人工智能导论 第一次作业 拼音输入法 实验报告

骆科运 致理书院-信计21 2022012405

一、实验环境介绍：

实验在windows11系统下进行，采用编程语言为python 3.13.2。

二、语料库与数据预处理方法介绍：

采用的语料库为新浪新闻2016年新闻语料库（sina_news_gbk）。

数据预处理方法：

1. 首先从拼音汉字表中提取汉字与拼音的多对多映射；
2. 提取每一篇新闻的html部分（即新闻正文），然后根据处理好的拼音汉字表将所有新闻正文划分为若干个文本段。（文本段长度在50000左右）

   每个文本段由若干句子（句子内部无任何符号）组成，句子之间用一个问号"?"分隔，无其他任何符号。

   然后使用**pypinyin**库对一个文本段注音（使用问号分割以及长度控制都是便于pypinyin处理），得到文本段的拼音。

   再从文本段中识别出每个短句，找到与其对应的拼音序列，然后扫描它并处理出**单字出现频率、两字出现频率、三字出现频率、四字出现频率 和 单字作为句首句尾出现频率，这些内容以json格式存储。**

三、基于字的二元模型的拼音输入法的介绍：

a. 原理与算法：

1. 由于汉字存在多音字的情形，因此我们将一个汉字和其一个发音组成一个唯一的pair，比如（“国”，“guo”），这样来处理多音字就不会有任何错误；
2. 对于一个给定的拼音序列O，我们实际上是需要找到一个pair序列S，使得P(S|O)最大。

   比如对于拼音序列 qing hua da xue，我们需要找到 ('清'，‘qing’）（‘华’，‘hua’） （‘大’，‘da’）（‘学’，‘xue’）。

   公式推导如下：

$$P(S|O)=\frac{P(S,O)}{P(O)}=\frac{P(O|S)*P(S)}{P(O)}$$

   注意到，由于S是pair序列，而每个pair都包含汉字和拼音，因此事实上有

$$P(O|S)=\begin{cases}1\text{，}S\text{中每个}pair\text{的拼音与}O\text{中对应位置拼音一致}\\0\text{，不一致}\end{cases}$$

   而我们进行搜索（动态规划）时，可以只去搜索满足拼音一致的pair序列，因此所有的P(O|S)值都**严格**为1，而P(O)与S无关，对于固定的O为常数，因此P(S|O)事实上可以由P(S)表示。

   对于一个pair序列S，

$$S=s_1s_2\dots s_n$$
$$P(S)=P(s_1)P(s_2|s_1)P(s_3|s_1s_2)\dots P(s_n|s_1\dots s_{n-1})$$

   在基于字的二元模型中，我们仅考虑一个字和它前一个字的耦合，忽略一个字与它前面一个字以前的所有字的关系，即

$$P(s_n|s_1s_2\dots s_{n-1})\approx P(s_n|s_{n-1})$$

   此时

$$P(S) = P(s_1)\prod_{i=2}^{n} P(s_i|s_{i-1})$$

其中$P(s_1) = \frac{cnt(s_1)}{total_{cnt}}$ ， $cnt(s_1)$是$pair\ s_1$在语料库中出现的次数，$total_{cnt}$是语料库中所有$pair$出现的总次数

$$P(s_i|s_{i-1}) = \frac{cnt(s_{i-1}s_i)}{cnt(s_{i-1})}$$ ， $cnt(s_{i-1}s_i)$是$s_{i-1}s_i$这两个$pair$在语料库中连续出现的次数

做变换：

$$-log(P(S)) = -log(P(s_1)) - \sum_{i=2}^{n} log(P(s_i|s_{i-1}))$$ ，将做变换过后的值记为F

3. 基于上述公式推导，我们可以确定以下算法流程：

    1. 首先对语料库进行预处理**（“训练”）**，处理好所有的$cnt(s_i)$与$cnt(s_{i-1}s_i)$;
    2. 对于第i个拼音，假设s_i是一个满足条件的pair，那么由二元模型的性质，计算所有以s_i结尾的pair序列的概率时，与前i-2个拼音对应什么pair都没有任何关系，而仅仅与第i-1个pair是什么有关系，这样的性质导出了如下的**动态规划**算法：

        记录所有以s_i pair结尾的pair序列中F值最小的pair序列是什么，以及其对应的F值是多少，然后按如下动规公式递推：

        $$F(s_i)表示所有以s_i结尾的序列中最小的F值$$

        $$F(s_i) = \begin{cases} min_{s_{i-1}}\{\ F(s_{i-1}) - log(P(s_i|s_{i-1}))\ \}\ ,\ i \geq 2 \\ -log(P(s_1))\ ,\ i = 1 \end{cases}$$

        在递推过程中记录取到最小F值的路径（即从头开始的pair序列）即可。

b. 实验效果：

选择新浪新闻作为语料库，训练时间为$1.5h$左右，主要耗费在调用**pypinyin**注音上。

TODO：准确率、生成测试样例总时间

c. 例子分析：

好的例子：人工智能技术发展迅猛；

这个句子中成分都可划分为二字词语，“人工“，”智能“，”技术“，”发展“，”迅猛“，而且这些二字词语与同拼音的其他二字词语相比较，其出现频率呈现断崖式的高，因此最终能够非常”稳妥“地锁定正确答案，其F值除以字数为3.92，属于较确定的猜测。

坏的例子：地道战是一种以地道为策略应用的陆军步兵战术；

运行结果：地道站是一种异地到位策锃应用的陆军步兵战术；

这个句子结果不好，其F值除以字数达到了7.19，属于较犹豫的猜测。分析其原因主要在于句子不能完全划分为二字词语，存在一些较复杂的语法，比如”是一种“、“...的...”等。同时”以地道为“又存在一个较明显的错误划分“异地”“到位”，因此对猜测产生了一定混淆。

四、思考题：

a. gbk和utf-8的区别：

gbk是一种编码范围包括ASCII码以及简体中文的编码方式，ASCII码占1字节，中文字符占2字节；

utf-8是一种编码范围覆盖全球所有语言的字符的编码方式，采用可变长度编码（1字节到4字节），ASCII码占1字节，中文字符占3字节；

总的来说，utf-8相比gbk更加通用、国际化，往往用于处理跨平台、数据库、网页等场景，而GBK仅仅适用于中文场景。但在处理中文字符时gbk编码比utf-8占用更小的空间。

b. 假设读音数量为常数m，则大小为V的字库（且读音均匀分布）中平均一个读音有$\frac{V}{m}$个对应的字。

那么在使用**viterbi**算法进行动态规划时，句子长度为n，则需要进行$O(n)$层转移，每层转移需要计算$O(\frac{V}{m})$个F值，而每计算一个F值需要考虑$O(\frac{V}{m})$种可能的转移（上一个字是什么），因此总共时间复杂度为$O(\frac{nV^2}{m^2})$。如果考虑m为常数，则时间复杂度为$O(nV^2)$。

在进行动态规划时可以使用**滚动数组**的空间优化技巧，这样空间复杂度为$O(\frac{V}{m})$，考虑m为常数，则为$O(V)$。但训练得到的数据集也是空间复杂度的一个重要部分，理论上每两个字都可能成为一对连续出现的两字，存储它们出现次数的空间复杂度为$O(V^2)$，因此总空间复杂度应为$O(V^2)$。