基于字的三元模型的拼音输入法

熊泽恩 计24

2024 年 5 月 30 日

目录

1 拼音输入法

1.1 简介

拼音输入法已经成了我们日常生活中不可分割的一部分。为了简化问题，我们只考虑全拼，即每个汉字对应的拼音串是合法的、完整的汉语拼音。现在的任务是，给定一个由全拼组成的拼音串，需要找到一个汉字串，使得这个汉字串的拼音是给定的拼音串，同时该串在语料库中出现的概率最大。概率越大，说明这个汉字串越有可能是用户想要输入的内容。

我们可以使用在《概率论与数理统计》这门课中学到的“条件概率”相关知识来解决这个问题。

1.2 输入输出格式

- 输入数据由多行拼音串构成，每行一个拼音串。拼音串中每个拼音之间用空格隔开。不含标点符号、阿拉伯数字等内容。
- 输出数据由多行汉字串构成，每行一个汉字串。汉字串中每个汉字之间没有空格。

输入(input.txt)	输出(output.txt)
gai lv lun yu shu li tong ji	概率论与数理统计
pin yin shu ru fa	拼音输入法
zhe shi yi ge ce shi	这是一个测试
ni hao ma	你好吗

表 1: 样例输入输出

2 基于字的三元模型

2.1 基本思路

设 $\mathcal{F}$ 为所有合法拼音构成的集合，$\mathcal{G}$ 为所有合法汉字构成的集合。从合法拼音的集合到合法汉字的集合的映射关系为 $\sigma : \mathcal{F} \to 2^{\mathcal{G}}$，其中 $2^{\mathcal{G}}$ 表示 $\mathcal{G}$ 的幂集。σ 表示了一个拼音对应的所有可能的汉字。

对于一个由 n 个拼音构成的序列 $S = s_1 s_2 \cdots s_n$，我们需要需要确定每个单字拼音 s_i 对应的中文字符 w_i，使得中文序列 $w_1 w_2 \cdots w_n$ 最佳。给定 $S = s_1 s_2 \cdots s_n$，我们定义 $P(w_1 w_2 \cdots w_n)$ 为 $w_1 w_2 \cdots w_n$ 与 S 的匹配概率。

形式化地，我们需要找到一个 $W = w_1 w_2 \cdots w_n$，满足 $w_i \in \sigma(s_i), \forall i \in \{1, 2, \cdots, n\}$，使得

$$P(w_1 w_2 \cdots w_n)$$

最大。

2.2 公式推导

根据条件概率的定义，我们有

$$\begin{aligned} \ln P(w_1 w_2 ... w_n) &= \ln P(w_1) + \ln P(w_2 \mid w_1) + \ln P(w_3 \mid w_1 w_2) + \cdots + \ln P(w_n \mid w_1 w_2 \cdots w_{n-1}) \\ &= \sum_{i=1}^{n} \ln P(w_i \mid w_1 w_2 \cdots w_{i-1}). \end{aligned} \tag{1}$$

这一结果也是很容易理解的：我们可以将 $w_1w_2\cdots w_n$ 看作是一个序列，找到最佳的 w_i 的过程即为“汉字接龙”的过程，在给定 $w_1w_2\cdots w_{i-1}$ 的条件下，我们需要找到一个 w_i 使得 $w_1w_2\cdots w_i$ 的匹配概率最大。例如，输入数据为 `dong wu yuan`，我们已经得到 $w_1 = 动, w_2 = 物$，在这个条件下，我们需要找到 $w_3 \in \sigma(\texttt{yuan})$ 使得 $P(动物 \mid w_3)$ 最大。由于园能使该概率最大，因此最合理的选择。

在基于字的 m 元模型中，我们认为在长度为 m 的汉字串中，w_i 仅仅与前 $m-1$ 个汉字有关。因此，我们可以将上述公式进一步简化为

$$\sum_{i=1}^{n} \ln P(w_i \mid w_{i-m+1}\cdots w_{i-1}),$$

当 $m = 3$ 时，即为基于字的三元模型，我们需要最大化 $\sum_{i=1}^{n} \ln P(w_i \mid w_{i-2}w_{i-1})$，或者等价地，求

$$\sum_{i=1}^{n} (-\ln P(w_i \mid w_{i-2}w_{i-1}))$$

的最小值。

由条件概率公式，对于每一项而言，有

$$P(w_i \mid w_{i-2}w_{i-1}) = \frac{P(w_{i-2}w_{i-1}w_i)}{P(w_{i-2}w_{i-1})}.$$

再根据极大似然的思想，由于数据量足够大，所以可以用频次之比来估计概率之比。所以，上式可以近似为

$$P(w_i \mid w_{i-2}w_{i-1}) \approx \frac{n(w_{i-2}w_{i-1}w_i)}{n(w_{i-2}w_{i-1})},$$

其中 $n(\cdot)$ 表示该汉字串在数据中的出现次数。

2.3 图论建模

我们可以将上述问题建模为一个有向无环图(Directed Acyclic Graph, DAG)，如下所示：

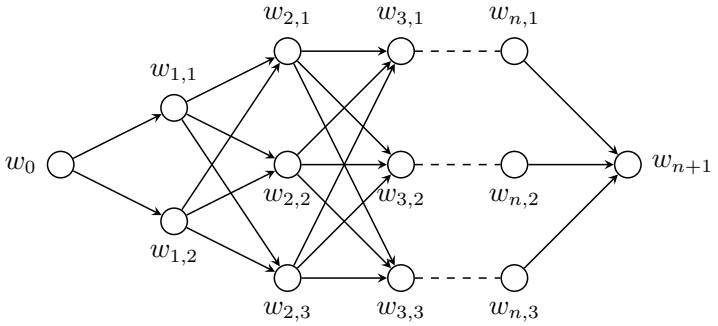

图 1: 有向无环图

建图如下：其中 w_0 为虚拟源点，w_{n+1} 为虚拟汇点。给 $\sigma(\cdot)$ 中的元素排序，记 $w_{i,j}$ 为拼音 $s_i \in \mathcal{F}$ 所对应的第 j 个汉字。则 $w_{i-1,k}$ 与 $w_{i,j}$ 之间的边权满足

$$\text{cost}(w_{i-1,k}, w_{i,j}) = \begin{cases} -\ln(P(w_{i,j})), & \text{if } i = 1, \\ -\ln(P(w_{i,j} \mid w_{i-1,k})), & \text{if } i = 2, \\ \min\limits_{v \in \sigma(s_{i-2})} (-\ln(P(w_{i,j} \mid v\ w_{i-1,k})), & \text{if } 3 \le i \le n, \\ 0, & \text{if } i = n+1. \end{cases}$$

这样我们得到了一个有向无环图。题目所求即为 w_0 到 w_{n+1} 的路径最小值。

2.4 动态规划求解

DAG 上最短路可以使用动态规划的思想来解决。设 $f(i,j)$ 表示从 w_0 开始，到达 $w_{i,\cdot}$ 这一层对应的 $w_{i,j}$ 所需边权和的最小值。则对于某一点 $w_{i,j}$ 而言，可以枚举它从上一层的哪一个点 $w_{i-1,k}$ 转移过来，即可得到状态转移方程：

$$f(i,j) = \max_{k \le |\sigma(s_{i-1})|} (f(i-1,k) + \mathrm{cost}(w_{i-1,k}, w_{i,j})).$$

题目所求的 w_0 到 w_{n+1} 的路径最小值，即为 $f(n+1, w_{n+1})$。

3 实验结果

定义字准确率 p_1 为

$$p_1 = \frac{\text{汉字正确个数}}{\text{总字数}},$$

句准确率 p_2 为

$$p_2 = \frac{\text{所有汉字均正确的句子个数}}{\text{总句数}}.$$

除了三元模型以外，我还实现了模型更为简单的二元模型，即 w_i 的选择仅与 w_{i-1} 有关；最终得到实验结果如下表所示：

评价指标	二元模型	三元模型
句准确率	38.92%	46.51%
字准确率	83.72%	87.53%
平均单次响应时长	0.0135 s	0.5909 s
总响应时长	6.801 s	296.1 s

表 2: 在测试集上的实验效果

实验效果较为理想。由于三元模型比二元模型复杂，所以对应的两种准确率均更高；而模型的复杂性也会导致所用计算时间更长，使得响应时长更长。代码实现和实验数据已上传并开源于 [1]。

4 总结

总而言之，我使用条件概率的方法，将拼音输入法问题建模为一个有向无环图上的最短路问题，通过使用和极大似然的思想，将图上边权的计算简化为数据中的频次之比，并最终使用动态规划的方法求解。

事实上，部分自然语言处理问题背后的原理与拼音输入法问题类似，本质上都是在已确定输出的前 i 个 token 的情况下，预测第 $i+1$ 个 token 的概率分布，并输出概率最大的 token。

我觉得这个问题的解决方法还是比较有意义的，因为拼音输入法是我们日常生活中不可或缺的一部分，概率论与数理统计的知识也是非常重要的。我通过利用课上学到的知识，将这两者结合起来，解决了一个实际问题，这让我感到非常有成就感。

参考文献

[1] Ze-en Xiong, *IAI_2022/homework/connect-4 at main · zhaochenyang20/IAI_2022 · GitHub*, 2022, https://github.com/zhaochenyang20/IAI_2022/tree/main/homework/connect-4.