{i-1}, \dots, p_{i-n+1} = q_{i-n+1}) &= \frac{\mathbb{P}(p_i = q_i, \dots, p_{i-n+1} = q_{i-n+1})}{\mathbb{P}(p_{i-1} = q_{i-1}, \dots, p_{i-n+1} = q_{i-n+1})} \\ &\approx \frac{\#(q_{i-n+1}, q_{i-n+2}, \dots, q_i)}{\#(q_{i-n+1}, q_{i-n+2}, \dots, q_{i-1})}.\end{aligned}$$

然而，需要注意的是，考虑到语料库不能完美反应输入词句的分布，如果一些不常用的词组在语料库中从未出现过的话，该模型会认为这个词组的出现概率为 0，也即输出中永远不会出现这个词组。为了避免这种情况，我们引入拉普拉斯平滑因子：

$$\frac{\#(q_{i-n+1}, q_{i-n+2}, \dots, q_i) + \epsilon}{\#(q_{i-n+1}, q_{i-n+2}, \dots, q_{i-1}) + k\epsilon}.$$

其中 ϵ 是一个小常数，k 是 q_i 可能取值的种数（拼音对应的汉字数）。这个平滑操作的意义就在于，即便对于语料库里从未出现过的组合，我们也认为它至少出现了 ϵ 次，以使得它能够在输出中出现。

3 算法实现

由于马尔可夫模型具备无后效性，因此采用动态规划算法求解最大概率。

对于二元模型，设 $f_{i,j}$ 表示对于输入的前 i 个拼音，第 i 个拼音对应汉字 j 的最大概率。枚举上一个汉字即可转移状态：

$$f_{i,j} = \max_k f_{i-1,k} \times \mathbb{P}(p_{i-1} = k | p_i = j).$$

其中转移概率可以通过预先统计语料库得到。转移过程中，还需要记录每一个 $f_{i,j}$ 所选择 k 的取值。最后，取 $\max_j f_{n,j}$（n 是拼音串长度）作为输出。通过记录转移来源，我们可以倒推出每一步汉字的选择。

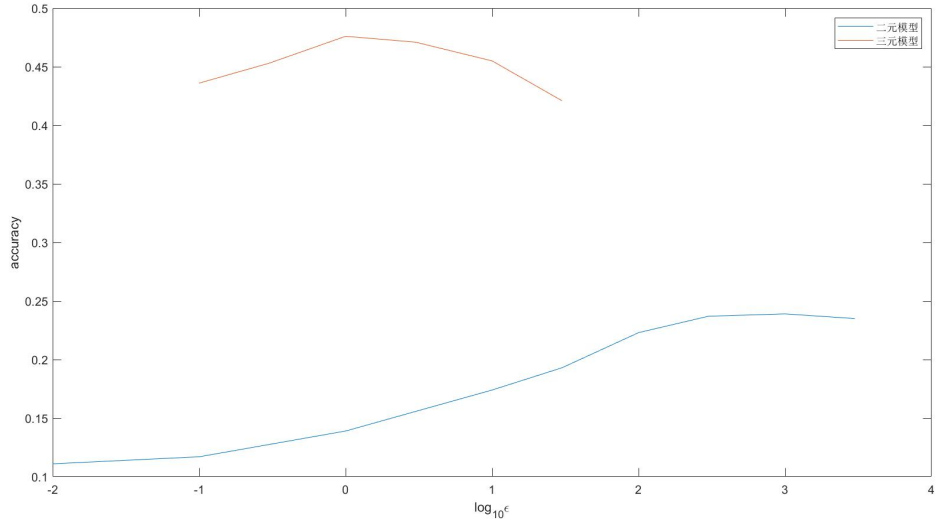
容易将此算法推广到三元模型。设 $f_{i,j,k}$ 表示对于输入的前 i 个拼音，第 $i-1$ 个拼音对应汉字 j，第 i 个拼音对应汉字 k 的最大概率。枚举第 $i-2$ 个拼音的汉字即可转移状态。

注意到汉字的种类较多，故转移概率一般较小，这导致当串长较大时 $f_{i,j}$ 的取值很小，可能会存在较大的数值精度问题，且在三元模型中尤甚。一个解决方法是对所有的概率取对数处理，将乘法转化为加法。但考虑到算法中存在着平滑因子等微小量的存在，取对数可能导致下溢发生，故笔者没有采用此方法。笔者使用了 python 语言的 `decimal` 库提高实数储存精度。由于动态规划过程中的状态不多，故该方法没有牺牲太多的时间效率。

考虑算法的时间复杂度。对于二元模型，有 $O(nk)$ 个状态（n 是拼音串长度，k 是一个拼音对应的同音字数量），转移需要枚举 $O(k)$ 次，故总的时间复杂度为 $O(nk^2)$。同样可得三元模型得时间复杂度为 $O(nk^3)$。可以将算法进一步拓展到更高维的模型，但是 n-gram 模型需要预处理所有 n 个汉字组合的出现频数，当 n 较大时所需的预处理时间和存储空间是难以忍受的。故笔者没有尝试更高维的 n-gram 模型。

4 参数选择

通过设置不同的平滑项常数 ϵ，可以调整模型的性能。笔者使用了一个自制的新闻测试集进行测试，不同 ϵ 下模型的句子准确率如图所示（为了便于显示，ϵ 坐标经过了取对数处理）。**对于二元模型，$\epsilon = 10^3$ 时可取得最大正确率 23.9%。对于三元模型，$\epsilon = 1$ 时可取得最大正确率 47.6%。**



可以发现，平滑常数过低或过高都会影响模型的性能。二元模型对平滑常数的变动较为敏感，而三元模型相对稳定。

5 样例分析

笔者使用一些样例进行了测试，得到了一些不错的结果，但也发现了一些效果较差的例子。

以下是一些效果较好的样例：

- 前国家主席江泽民
- 两会在北京召开
- 机器学习
- 警方成功抓获犯罪嫌疑人

由于语料库是新闻集合，对于一些政治类短语有较高的准确度。此外，二元和三元模型对较短的专有名词有较高的识别度。

以下是一些典型的错误样例：

- 他是一个女人（她是一个女人）
- 数值分析预算法（数值分析与算法）
- 全国大学生应与四六级考试（全国大学生英语四六级考试）

对于第一个例子，由于 n-gram 模型存在固有缺陷，无法处理相隔超过 n 的上下文信息，故难以解决。此外，n-gram 模型因为没有词语的概念，难以对文本进行正确的分词。

6 总结讨论

经过测试可以发现使用 n-gram 马尔可夫模型实现拼音输入法有一定的可行性，但是缺点也很明显。由于 n-gram 模型需要的参数随 n 指数级增长，很难将 n 进一步扩大，考虑更多的上下文信息。而且，当 n 较大时，会出现过长的短语在语料集出现次数较为稀疏的问题。三元模型比二元模型的性能有显著优势，但是难以进一步拓展。

n-gram 模型对训练集的依赖性也很强。由于无法处理语法结构，模型只有在较为接近训练集分布的测试集上才能取得较好的效果。笔者虽然没有实现基于词的 n-gram 模型，但笔者推测，使用词作为处理语言的基本单元能够更好地提高句子的连贯性。

n-gram 模型的核心问题在于 n 较小，难以考虑上下文信息。使用深度学习中的 Attention 机制等方法或许可以解决这个问题。