WEB INFO LAB1实验报告

PB19000352 易元昆 PB19000186 王晨晗

bool检索

> 对于经过预处理的文档集合 $D={D_1,D_2,...,D_N}$，根据倒排索引算法建立倒排索引表，并以合适的方式存储生成的倒排索引文件。

1.首先对原文档进行预处理，经过变小写、去除停用词、词干化（对应库函数为nltk.stem.SnowballStemmer）之后得到名为{文件夹名称}.{原文件名称}的中间文档集合。这里调用的库函数有nltk.corpus里面的停用词(stopwords), `nltk.tokenize`中`WhitespaceTokenizer`的分词

2.对经过预处理的文档集合内容进行统计，按照倒排索引算法建立倒排索引表，其中表中还添加了词项在文档中出现的次数作为排序的判断标准。即，对于未出现的词项，建立新的键值对，对于出现的词项，若文档ID已存在，计数+1，未存在，则初始化{文档ID:1}。

3.得到的倒排索引表存入./IndexTable.txt文件。

> 对于给定的 bool 查询 （的书写规则以上课内容为准），根据你生成的倒排索引表，返回符合查询规则的文档集合 。

1.利用ply库，建立lexer和parser，实现对于输入的bool检索表达式的分析。这里的具体语法学习来自于编译原理实验和CSDN。其中对于输入的停用词会返回错误提示，处理时将停用词对应文档集合视作空集合。

2.对于and、or、not，为了区分与正常查找词区别，输入后用replace函数将其变为大写，方便分析。在处理过程中，AND、OR会分别通过简单的方式改变相关文档排序优先级，具体如图。

```
for key in p[1]:
    if p[3].get(key) != None:
        p[0][key]=p[1][key]+p[3][key]
    else:
        p[0][key]=p[1][key]+0
for key in p[3]:
    if p[0].get(key) == None:
        p[0][key]=p[3][key]
```

```
for key in p[1]:
    #print(key,'!!!')
    if p[3].get(key) != None:
        p[0][key]=p[1][key]+p[3][key]
```

3.最终通过全局变量result将查找到的相关文档路径存入./output/bool_search_result文件夹，并将排序结果靠前的文档输出在命令行，个数由变量search_num指定。

```
Congratulations!  Index Table 生成成功
What do you wanna search for?
Please enter your imformation in bool mode:company and percent and income or march and not statemen
t or (loss and tax)
2018_02/news_0003040.json
2018_02/news_0028255.json
2018_02/news_0060977.json
2018_02/news_0004149.json
2018_05/news_0004746.json
2018_02/news_0009834.json
2018_05/news_0011166.json
2018_02/news_0016497.json
2018_05/news_0021516.json
2018_04/news_0056133.json
```

bool_search_result.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

./US_Financial_News_Articles/2018_01/blogs_0000603.json
./US_Financial_News_Articles/2018_01/blogs_0006352.json
./US_Financial_News_Articles/2018_01/blogs_0006465.json
./US_Financial_News_Articles/2018_01/blogs_0009144.json
./US_Financial_News_Articles/2018_01/blogs_0009253.json
./US_Financial_News_Articles/2018_01/blogs_0009761.json
./US_Financial_News_Articles/2018_01/blogs_0011060.json
./US_Financial_News_Articles/2018_01/blogs_0011666.json
./US_Financial_News_Articles/2018_01/blogs_0012693.json
./US_Financial_News_Articles/2018_01/blogs_0013064.json
./US_Financial_News_Articles/2018_01/blogs_0013208.json
./US_Financial_News_Articles/2018_01/blogs_0015290.json
./US_Financial_News_Articles/2018_01/blogs_0015963.json
./US_Financial_News_Articles/2018_01/blogs_0015991.json
./US_Financial_News_Articles/2018_01/blogs_0017763.json
./US_Financial_News_Articles/2018_01/blogs_0017839.json
./US_Financial_News_Articles/2018_01/blogs_0018459.json
./US_Financial_News_Articles/2018_01/blogs_0018937.json
./US_Financial_News_Articles/2018_01/blogs_0020273.json

> 一些要点

1.词干化的库函数对于一些词无法将它恢复至正确词根，比如`created`词干化后会变为`creat`而非`create`，但是该库函数可以令`create`同样变为`creat`。因此对于输入的查询项，我们对其进行词干化后再去倒排索引表中查找，这样同样可以达到想要的效果

2.这里bool检索的相关性只利用了文档中词项出现次数作为参数，较为简略，主要考虑到bool检索本身的限制以及题目中未要求排序，就没有进一步使用更准确的消除各种影响因素的参数。

整体运行时间约为25min。

TF-IDF

通过semantic_search.py实现

1. 检索所有文件夹，读取其中的text项目。调用库函数，进行分词、去停用词，词频统计，计算tf-idf。

其中通过遍历所有json文件读取其中的`text`相关内容，读取的内容以`list`格式存储于`data`里面，文件路径存储于`FileList`中，同时将文件路径输出到`output`文件目录下`name.txt`中，**之后计算中通过数字代指文件，减少数据读取，方便计算，减少运行时间**

调用库`scikit-learn`中的文本处理模块进行处理，其中`vectorizer`用于对原始数据进行处理包括标准化和去停用词，`TfidfTransformer`用于生成tf-idf矩阵。

生成完毕后存储于`output`文件目录中的`tfidf_matrix.txt`文件中(分词后的词排序用`feature_name`存储，生成的tf-idf矩阵用`tf-idf`存储，存储类型为`scipy.sparse.csr.csr_matrix`稀疏矩阵)

存储是通过python库`pickle`实现，`pickle`库可以实现python`list、dict`等数据类型的存储和读取，方便后续处理数据

2. 输入相应查询词，进行计算

读入先前处理过的数据， 读入输入的查询词集合。对输入的查询词进行处理(仍然使用`vectorizer = CountVectorizer(vocabulary=feature_name)`此处附带参数为按照步骤2中生成的分词表生成稀疏矩阵)，然后计算查询词的tf-idf向量。

通过`cosine_similarity()`函数计算查询词集合向量与原来生成的tf-idf向量的VSM，并返回最相似的10个文件的集合，然后输出

读取数据需要大致30分钟(占主要时间),建立矩阵大致需要6分钟(运行环境见readme),查询输入词需要8分钟

3. 输出图片

实现了选做4

在输出查询所得的10个词的同时，通过返回的文件路径重新打开对应的json文件，从中提取`main_image`字段，返回图片URL

测试结果

时间统计：

其中读入文件部分用时35min，生成词频统计矩阵花费4分钟，计算生成tf-idf稀疏矩阵1分钟，计算查询词向量与其余向量之间的余弦值花费8-10min(取决于查询词数量)

```
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_05\news_0063140.json
请输入查询的词集合
company market percent income quarter financial share cash march statement business billion tax loss president sale revenue asset share source report earnings captial
F:\ProgramData\Anaconda3\envs\src\lib\site-packages\sklearn\feature_extraction\text.py:1208: UserWarning: Upper case characters found in vocabulary while 'lowercase' is T
  warnings.warn(
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0001620.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/01/104923013-2ED1-RDN-KingsleyJones-010118.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0001305.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/03/104926004-4ED1-SOTS-PisaniETF-010318.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0001192.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/01/104923111-2ED1-SSA-MartinLakos-010118.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0001079.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/02/104924244-3ED1-REQ-Segment1FN-010218.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0001076.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/01/104923039-2ED1-ASB-JonathanBarratt-010118.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0000841.json
Graph:  https://s4.reutersmedia.net/resources_v2/images/rcom-default.png
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0000701.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2013/12/10/101260233-smoking.1910x1000.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0000609.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/01/02/104924258-3ED1-REQ-Segment3FN-010218.600x400.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0000602.json
Graph:  https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2017/11/08/104828847-GettyImages-471172691.1910x1000.jpg
F:\学习\作业\大三\web\web-info\lab1\exp1\dataset\US_Financial_News_Articles_3\2018_01\blogs_0000083.json
Graph:  https://s4.reutersmedia.net/resources_v2/images/rcom-default.png
```

此处输入了查词表中2/3的单词，输出结果如上所示，运行时间为9 min 23 s