拼音输入法 实验报告

2017011568 计76 张翔

概况信息

程序运行方法

安装依赖库（项目根目录下）

```
pip install -r requirements.txt
```

训练好的数据在trained目录下。bin里面带有Windows 64位Python 3.7的预编译包，如果系统相同可以直接使用，否则应在src目录下执行以下命令

```
python setup.py build_ext --inplace
```

然后执行主程序（bin目录或src目录下）即可，**默认打开模式是交互式shell**，可以输入拼音转换成文字

```
python main.py -n 3 ##默认为三元模型，改成2以使用二元模型
```

如果需要文件输入输出，则如下操作

```
python main.py -n 3 --file -i [input file] -o [output file]
```

目录结构

- bin为编译好的二进制文件，目前只编译了Windows 64位Python 3.7的程序包
- data为测试样例数据
- src为源代码，evaluate.py用于测试输入法性能，config.py为配置文件，setup.py用于编译pyx文件
- trained为训练得到的字频表，character.txt与pinyin_characters.txt为汉字和拼音汉字对应表，freq_gramX为一元组、二元组、三元组的频率，polyphone.freq为多音字频率，这些文件缺一不可，否则无法运行。

主要算法

拼音输入法要解决的问题

用户输入为拼音s，希望得到的输出为汉字w，则需要求$\arg\max P(w|s)$，该项概率根据Bayes公式可以表示成

$$\frac{P(s|w)P(w)}{P(s)}=\frac{P(s_1,s_2,\cdots,s_n|w_1,w_2,\cdots,w_n)P(w_1,w_2,\cdots,w_n)}{P(s)}$$

分母部分为常数，而分子部分可以用一阶隐马尔可夫模型简化，从而只需要求

$$\arg\max\prod_{i=1}^{N}P(w_i|w_{i-1})P(y_i|w_i)=\arg\max\prod_{i=1}^{N}P(w_i|w_{i-1})CF(w_i)$$

其中$CF(w_i)$是识别信度，含义为某字对应某拼音的概率，如果是非多音字，该项为1，否则该项和多音字的某读音y的频率呈正相关关系。开始时为了简化，可以将该项取为1。

概率项的来源

上式的$P(w_i|w_{i-1})$可以通过频率估计概率的思想得到，具体方法是统计语料中所有二元组出现的频率，然后对所有首字相同的二元组进行归一化，即可得到该项条件概率。计字符集为W，则

$$P(w_y|w_x)=\frac{\#w_xw_y}{\sum_{w_z\in W}\#w_xw_z}$$

求解最大概率的方法

将每个拼音对应的字视为有向图中的节点，w_{ik}到$w_{(i+1)j}$之间的有向边的权值为$P(w_{(i+1)j}|w_{ik})$，而图中每个节点w_i均有自己的权值，实际上就是$\prod_{k=1}^{i}P(w_i|w_{i-1})$。将所有概率取负对数后，问题转换成求该图中起点到终点的最短路。该图的节点根据拼音可以分层（某拼音对应的所有字均在同一层），有向边仅在相邻层出现，有这种特性的网络的最短路径可以通过Viterbi算法实现。算法思路为，设每层的节点均能存储起始位置到它的最短路径，对于后一层的节点，只需要找它的所有前驱中到它的总路径长度最短的即可。算法的时间和空间复杂度均为$O(LN^2)$，L为网络层数，N为网络每层的节点数。

具体实现步骤

- 读入二级字符表，根据读入顺序为汉字编号，共约6700字；读入拼音，建立拼音与汉字编号的联系
- 遍历语料，统计单字频和二元组的频率，利用scipy存储为稀疏矩阵和向量
- 取负对数前使用laplace平滑，避免频率为0的二元组影响计算
- 预测时使用Viterbi算法进行最大概率的查找

效果展示

Good Case

- xi tong dui yu ren men lai shuo bing bu shi yi ge mo sheng de ming ci

  系统对于人们来说并不是一个陌生的名词
- mei jun fang cheng bu cheng ren zhong guo dong hai fang kong shi bie qu

  美军方称不承认中国东海防空识别区
- shen du shen jing wang luo dui ji suan zi yuan de xiao hao hen da

  深度神经网络对计算资源的消耗很大
- gong qi jun de man hua zuo pin kan cheng jing dian

  宫崎骏的漫画作品堪称经典

Bad Case

- qia si na duo lian hua bu sheng liang feng de jiao xiu

  卡斯那朵莲花不乘凉风的校宿

- ta yang le yi zhi qing wa dang chong wu

  他养了一致青瓦当宠物

- wen ming jiao liu hu jian tui dong gou jian ren lei ming yun gong tong ti

  文明交流和践推动构建人类命运共同体

- zai dong tian diao jin bei da wei ming hu shi zen yang de ti yan

  在冬天掉进北大为名护士怎样的体验

以上Bad Case选取的是一些反映了典型问题的错误，分析如下

问题分析

- 语料库的范围较狭窄，只用新浪新闻作为语料库，存在词频不够平衡的问题，如“青瓦”“校宿”出现频率过高
- 二元模型的固有缺陷——目光短浅，如“掉进北大未名湖是怎样的体验”，只因为“护士”这个词频率比较高而选用，未考虑更前面的字
- 多音字识别问题，如“娇羞”识别成“校宿”，而“校”读“jiao”的频率略低于“xiao”

改进方法

引入多音字

使用pypinyin模块配合jieba分词工具，对语料中的多音字读音进行统计，将$CF(w_i)$替换成实际的$P(s_i|w_i)$，即该字读相应拼音的概率。

增加语料

爬取知乎热门话题回答约100MB，与原有新浪新闻语料混合使用。

采用基于字的三元模型

将计算模型变为

$$\arg\max\prod_{i=2}^{N}P(w_i|w_{i-1}w_{i-2})P(y_i|w_i)=\arg\max\prod_{i=2}^{N}P(w_i|w_{i-1}w_{i-2})CF(w_i)$$

即每个字和它前两个字均有关系。此时Viterbi算法应稍作改动，每一层的节点需要查找它的前两层节点中的最佳路径，这样时间复杂度变为$O(LN^3)$。

改进结果

准确率测试

随机从最近的新闻门户网站下爬取一些新闻，总共约1000句话，用pypinyin转换成拼音后进行测试，并用编辑距离（使用python-Levenshtein包进行计算）表征字准确度与句准确度

模型	字准确率	句准确率
裸的二元模型	87.06%	39.35%
裸的三元模型	92.43%	51.40%
二元+多音字	87.97%	41.55%
三元+多音字	92.88%	52.78%
二元+多音字+扩大语料	88.32%	42.98%
三元+多音字+扩大语料	93.03%	53.58%

可以看出，扩大语料和引入多音字能够小幅提高准确率，而从二元模型换为三元模型能较大程度上提高准确率，尤其是句准确率，这和三元模型对相邻字之间的约束加强有着密切关系。

一些样例

- you ke yu yuan tan shang ying hei ting che fei fa shou fei
    - 三元+多音字+语料：游客玉渊潭赏樱黑停车非法收费（正确）
    - 二元+多音字+语料：游客**余元摊上映**黑停车非法收费（错误）
    - 裸的三元：游客玉渊潭赏樱黑停车非法收费（正确）
    - 裸的二元：游客**余元摊上映**黑停车非法收费（错误）
- tuan dai wang she fei fa xi shou cun kuan bei zhen cha
    - 三元+多音字+语料：团贷网涉非法吸收存款被侦查（正确）
    - 二元+多音字+语料：**团带往**涉非法吸收存款被侦查（错误）
    - 裸的三元：团贷网涉非法吸收存款被侦查（正确）
    - 裸的二元：**团大网**设非法吸收存款被侦查（错误）
- wen ming jiao liu hu jian tui dong gou jian ren lei ming yun gong tong ti
    - 三元+多音字+语料：文明交流互鉴推动构建人类命运共同体（正确）
    - 二元+多音字+语料：文明交流**和践**推动构建人类命运共同体（错误）
    - 裸的三元：文明交流互鉴推动构建人类命运共同体（正确）
    - 裸的二元：文明交流**和践**推动构建人类命运共同体（错误）
- ta yang le yi zhi qing wa dang chong wu（忽略主语的性别属性 他/她均可）
    - 三元+多音字+语料：她养了一只青蛙当宠物（正确）
    - 二元+多音字+语料：他养了**一致青瓦**当宠物（错误）
    - 裸的三元：她养了一只青蛙当宠物（正确）
    - 裸的二元：他养了**一致青瓦**当宠物（错误）
- qia si na duo lian hua bu sheng liang feng de jiao xiu
    - 三元+多音字+语料：恰似那朵莲花不胜凉风的娇羞（正确）
    - 二元+多音字+语料：恰似那朵莲花不胜凉风的**角修**（错误）
    - 裸的三元：恰似那朵莲花**步乘**凉风的**脚臭**（错误）
    - 裸的二元：**卡斯**那朵莲花不乘凉风的**校宿**（错误）
- jiu shi yong ya zui bi ba zhe ge shui nong dao li mian
    - 三元+多音字+语料：就是用鸭嘴笔把这个水弄到里面（正确）
    - 二元+多音字+语料：就是用**牙最必**把这个水弄到里面（错误）
    - 裸的三元：就是用**牙咀笔**把这个**说**弄到里面（错误）
    - 裸的二元：就是用**鸭嘴必**把这个水弄到里面（错误）
- gong qi jun de man hua zuo pin kan cheng jing dian

    - 三元+多音字+语料：宫崎骏的漫画作品堪称经典（正确）
    - 二元+多音字+语料：宫崎骏的漫画作品堪称经典（正确）
    - 裸的三元：宫崎骏的漫画作品堪称经典（正确）
    - 裸的二元：红旗军的漫画作品堪称经典（正确）
- zai dong tian diao jin bei da wei ming hu shi zen yang de ti yan
    - 三元+多音字+语料：在冬天掉进北大未名湖是怎样的体验（正确）
    - 二元+多音字+语料：在冬天掉进北大**为名护士**怎样的体验（错误）
    - 裸的三元：在冬天掉进北大未名湖是怎样的体验（正确）
    - 裸的二元：在冬天掉进北大**为名护士**怎样的体验（错误）
- mo dao sang yu wan wei xia shang man tian
    - 三元+多音字+语料：莫道桑榆晚为霞尚满天（正确）
    - 二元+多音字+语料：**磨刀颡鱼丸为下上**满天（错误）
    - 裸的三元：莫道桑榆晚为霞尚满天（正确）
    - 裸的二元：**没到桑余万尾虾上**满天（错误）
- lu man man qi xiu yuan xi wu jiang shang xia er qiu suo
    - 三元+多音字+语料：路漫漫其修远兮吾将上下而求索（正确）
    - 二元+多音字+语料：路漫漫**奇秀员席吴江**上下而**求所**（错误）
    - 裸的三元：路漫漫其修远兮吾将上下而求索（正确）
    - 裸的二元：**绿满满期修院系误将**上下而**求所**（错误）
- qing hua da xue ji suan ji xi deng jun hui jiao shou
    - 三元+多音字+语料：清华大学计算机系邓俊辉教授（正确）
    - 二元+多音字+语料：清华大学计算机系**等均会**教授（错误）
    - 裸的三元：清华大学计算机系邓**骏**辉教授（错误）
    - 裸的二元：清华大学计算机系**等均会**教授（错误）
- xiang xue hao wei ji fen shi bu ke neng de
    - 三元+多音字+语料：想学好微积分是不可能的（正确）
    - 二元+多音字+语料：**向学号卫计分**是不可能的（错误）
    - 裸的三元：**降雪号**微积分是不可能的（错误）
    - 裸的二元：**降雪号卫计分**是不可能的（错误）

从以上样例可以看出，三元语法配合多音字处理以及增加新鲜的语料，可以大幅提高准确度，特别对于刁钻的样例，如古诗词、专有名词等，可以较好地识别，而二元模型相比就较为逊色了。

调参

二元模型

针对二元模型，有时因为某些二元组从没出现，可以采用$\lambda P_1 + (1-\lambda)P_2$ 的方式进行加权，P_1为单字概率，P_2为二元组对应的条件概率。以下为部分参数对应的结果，从表中可以看出，当λ较小时，不影响句准确率（全局正确），并可以适度提升字准确率（局部正确）

λ	Word Accuracy	Sentence Accuracy
0	88.321369%	42.979943%
1e-2	88.397294%	42.884432%
1e-3	88.342076%	42.979943%
1e-4	88.321369%	42.979943%
0.1	88.038377%	42.120344%

λ	Word Accuracy	Sentence Accuracy
0.15	87.845113%	41.451767%

经过调参，$\lambda = 10^{-3}$时效果较好，字准确率88.34%，句准确率42.98%。相较于前面二元模型的最佳状态，字准确率略有提升的同时保持了句准确率，而$\lambda = 10^{-2}$虽然字准确率略有提高，但牺牲了部分句准确率，从而不采用这个参数。

三元模型

对于三元模型，存在三元组从未出现过，退化为使用二元模型的情况。对于此种情况，定义参数$\mu(\mu \geq 1)$，作为惩罚值乘上二元组对应的概率。以下是部分测例

μ	Word Accuracy	Sentence Accuracy
1	93.077029%	58.166189%
1.1	93.953617%	60.362942%
1.15	94.022639%	60.649475%
1.2	93.932910%	60.267431%
1.3	93.905301%	60.267431%
1.4	93.843181%	59.789876%
1.01	93.235781%	58.643744%
1.03	93.429045%	59.216810%
1.001	93.097736%	58.166189%

从表中可以看出，当μ大约在1.2-1.3之间时可以获得较好的字句准确率，但此时会导致前面一些较偏的测例出现错误（如“冬天掉进北大未名湖是怎样的体验”）。综合考虑准确度提升的同时加上特殊测例的正确性，选取$\mu = 1.03$，此时相比未调参时字、句准确率均有不少提升，并且前述测例均能正常输出。

总结

通过对不同方法结果的比较，以及对一些错误识别情况的分析，我总结出以下内容：

- 三元模型对于常见的短语等有较好的识别率，比二元模型更加实用，但使用时需要注意性能问题，一种策略是每次对某层的某个节点求权值时，可以取前两层节点前NUM个较好的，避免$O(LN^3)$级别的复杂度
- 语料库应足够大，且最好比较平衡，能够涵盖各个方面
- 通过对多音字读音频率的统计，可以对识别信度进行调整，从而获得更为合理的结果，避免出现一些多音字的奇怪用法（“校宿”对比“娇羞”）
- 适当调整概率的权重分配（前面提到的λ和μ参数），可以稍微提高一些准确率
- 用户输入拼音时，如果输入含有较为完整且合理的意群的句子，正确率会比较高，如“特总统连任好不好啊”对比“董先生联人（连任）好不好啊”，前者在语义上是正规的，而后者则出现了特殊搭配“先生连任”，对于识别率影响较大
- 如果想进一步提升准确率，可以考虑以下方法：
    - 使用基于词的二元和三元模型，充分利用分词信息
    - 结合句法和词性的信息，对内容进行综合推测
    - 考虑不同字词在句中出现的位置（如主语一般在开头出现），由此增加约束
    - 使用新的方法，如seq2seq等