

拼音输入法实验报告

计94 周堯 2019011301

一、程序运行方法

基于 zhihu 语料库的字二元拼音输入法

如果没有装有python环境，可运行可执行文件运行。在根目录下依次执行以下操作：

```
cd bin
pinyin xxx yyy
```

其中 xxx 表示输入文件的路径，yyy 表示输出文件的路径。若不指定（即无这两个参数），则使用程序默认的路径。默认路径中，输入文件在input文件夹中，输出文件在result文件夹中。示例：

```
pinyin ../input/input_py.txt ../result/output.txt
```

（暂不支持路径中包含不存在的文件夹）

或者如果装有python环境，建议直接使用python运行。在根目录下依次执行以下操作：

```
cd zhihu
python pinyin.py xxx yyy
```

其中 xxx 表示输入文件的路径，yyy 表示输出文件的路径。若不指定（即无这两个参数），则使用程序默认的路径。其中，指定的路径相对路径、绝对路径均可。（暂不支持路径中包含不存在的文件夹）

基于 sina 语料库的字二元拼音输入法

仅支持 python 运行。

```
cd sina
python pinyin.py xxx yyy
```

其中 xxx 与 yyy 的含义与上一部分相同

准确率计算

在根目录下进入 calcu 文件夹

```
cd calcu
```

有两个python脚本可以执行以计算整准确率

```
python calcu.py xxx
```

执行上述命令以计算 result 文件夹下所有结果文件的句**句准确率**。其中，xxx 表示标准答案文件的路径。不添加则使用默认答案文件。结果输出在当前目录下。

或者：

```
python calcu_char.py
```

执行上述命令以计算 result 文件夹下所有结果文件的**字准确率**。其中，xxx 表示标准答案文件的路径。不添加则使用默认答案文件。结果输出在当前目录下。

二、基本思路和实现过程

基础功能版本的拼音输入法，采用字二元模型。

字二元模型

字二元模型是以字为单位，并假设某个字出现的概率仅取决于前一个字的模型。字二元模型下，后验概率被定义为：

$$P(w_i|w_{i-1}) = \frac{w_i w_{i-1} \text{ 出现的次数}}{w_{i-1} \text{ 出现的次数}}$$

其中，字以及字的组合出现的次数从预训练得到的语料库中获取。而每句话第一个字，做如下补充定义：

$$P(w_1|w_0) = \frac{w_1 \text{ 出现的次数}}{\text{语料库总字数}}$$

那么，使用字二元模型求解拼音输入法的目标就是，求使得长为 n 的字符串的如下评分：

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

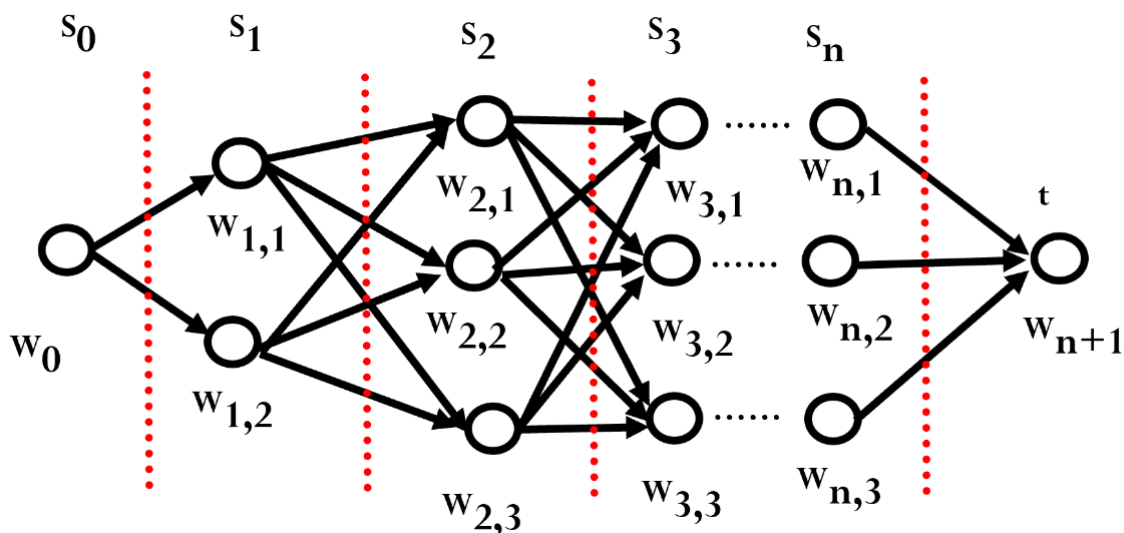
最大的一个。

语料库

语料库的选取上，起先选取了课程

Viterbi算法

动态规划算法。



如图。对于每个读音 s_i ，其在拼音汉字表中对应的汉字有 k_i 个，每个字对应一个节点，这个节点储存算到该字为止的最优分数，以及该分数对应的路径（字符串）。对于这 k_i 个节点，分别根据上一层算得的分数 w 来递推得到该节点的分数 $w_{i,1}, w_{i,2}, \dots, w_{i,k_i}$ 。重复以上过程，直至到最后一个拼音 s_n ，再从最后一层节点中取出分数最大的，其分数便对应全局的最高分，

注意点

平滑

为了防止出现 $P(w_i|w_{i-1}) = 0$ 的情况，导致概率计算无效化，需要在分数的计算上作平滑处理。

用如下替换（线性插值平滑）：

$$P(w_i|w_{i-1}) \rightarrow \lambda P(w_i|w_{i-1}) + (1 - \lambda)P(w_i)$$

即可规避这样的问题。其中， λ 为一个接近1（小于1）的参数，其选取影响到最后拼音输入法的准确率，是需要进行调参的重点对象。

三、实验效果

准确率

基于 zhihu 语料库的字二元拼音输入法，可以在课程群中给的测例上做到82.6%的子准确率，以及38.1%的句准确率。在实际实验中，一些具体的实例如下：

好的例子

- ren gong zhi neng dao lun hen you yi si
人工智能导论很有意思
- qing hua da xue shi shi jie yi liu da xue
清华大学是世界一流大学
- shen du you xian sou suo xu yao si kao
深度优先搜索需要思考
- wo men yao ai xi shi jian jiu xiang ai xi zi ji de sheng ming yi yang
我们要爱惜时间就像爱惜自己的生命一样
- jing ji jian she he wen hua jian she tu chu le shi ba da jing shen de zhong yao xing
经济建设和文化建设突出了十八大精神的重要性
- hu lian wang hai neng gou cheng wei tu shu chuan bo de ping tai
互联网还能够成为图书传播的平台
- shi qi zhang pai ni neng miao wo
十七张牌你能秒我
- qian fang gao neng fei zhan dou ren yuan qing kuai su che li
前方高能非战斗人员请快速撤离
- xiao chu kong ju de zui hao ban fa jiu shi mian dui kong ju
消除恐惧的最好办法就是面对恐惧

分析：由于用的是知乎上爬取的中文文本，因此在语言使用风格上大概是近十年的风格，在一些新颖的搭配上表现较好。

坏的例子

- yi zhi ke ai de da huang gou
一直可爱的大黄狗（一只可爱的大黄狗）
分析：同音字问题，一只、一直都是高频词，分数相近，较难辨认。
- ta shi wo de mu qin
他是我的母亲（她是我的母亲）
分析：同音字，在语义上的分辨难以在字二元模型上体现。
- shang hai zi lai shui lai zi hai shang
上孩子来说来自海上（上海自来水来自海上）
分析：词语分割，上海、孩子都是高频词，分数相近，难以辨认；多音字，在我的模型中没有处理。
- shi jie qi xiang zu zhi jin tian fa biao sheng ming
世界奇想阻止今天发飙升名（世界气象组织今天发表声明）
分析：同音词语，较难辨认
- qiang lie qian ze di lie she hui fen zi
强烈谴责的裂社会分子（强烈谴责低劣社会分子）
分析：多音字，“的”字有“di”读音，出现频率过高。
- wo he peng you men zheng zai wu you wu lv de wan shua
我和朋友们正在无忧无虑的玩耍（我和朋友们正在无忧无虑地玩耍）
分析：同音字，“的”“地”“得”在不引入外部知识的情况下实在难于区分。

缺陷分析

如上例子可见，有以下几个问题：

1. 语料库仍不够大、不够完善，部分常用词的频率低于一些奇怪的搭配，与真实使用时的词频有略微的偏差
2. 同音字词频率相近，在一些情况下容易辨认错
3. 多音字的问题没有解决，一些高频字有一些不常用读音，会带来极大的困扰
4. 字二元模型中，一个字之后前后两字有关联，无法通过整句语境来很好地决定用词。同音、近义的词语（如他她它、的地得）在不引入外部知识的情况下难以准确区分。

四、参数选择

不同模型下的字准确率：

```

..\result\output.txt : acc:82.62879788639366%
..\result\sina_0.99999.txt : acc:81.80317040951122%
..\result\sina_jieba_0.99999.txt : acc:77.88969616908851%
..\result\sina_nbest_0.999992.txt : acc:81.67107001321003%
..\result\sina_seg_0.99999.txt : acc:58.60303830911493%
..\result\zhihu_0.99992.txt : acc:81.5554821664465%
..\result\zhihu_0.99995.txt : acc:82.56274768824306%
..\result\zhihu_0.99998.txt : acc:82.67833553500661%
..\result\zhihu_0.999985.txt : acc:82.76089828269485%
..\result\zhihu_0.999988.txt : acc:82.76089828269485%
..\result\zhihu_0.99999.txt : acc:82.62879788639366%
..\result\zhihu_0.999992.txt : acc:82.612285336856%
..\result\zhihu_0.999995.txt : acc:82.21598414795245%
..\result\zhihu_jieba_0.999999.txt : acc:75.69352708058125%
..\result\zhihu_nbest_0.999992.txt : acc:82.21598414795245%
Done.

```

其中，文件名中的数字表示平滑参数 λ ；zhihu / sina 表示该模型基于的语料库。（nbest在后面一章节会提到）

可以看到， $\lambda = 0.999985$ 或 0.999988 时字准确率最高。

不同模型下的句准确率：

```

..\result\output.txt : acc:38.095238095238095%
..\result\sina_0.99999.txt : acc:33.66174055829228%
..\result\sina_jieba_0.99999.txt : acc:24.794745484400657%
..\result\sina_nbest_0.999992.txt : acc:56.97865353037766%
..\result\sina_seg_0.99999.txt : acc:19.21182266009852%
..\result\zhihu_0.99992.txt : acc:34.646962233169134%
..\result\zhihu_0.99995.txt : acc:36.7816091954023%
..\result\zhihu_0.99998.txt : acc:37.274220032840724%
..\result\zhihu_0.999985.txt : acc:37.60262725779967%
..\result\zhihu_0.999988.txt : acc:38.25944170771757%
..\result\zhihu_0.99999.txt : acc:38.095238095238095%
..\result\zhihu_0.999992.txt : acc:37.93103448275862%
..\result\zhihu_0.999995.txt : acc:37.4384236453202%
..\result\zhihu_jieba_0.999999.txt : acc:20.689655172413794%
..\result\zhihu_nbest_0.999992.txt : acc:59.60591133004927%
Done.

```

可以看到， $\lambda = 0.999988$ 时句准确率最高。

因此，采用 $\lambda = 0.999988$ 为最终的平滑参数。

五、扩展功能

(1) n-best

对于普通的 viterbi 算法，只能得到全局分数最高的 **1个** 句子作为最终结果。而在实际的输入法中，由于之前提到的缺陷，必会出现模棱两可的选项需要进行人工选择，此时就需要给出 **n个** 分数最高的句子。

显然，简单地在最后一层取前 n 个是断然不行的，这样的结果不一定是全局最高的 n 个句子。如何求全局分数前 n 高的句子呢？我的做法是，为每一个节点维护一个容量上限为 n 的小顶堆。每个节点**不再只存一个分数、一个路径**，而是**存储这么一个小顶堆，包含至多 n 个分数、 n 条路径**。这样，即可保证最终能取到全局前 n 高分数的结果。

实际输出的示例如下：

```
result > zhihu_nbest_0.999992.txt
1 背景是一个美丽的城市 北京是一个美丽的城市 北京市一个美丽的城市 北京师一个美丽的城市 北京时一个美丽的城市
2 一直漂亮的消化矛 一直漂亮的小花猫 一直漂亮的消化貌 一直漂亮的笑话貌 一直漂亮的小花毛
3 一直可爱的大黄狗 意识可爱的大黄狗 以至可爱的大黄狗 一直可爱的大荒沟 一只可爱的大黄狗
4 机器学习机器应用 机器学习期期英勇 机器学习期期应用 记起学习机器应用 激起学习机器应用
5 人工智能记抒发展迅猛 人工智能技术发展迅猛 人共只能记抒发展迅猛 人共只能技术发展迅猛 人公只能记抒发展迅猛
6 每个思念一次的奥运会就要找开了 每个思念一次的奥运会就要召开了 每个思念一次的奥运会就咬着了 每个四年一次的奥运会就要找开了 每个思念一次的奥运会
7 今年年轻狂不太好 今年情况不太好 近年年轻狂不太好 近年情况不太好 今年年轻狂补太好
8 激动车驾驶员培训手册 激动车驾驶员培训收测 激动车驾驶员培训手厕 激动车驾驶员培训手策 已动车驾驶员培训手册
9 给阿姨到一杯卡布奇诺 给阿姨道一杯卡布奇诺 给阿姨到一杯咖不奇诺 给阿姨到一杯咖不起挪 给阿姨倒一杯卡布奇诺
10 如果拼音首字母打叶会怎么样 如果拼音首字母大写会怎么样 如果拼音首字幕大写会怎么样 如果拼音首字莫大写会怎么样 如果拼音首字母打协会怎么样
11 拼音输入法测试阳历证给 拼音输入法测是阳历证给 拼音输入法测试阳历证其 拼音输入法测试阳历正给 拼音输入法测试阳历证基
12 拼音之间用空格格开 品因之间用空格格开 频因之间用空格格开 品饮之间用空格格开 拼音之间用空哥哥开
13 情不要输入奇怪的车子 情不要输入奇怪的柜子 情不要输入奇怪的橘子 请不要输入奇怪的车子 亲不要输入奇怪的车子
14 她养了一直庆蛙当宠物 他养了一直庆蛙当宠物 她养了一致青蛙当宠物 她养了一直青蛙当宠物 他养了一致青蛙当宠物
15 他是我的母亲 她是我母亲 踏实我的母亲 它是我的母亲 他时我的母亲
16 你的理解释对的 你的理解是对的 你的理解释对得 你的离家是对的 你理解解释对的
17 开通仅仅四十八小时习氛二十九晚 开通仅仅四十八小时细分二十九晚 开通仅仅四十八小实习氛二十九晚 开通仅仅四十八校实习氛二十九晚 开通仅仅四十八小事氛
18 为几百科室一个网络百科全书项目 为几百科是一个网络百科全书项目 为几百可是一个网络百科全书项目 为即白可是一个网络百科全书项目 危急败可是一个网络百科
19 不再滑下 不在华夏 不在滑下 不再花下 不在画下
20 你在干什么啊团长 呢在干什么啊团长 你再干什么啊团长 贼在干什么啊团长 你在干什么啊团账
21 我去给你买一个橘子 我去给你买一个柜子 我去给你买一个车子 我去给你买一个句子 我去给你买一个车自
22 去南极痴迷分开又不 去南极痴迷分开开有不 去南极痴迷分开开友不 去南极痴迷分开开游不 区南极痴迷分开又不
23 他的人生轨迹就是一个贪心酸发 她的人生轨迹就是一个贪心酸发 他的人生轨迹就是一个贪心酸发 她的人生轨迹就是一个贪心酸发 他的人生轨迹就是一个贪心酸发 她的人生轨迹就是一个贪心酸发
24 永远距步最有酒的人到不了全局最有 永远距步最有酒的人到不了全剧最有 永远距不最有酒的人到不了全局最有 永远距不最有酒的人到不了全剧最有 永远距步最有酒的人到不了全局最有 永远距步最有酒的人到不了全剧最有
25 知道博士毕业迷茫的时候 直到博士毕业迷茫的时候 识到博士毕业迷茫的时候 知道波士毕业迷茫的时候 直到波士毕业迷茫的时候 直到波士毕业迷茫的时候
26 才发现自己没有想好要做什么成为怎样的人 才发现自己没有想好做什么成为怎样的人 才发现自己没有想好要做什么称为怎样的人 才发现自己没有想好要做什么称为怎样的人 才发现自己没有想好做什么成为怎样的人 才发现自己没有想好做什么成为怎样的人
27 随便学点计算机找工作 随便血点计算机找工作 随便血点计算机找工作 随便血点计算机找工作 随便血点计算机找工作 随便血点计算机找工作
28 变成了一个高学历思考能力超群的普通人 变成了一个高血立思考能力超群的普通人 变成了一个高学立思考能力超群的普通人 变成了一个高学立思考能力超群的普通人 变成了一个高学立思考能力超群的普通人 变成了一个高学立思考能力超群的普通人
29 一个自认为潜力不凡的人通过艰苦着觉得努力 一个自认为潜力不凡的人通过肩背着觉得努力 一个自认为潜力不凡的人通过艰苦卓绝的努力 一个自认为潜力不凡的人通过艰苦卓绝的努力 一个自认为潜力不凡的人通过艰苦卓绝的努力 一个自认为潜力不凡的人通过艰苦卓绝的努力
30 最后成为了一个在连书写代码的普通人 最后成为了一个在练书写代码的普通人 最后成为了一个在陪书写代码的普通人 最后成为了一个在再联书写代码的普通人 最后成为了一个在再联书写代码的普通人 最后成为了一个在再联书写代码的普通人
```





















可以看到，红线部分画出来的这些，都是并非分数最高，但是也在前列的句子。这些句子作为备选选项，是有极大参考意义的，这些备选选项的加入，让viterbi算法的收益被进一步激发了。

采用 n-best 的算法后，如果正确答案出现在这前 n 个句子中就判定为正确，则句准确率可以达到惊人的 59.6！。考虑到该测试集中存在许多奇奇怪怪的样例，这样的结果可以说是较为不错的。

若要运行 n-best 方法的程序，请进入 zhihu 或 sina 文件夹执行名为 pinyin_nbest.py 的 python 文件即可。

(2) 爬取知乎汉语语料

爬取了知乎回答最多的二十多个问题下的回答。采集了共200+MB的html资料，其中包含大量的中文语料材料。其优点是时间较近，网络用语较多，风格较为多变。但由于个人爬虫的限制，整体语料库大小还不是太大，不够完整全面。

名称	修改日期	类型	大小
 zhihu_01	2021/1/25 11:11	文本文档	935 KB
 zhihu_02	2021/1/25 12:38	文本文档	13,276 KB
 zhihu_03	2021/1/25 14:21	文本文档	3,807 KB
 zhihu_04	2021/1/25 15:32	文本文档	11,945 KB
 zhihu_05	2021/1/25 17:42	文本文档	13,285 KB
 zhihu_06	2021/1/25 20:06	文本文档	7,827 KB
 zhihu_07	2021/1/25 22:43	文本文档	4,988 KB
 zhihu_08	2021/1/26 1:08	文本文档	14,780 KB
 zhihu_09	2021/1/26 11:46	文本文档	20,741 KB
 zhihu_10	2021/1/26 15:06	文本文档	2,632 KB
 zhihu_11	2021/1/26 17:32	文本文档	2,338 KB
 zhihu_12	2021/1/27 3:22	文本文档	17,220 KB
 zhihu_13	2021/1/27 16:27	文本文档	9,259 KB
 zhihu_14	2021/1/27 19:44	文本文档	18,315 KB
 zhihu_15	2021/1/27 21:36	文本文档	4,551 KB
 zhihu_16	2021/1/27 22:22	文本文档	13,514 KB
 zhihu_17	2021/1/28 0:23	文本文档	1,620 KB
 zhihu_18	2021/1/28 10:26	文本文档	10,638 KB
 zhihu_19	2021/1/28 12:34	文本文档	7,715 KB
 zhihu_20	2021/1/28 15:02	文本文档	1,505 KB
 zhihu_21	2021/1/28 18:43	文本文档	3,672 KB
 zhihu_22	2021/1/29 17:44	文本文档	12,785 KB
 zhihu_23	2021/1/29 18:39	文本文档	6,796 KB
 zhihu_24	2021/1/30 12:49	文本文档	10,650 KB
 zhihu_25	2021/1/30 14:22	文本文档	3,939 KB
 zhihu_26	2021/1/31 12:27	文本文档	1,604 KB
 zhihu_27	2021/1/31 18:15	文本文档	27,372 KB

(3) 拼音分割

在根目录下的 segment.py 文件中实现了将无空格拼音进行分割的 `segment` 函数：

```
def segment(pylist):
    length = len(pylist)
    cur = 0
    res = []
    buf = ""
    match = ""
    greedy = False
    while True:
        if cur >= length:
            if buf != "":
                res.append(buf)
            break
        match += pylist[cur]
        if match in all_list:
            buf = match
            greedy = True
        elif greedy:
            if buf != "":
                res.append(buf)
            buf = ""
            match = match[-1]
            greedy = False
        cur += 1
    return res
```

该方法的准确率达到 55%，主要出错的原因在于该方法为贪婪匹配，暂无法做到智能识别。如 “ju'an'si'wei”（居安思危）这类有多种分割方式的词句难以做到很好的分割。这是一大困难。

不过，我认为这是一个**必要**的想法，因为实际拼音输入法获取键盘输入时并没有人来把它自动分割成单独的一个个音节，这一步工作是必备的。只不过实际实现的效果还有待优化，值得进一步探究。

(4) 统计词频时的分词

在预处理时，曾经想过利用 jieba 分词的第三方库先进行分词，再去统计词频。但实际上，有些字字对的出现并不一定被判定为在一个词语中，因此这样做的效果可能还不如不分词。后来的实验证明，分词实在是多此一举了，不光画蛇添足，还使准确率下降了小几个点。

六、总结与改进

基于字二元模型的拼音输入法已经能达到不俗的效果，但在上下文联系、同音字词辨析、多音字上表现较为乏力，会出现一些错误。采用 n-best 策略后，输入法整体更加完整，功能更为强大了。















需要改进的方面有以下几点：

1. 可采用字三元模型
2. 可采用词二元、三元模型
3. 解决多音字的问题
4. 进一步扩充语料库。不要停止与训练！

这些方面在今后有时间的时候可能会去优化。

本次实验让我在一个实际问题中实践了启发式搜索问题，对其理解更加深刻了。同时，拼音输入法的实现也要求多方面的能力，爬虫、文本清洗、数据统计整理、动态规划算法、数据结构设计、文件读写、调参优化等各个方面都有涉及，极大地锻炼了我的能力。

七、文件目录说明

 __pycache__	2021/4/3 14:58	文件夹	
 bin	2021/4/7 1:23	文件夹	
 calcul	2021/4/4 10:01	文件夹	
 input	2021/4/3 13:30	文件夹	
 result	2021/4/7 1:15	文件夹	
 sina	2021/4/3 14:11	文件夹	
 spider	2021/4/3 13:20	文件夹	
 src	2021/4/6 8:07	文件夹	
 zhihu	2021/4/7 1:23	文件夹	
 __init__	2021/1/24 14:07	Python 源文件	0 KB
 all_list	2021/4/6 8:10	Python 源文件	6 KB
 segment	2021/4/3 14:49	Python 源文件	6 KB
 拼音输入法 周葵	2021/4/7 2:07	Markdown File	13 KB
 拼音输入法 周葵	2021/4/7 2:07	Microsoft Edge PD...	1,423 KB

- 根目录下包含若干文件夹以及拼音分割所需要的 python 文件，以及report(.pdf & .md)
- bin 目录下是可执行文件（包含依赖项，因此请勿将该可执行文件移至其他目录运行）
- calcul 目录下是执行准确率计算的 python 文件
- input 目录包含输入文件；result 文件包含结果文件，以及标准答案
- sina 目录下是基于 sina 语料库的输入法 python 源代码
- spider 目录下是爬虫脚本，用语爬取知乎的中文文本资料
- src 目录下是预处理的一些数据文件，在输入法运行时需要用到
- zhihu 目录下是基于 zhihu 语料库的输入法 python 源代码

最后，再次感谢马老师负责的讲解，以及助教的指导。感谢！

2021.4.7