Oliver

学　　院：信息科学与工程学院

2021 年 12 月 07 日

目 录

一、项目概述

1.1 项目简介

文本信息检索系统是针对文档内容设计的,且是以用户检索文本内容需求为出发点的。此文本信息检索系统可以向用户提供文本内容检索服务，实现文本检索功能，提高了用户操作效率，使文本信息检索更加规范、方便、快捷。

1.2 项目功能

本文本信息检索系统根据提供的本文内容进行预处理并建立词项倒排索引，可通过用户输入单个词项进行词项的检索，也可通过用户输入布尔检索式进行两个词项以上的的布尔检索，最后对检索结果进行排序。

1.3 项目模块介绍

文本预处理：

对单个文本进行读取，首先对文本进行去除标点符号操作，其次去除数字，对文本按照空格进行分割词项，最后去除词项中的停用词。

倒排索引：

将分割后的词项，使用字典数据结构，按照词项字母顺序建立倒排索引。

检索：

分析用户输入内容，若为单个单词，则进行单个词项检索；

若用户输入布尔检索式，则按照布尔检索式中运算符输入顺序进行多个词项的布尔检索。

结果排序：

对检索结果根据词项的 tf-idf 降序进行排序

二、可行性研究

2.1 技术可行性分析

使用到的编程语言有 python，使用到的编译软件有 pycharm，使用这些技术就可以实现这个系统。

2.2 系统流程图

如图 1 所示，

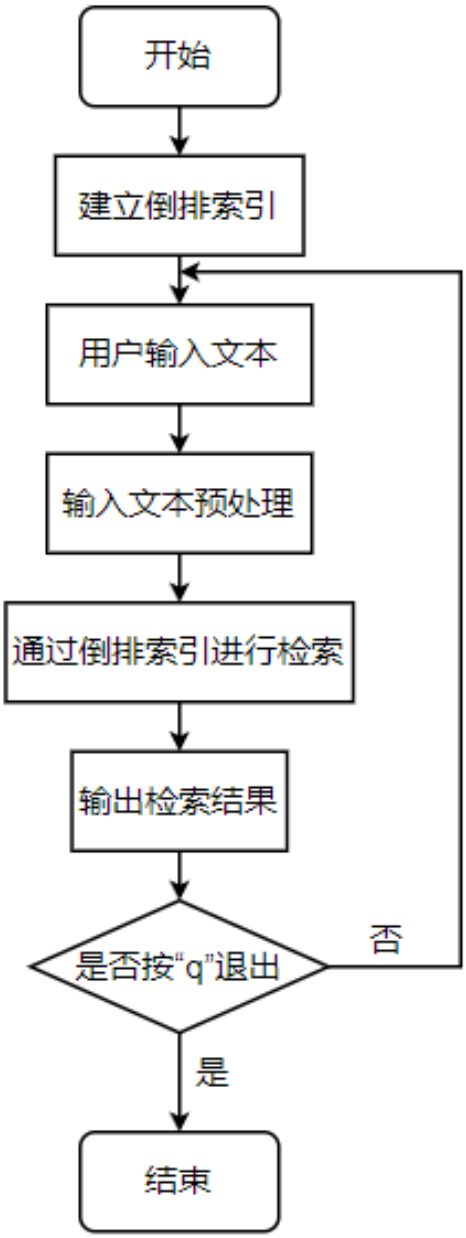

图 1 系统流程图

三、功能介绍

3.1 预处理

对于检索文档内容：

通过相对路径逐个读取文件内容，并将文件内容存入列表中，代码如图 2 所示。

```
for root, dirs, files in os.walk(dataset_path):
    for name in files:
        text_files.append(os.path.join(root, name))#每个文件的相对路径，从hyatt-k开始
```

图 2 读取文件内容

再将文件内容通过 base 类中的预处理函数进行文本预处理，包括全部转换小写，去除标点符号，去除数字，函数如图 3，4，5 所示。

```
# 去除标点符号
def punctuation_remove(self):
```

图 3 去除标点符号函数

```
# 去除文本中的数字
def num_remove(self):
```

图 4 去除数字函数

```
# 小写
def case_folding(self):
```

图 5 转换小写函数

处理过后，通过空格划分词项得到全部词项。此时，读入停用词表，去除全部词项中的部分停用词，停用词表如表 1 所示。

a	all	an	and	any
are	as	be	been	but
by	few	for	have	he
her	here	him	his	how
i	in	is	it	its

many	me	my	none	of
on	or	our	she	some
the	their	them	there	they
that	this	us	was	what
when	where	which	who	why
will	with	you	your	

表 1 停用词表

对于输入内容：

首先对用户输入内容利用文本处理器调用 base 中的函数进行预处理，包括去除标点符号，去除数字操作，调用函数如图 6,7,8 所示，其次按照空格对输入内容进行分割，提取词项以及布尔操作符。

```
def _preprocess(self, q) -> str:
    """
    用于在处理之前预处理检索内容
    """
```

图 6 输入内容文本处理器

```
# 去除标点符号
def punctuation_remove(self):
```

图 7 去除标点符号函数

```
# 去除文本中的数字
def num_remove(self):
```

图 8 去除数字函数

3.2 倒排索引

倒排索引：由单词词典和倒排文件组成的，实现单词—文档矩阵的一种具体存储形式，是单词到文档映射关系的最佳实现方式（可以根据单词快速获取包含该单词的文档列表）。

在第一次运行程序时，将按照文件内容进行建立倒排索引

1、获取每个文档的单词表，即通过上述预处理后的每个文档的单词表，将文档中的单

词做为 key，出现的文档编号做为内容。

2、合并所有文档的单词表

3、将相同词项的文档编号进项合并，由于很多单词同时出现在不同的文档，所以这个列表的词典项有重复，所以将文档编号插入对应词项列表中。

4、最后按照词项字母顺序进行排序

将建立好的倒排索引表按照二进制形式存入 dictionary.pkl 文件中,再次运行程序时将不再重新建立倒排索引表而是直接通过文件读入倒排索引表。

```
for doc in self.textp.docs:
    #得到倒排记录表
    tokens = self.textp.tokenize(doc.content)
    tokens = list(set(tokens) - set(self.textp.stopwords))#去停用词
    #添加词项
    for token in tokens:
        self.add(token, doc.id)
```

图 9 建立倒排索引部分代码

```
abercrom: ['hyatt-k\\tw\\6']
aberdyaolcom: ['hyatt-k\\deleted_items\\131', 'hyatt-k\\deleted_items\\396', 'hyatt-k\\deleted_
abide: ['hyatt-k\\deleted_items\\205']
abil: ['hyatt-k\\deleted_items\\142', 'hyatt-k\\projects\\15', 'hyatt-k\\sent_items\\48']
abilene: ['hyatt-k\\deleted_items\\468', 'hyatt-k\\personal\\cars\\10']
abilities: ['hyatt-k\\deleted_items\\20', 'hyatt-k\\projects\\tsunami\\2']
ability: ['hyatt-k\\corp_memos\\21', 'hyatt-k\\deleted_items\\189', 'hyatt-k\\deleted_items\\21'
abix: ['hyatt-k\\deleted_items\\350', 'hyatt-k\\deleted_items\\383']
able: ['hyatt-k\\corp_memos\\15', 'hyatt-k\\corp_memos\\17', 'hyatt-k\\corp_memos\\22', 'hyatt-
ablecom: ['hyatt-k\\deleted_items\\489']
abloy: ['hyatt-k\\deleted_items\\522', 'hyatt-k\\deleted_items\\527', 'hyatt-k\\deleted_items\\
abn: ['hyatt-k\\deleted_items\\553', 'hyatt-k\\inbox\\enron_news\\5']
abner: ['hyatt-k\\deleted_items\\458', 'hyatt-k\\deleted_items\\459', 'hyatt-k\\deleted_items\\
abo: ['hyatt-k\\projects\\mckinley\\5']
abolish: ['hyatt-k\\deleted_items\\637']
aboriginal: ['hyatt-k\\market_intel\\38', 'hyatt-k\\projects\\tsunami\\10', 'hyatt-k\\projects\'
aborted: ['hyatt-k\\projects\\tsunami\\3']
abortion: ['hyatt-k\\deleted_items\\229']
about: ['hyatt-k\\corp_memos\\11', 'hyatt-k\\corp_memos\\13', 'hyatt-k\\corp_memos\\14', 'hyatt
aboutabbr: ['hyatt-k\\deleted_items\\612']
aboutcom: ['hyatt-k\\deleted_items\\3']
aboutcoms: ['hyatt-k\\deleted_items\\3']
aboutop: ['hyatt-k\\deleted_items\\428']
above: ['hyatt-k\\corp_memos\\13', 'hyatt-k\\corp_memos\\21', 'hyatt-k\\corp_memos\\4', 'hyatt-
```

图 10 部分倒排索引表

3.3 布尔检索

布尔检索以布尔表达式的形式表示（结合逻辑运算符 AND，OR，NOT）的任何查询。

用户输入布尔表达式，通过预处理进行词项分隔，进行布尔检索

1、在词典中定位左词项

2、返回其倒排记录，作为左列表

3、在词典中定位右词项

4、返回其倒排记录，作为右列表

5、对左右两个倒排记录表进行求取交集、并集、差集。

对单个词项进行检索

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：zone
检索结果：
hyatt-k\deleted_items\229
hyatt-k\deleted_items\511
hyatt-k\deleted_items\514
hyatt-k\deleted_items\516
hyatt-k\deleted_items\528
hyatt-k\deleted_items\556
hyatt-k\deleted_items\606
hyatt-k\deleted_items\649
hyatt-k\market_intel\11
hyatt-k\sent_items\247
```

图 11 单个词项检索

1、使用 AND 操作符进行检索

w1 AND w2 AND w3...AND wn

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able AND write
检索结果：
hyatt-k\deleted_items\384
hyatt-k\power_conf\1
hyatt-k\sent_items\286
```

图 12 两个词项 AND 操作

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able AND work AND apply
检索结果：
hyatt-k\inbox\39
```

图 13 多个词项 AND 操作

2、使用 OR 操作符进行检索

w1 OR w2 OR w3...OR wn

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able OR write
检索结果：
hyatt-k\corp_memos\15
hyatt-k\corp_memos\17
hyatt-k\corp_memos\22
hyatt-k\corp_memos\23
hyatt-k\corp_memos\24
hyatt-k\corp_memos\26
hyatt-k\corp_memos\4
hyatt-k\corp_memos\42
```

图 14 两个词项 OR 操作

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able OR write OR apply
检索结果：
hyatt-k\corp_memos\15
hyatt-k\corp_memos\17
hyatt-k\corp_memos\22
hyatt-k\corp_memos\23
hyatt-k\corp_memos\24
hyatt-k\corp_memos\26
hyatt-k\corp_memos\4
hyatt-k\corp_memos\42
hyatt-k\deleted_items\116
```

图 15 多个词项 OR 操作

3、使用 NOT 操作符进行检索

w1 NOT w2 NOT w3...NOT wn

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able NOT write
检索结果：
hyatt-k\corp_memos\15
hyatt-k\corp_memos\17
hyatt-k\corp_memos\22
hyatt-k\corp_memos\23
hyatt-k\corp_memos\24
hyatt-k\corp_memos\26
hyatt-k\corp_memos\4
hyatt-k\corp_memos\42
```

图 16 两个词项 NOT 操作

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able NOT write NOT apply
检索结果：
hyatt-k\corp_memos\15
hyatt-k\corp_memos\17
hyatt-k\corp_memos\22
hyatt-k\corp_memos\23
hyatt-k\corp_memos\24
hyatt-k\corp_memos\26
```

图 17 多个词项 NOT 操作

4、使用复合布尔操作符进行检索

w1 AND w2...AND wn OR wn+1 OR wn+2 ...OR wn+m NOT wn+m NOT wn+m+1... NOT wn+m+s

```
请输入要查询的内容（布尔检索操作符：AND，OR，NOT，输入q退出系统）：able AND write OR apply NOT zone
检索结果：
hyatt-k\deleted_items\116
hyatt-k\deleted_items\155
hyatt-k\deleted_items\384
hyatt-k\deleted_items\388
hyatt-k\deleted_items\431
hyatt-k\deleted_items\509
hyatt-k\deleted_items\85
hyatt-k\deleted_items\so_trails\2
hyatt-k\deleted_items\so_trails\5
hyatt-k\deleted_items\so_trails\8
```

图 18 复合布尔查询

3,4 结果排序

影响一个词项在一篇文档中的重要性主要有两个因素：

Term Frequency (tf)：即此 Term 在此文档中出现了多少次。tf 越大说明越重要。

Document Frequency (df)：即有多少文档包含次 Term。df 越大说明越不重要。

TF-IDF 是一种用于信息检索与数据挖掘的常用加权技术。TF 是词频，IDF 是逆文本频率指数。TF-IDF 的主要思想是：如果某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

用户输入的检索词在文件中出现的频率越高，出现的位置越重要，那么就认为该文件与此检索词的相关度越高，其在搜索结果出现的位置越靠前。

```
hyatt-k\corp_memos\14
hyatt-k\corp_memos\42
hyatt-k\deleted_items\458
hyatt-k\deleted_items\575
hyatt-k\deleted_items\615
hyatt-k\projects\27
hyatt-k\projects\sun_devil\104
hyatt-k\projects\sun_devil\36
hyatt-k\projects\sun_devil\37
hyatt-k\projects\tsunami\12
```

图 19 结果排序

四、总结

本次实验完成了倒排索引，单个词项的检索，布尔查询的要求，复杂的布尔查询也可以在基本的 AND、OR、NOT 逻辑基础实现上嵌套实现，最后通过用查询的单词在该文档中出现的个数/总数作为简单的排序检索比较简陋，结果需要进一步评估，本系统考虑不周，还需要进一步完善来满足更高级的应用需求。