

武汉大学国家网络安全学院

实验报告

课程名称 信息检索

专业年级 网络空间安全-21 级研一

姓 名 王泽鹏

学 号 2021202210073

协 作 者 无

实验学期 第一 学年 第一 学期

课堂时数 课外时数

填写时间 2021 年 10 月 27 日

实验概述
<p>【实验项目名称】：</p> <p>自定义命令行搜索引擎编写</p>
<p>【实验目的】：</p> <p>通过编写一个搜索引擎对信息检索的关键知识点进行实践、巩固与掌握。</p> <p>【实验环境】（使用的软件）：</p> <p>(1) 硬件环境：无</p> <p>(2) 操作系统环境：Ubuntu 1604（虚拟机）</p> <p>(3) 测试脚本编程语言：python</p> <p>(4) 被测系统编程语言：python</p> <p>(5) 网络环境：无</p> <p>(6) 其他环境：openjdk-8-jdk、ant、pylucene7.7.1（jdk 和 ant 为 pylucene 的依赖环境）</p> <p>【参考资料】：</p> <p>Lucene7.7.1 官方文档、CSDN 相关博客、Wiki 等</p>
实验内容
<p>【实验方案设计】：</p> <p>一、实验要求</p> <ol style="list-style-type: none"> 1. 编写一个自定义命令行搜索引擎来索引 37,600 多篇 TDT3 新闻文章 2. 使用 Lucene 建立索引 3. 自定义相关性评分函数 4. 通过控制台/屏幕上返回数据集文章中与查找信息相关性排名前 N 的文章信息，包括排名、分数、docID 和摘要片段。 5. 查询包括：①free text (order independent)；②Doubly quoted phrase queries mixed with free text queries 6. 具体查询内容包括： <ul style="list-style-type: none"> • Q1 - hurricane

- Q2 - mitch george
- Q3 - bill clinton israel
- Q4 - “newt gingrich” down
- Q5 - nba strike closed-door

7. 报告中包含每项查询的前 10 相关信息的结果截图

8. 额外项：stemming、upper/lower case characters、numbers、punctuations including hyperns

二、实验整体流程

0. pylucene 相关环境配置与安装

由于想使用 **python** 解决问题，遂尝试使用 **lucene** 的 **python** 版本 **pylucene**，但 **pylucene** 的环境配置及安装实在是处处是坑，在 **windows** 尝试安装却遇到无法解决且无处可查的问题后（如下图所示），转向 **linux** 版本，终于在虚拟机上的 **ubuntu1604** 环境下安装成功。

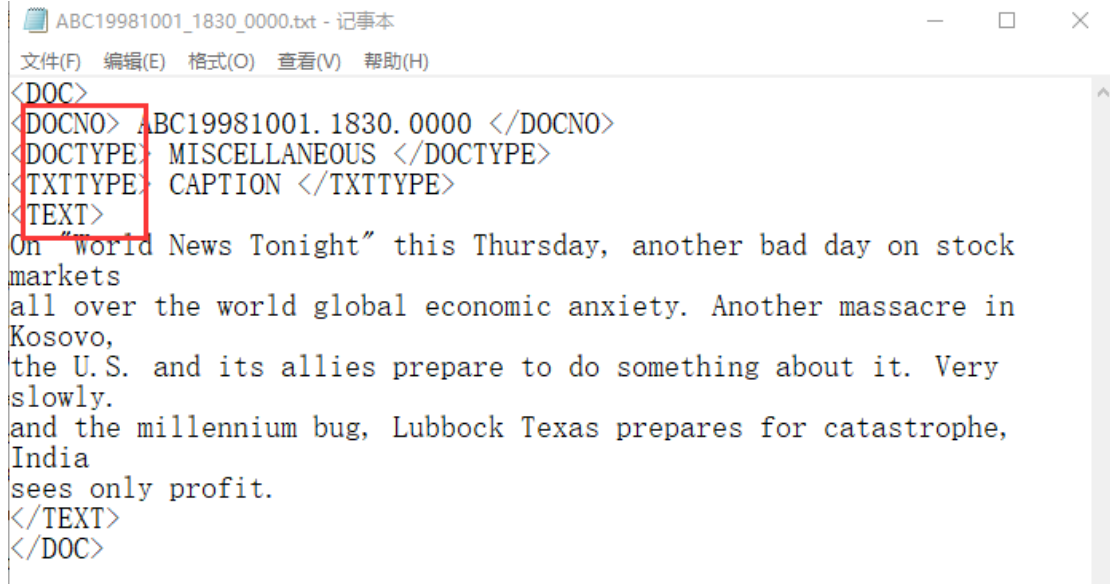
```
(search_engine) E:\>python
Python 3.6.2 |Continuum Analytics, Inc. | (default, Jul 20 2017, 12:30:02) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import jcc
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "E:\Anaconda3\envs\search_engine\lib\site-packages\jcc-3.5-py3.6-win-amd64.egg\jcc\_init_.py", line 31, in <module>
    import jcc._jcc3 as _jcc3
ImportError: DLL load failed: 找不到指定的模块。
>>> ^Z
```

```
wzp@wzp-virtual-machine: ~/pylucene-7.7.1
creating build/bdist.linux-x86_64/egg/EGG-INFO
copying lucene.egg-info/PKG-INFO -> build/bdist.linux-x86_64/egg/EGG-INFO
copying lucene.egg-info/SOURCES.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying lucene.egg-info/dependency_links.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying lucene.egg-info/not-zip-safe -> build/bdist.linux-x86_64/egg/EGG-INFO
copying lucene.egg-info/top_level.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
writing build/bdist.linux-x86_64/egg/EGG-INFO/native_libs.txt
creating dist
creating 'dist/lucene-7.7.1-py3.6-linux-x86_64.egg' and adding 'build/bdist.linux-x86_64/egg' to it
removing 'build/bdist.linux-x86_64/egg' (and everything under it)
Processing lucene-7.7.1-py3.6-linux-x86_64.egg
creating /home/wzp/anaconda3/envs/lucene/lib/python3.6/site-packages/lucene-7.7.1-py3.6-linux-x86_64.egg
Extracting lucene-7.7.1-py3.6-linux-x86_64.egg to /home/wzp/anaconda3/envs/lucene/lib/python3.6/site-packages
Adding lucene 7.7.1 to easy-install.pth file

Installed /home/wzp/anaconda3/envs/lucene/lib/python3.6/site-packages/lucene-7.7.1-py3.6-linux-x86_64.egg
Processing dependencies for lucene==7.7.1
Finished processing dependencies for lucene==7.7.1
(lucene) wzp@wzp-virtual-machine:~/pylucene-7.7.1$
```

1.数据集信息爬取处理

首先观察数据集中文章的具体结构，确定信息如何进行预处理，再用于索引建立存储。



```
ABC19981001_1830_0000.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<DOC>
<DOCNO> ABC19981001.1830.0000 </DOCNO>
<DOCTYPE> MISCELLANEOUS </DOCTYPE>
<TXTTYPE> CAPTION </TXTTYPE>
<TEXT>
On "World News Tonight" this Thursday, another bad day on stock
markets
all over the world global economic anxiety. Another massacre in
Kosovo,
the U.S. and its allies prepare to do something about it. Very
slowly.
and the millennium bug, Lubbock Texas prepares for catastrophe,
India
sees only profit.
</TEXT>
</DOC>
```

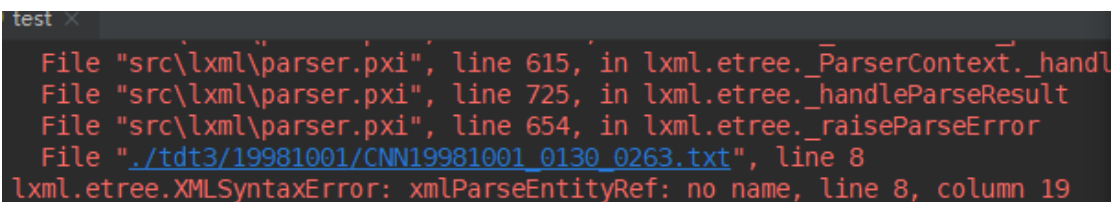
可以看到文档内容实际上是一个 **xml** 文件的形式，所以我们可以按照处理 **xml** 文件的方法将 **docno**、**doctype**、**txttype** 和 **text** 四个域的关键信息提取出来用于后续的存储。

这里我们使用 **lxml.etree** 模块进行四个域信息的提取，关键代码如下：

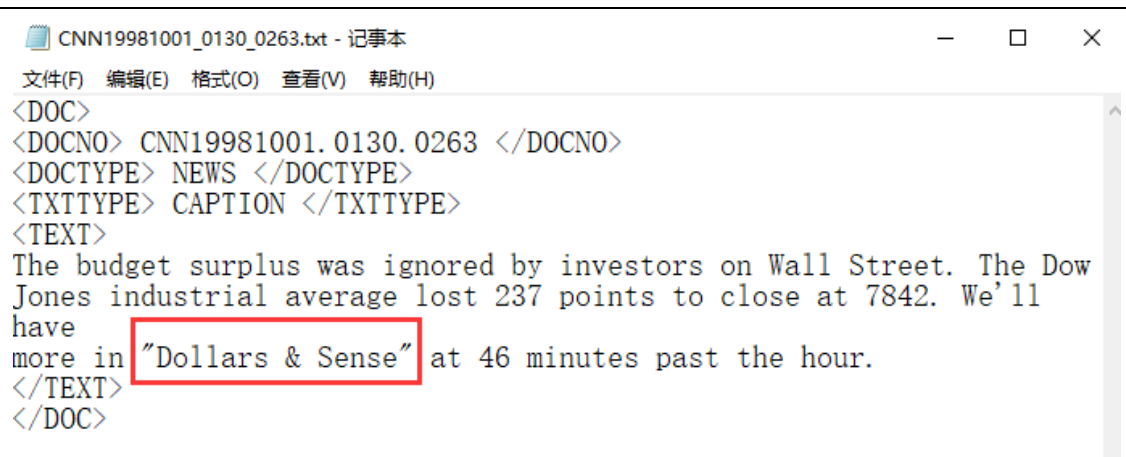
```
def xml_crawling(filename):
    xml = etree.parse(filename)
    root = xml.getroot() # 获取根节点
    doc_no = root.xpath('//DOCNO')[0].text.strip()
    doc_type = root.xpath('//DOCTYPE')[0].text.strip()
    txt_type = root.xpath('//TXTTYPE')[0].text.strip()
    # text = root.xpath('//TEXT')[0].text.replace('\n', '').replace('\r', '')
    text = root.xpath('//TEXT')[0].text # 为了print整洁没必要去除换行符
    # print(text)

    return [doc_no, doc_type, txt_type, text]
```

其中需要注意的是，文档中文本包含的特殊符号，如“&”会影响到 **xml** 的读取，如下图所示：

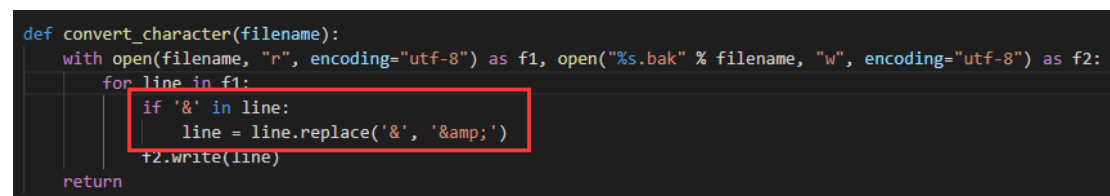


```
test x
File "src\lxml\parser.py", line 615, in lxml.etree._ParserContext._handle
File "src\lxml\parser.py", line 725, in lxml.etree._handleParseResult
File "src\lxml\parser.py", line 654, in lxml.etree._raiseParseError
File "./tdt3/19981001/CNN19981001_0130_0263.txt", line 8
lxml.etree.XMLSyntaxError: xmlParseEntityRef: no name, line 8, column 19
```



```
CNN19981001_0130_0263.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<DOC>
<DOCNO> CNN19981001.0130.0263 </DOCNO>
<DOCTYPE> NEWS </DOCTYPE>
<TXTTYPE> CAPTION </TXTTYPE>
<TEXT>
The budget surplus was ignored by investors on Wall Street. The Dow
Jones industrial average lost 237 points to close at 7842. We'll
have
more in "Dollars & Sense" at 46 minutes past the hour.
</TEXT>
</DOC>
```

故需要对特殊符号进行转义替换，经测试发现只有“&”会影响 xml 的读取，故只需处理该符号的转义即可：



```
def convert_character(filename):
    with open(filename, "r", encoding="utf-8") as f1, open("%s.bak" % filename, "w", encoding="utf-8") as f2:
        for line in f1:
            if '&' in line:
                line = line.replace('&', '&')
            f2.write(line)
    return
```

最后我们得到了包含全部文本文件的 **docno**、**doctype**、**txttype** 和 **text** 四个域的信息的列表，用于下一步建立索引存储。

2. 建立索引

建立索引的过程按照 **Lucene** 建立索引的标准过程操作即可。

首先实例化一个 **SimpleFSDirectory** 对象用于存储索引。

然后为这个 **Directory** 对象实例化一个 **IndexWriter** 对象，同时为其进行相应设置，包括设置分析器为 **StandardAnalyzer**，其可以根据空格和符号完成分词，还可以完成数字、字母等分析处理，满足了需求文档对数字、字母等处理的额外要求。另外在这里要对文档相似度函数进行设置，由于要自定义评分函数，故我们放弃使用 **Lucene7.7.1** 默认的 **BM25Similarity** 相似度函数，而是重写了一个相似度计算函数类，其相似度计算规则按照 **TFIDFSimilarity**（旧版本 **Lucene** 默认的相似度计算函数）进行编写：

```

# ***implement the document relevance score function by yourself***
class SimpleSimilarity(PythonClassicSimilarity):
    def lengthNorm(self, numTerms):
        return math.sqrt(numTerms)

    def tf(self, freq):
        return math.sqrt(freq)

    # def sloppyFreq(self, distance):
    #     return 2.0

    def idf(self, docFreq, numDocs):
        return math.log((numDocs+1)/(docFreq+1))+1

    def idfExplain(self, collectionStats, termStats):
        return Explanation.match(1.0, "inexplicable", [])

```

具体实现思路如下：

ClassicSimilarity 曾经是 **Lucene** 的默认评分公式，但是从 **lucene-6.0** 开始已经改成 **BM25Similarity** 了。

通过层层继承相关基础类（见下图），我们可以实现并优化曾经的默认评分公式 **TFIDFSimilarity**，只需重载几个关键函数即可。

← → ↺ lucene.apache.org/core/7_7_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

```

public abstract class TFIDFSimilarity
extends Similarity

```

Implementation of Similarity with the Vector Space Model.

Expert: Scoring API.

TFIDFSimilarity defines the components of Lucene scoring. Overriding computation of these components is a convenient way to alter Lucene scoring.

```

from org.apache.lucene.search import PythonClassicSimilarity
from org.apache.lucene.search import BM25Similarity
from org.apache.lucene.search import Explanation

```

```

# ***implement the document relevance score function by yourself***

```

```

class SimpleSimilarity(PythonClassicSimilarity):
    def lengthNorm(self, numTerms):
        return math.sqrt(numTerms)

    def tf(self, freq):
        return math.sqrt(freq)

    # def sloppyFreq(self, distance):
    #     return 2.0

    def idf(self, docFreq, numDocs):
        return math.log((numDocs+1)/(docFreq+1))+1

    def idfExplain(self, collectionStats, termStats):
        return Explanation.match(1.0, "inexplicable", [])

```

```

20 import org.apache.lucene.search.TermStatistics;
21 import org.apache.lucene.search.similarities.ClassicSimilarity;
22 import org.apache.lucene.index.FieldInvertState;
23
24
25 public class PythonClassicSimilarity extends ClassicSimilarity {
26
27     @Override
28     public native float lengthNorm(int numTerms);
29     @Override
30     public native float tf(float freq);
31     @Override
32     public native float sloppyFreq(int distance);
33     @Override
34     public native float idf(long docFreq, long numDocs);
35     @Override
36     public native Explanation idfExplain(CollectionStatistics collectionStats,
37                                         TermStatistics[] stats);
38 }

```

首先我们要明确早期 **TFIDFSimilarity** 的评分公式：

$$score(q, d) = coord(q, d) \times queryNorm(q) \times \sum_{t \in q} (idf^2(t, q) \times tf(t, d)) \times doc_len_norm(d)$$

再结合 **Lucene** 的官方文档对 **TFIDF** 的介绍：

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	0, 1
raw count	$f_{t,d}$
term frequency	$f_{t,d} / \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

最简单的方式是使用原始计数 $f_{t,d}$ 作为 **tf** 值，或者改为 **log normalization** 形式的 $\log(1+f_{t,d})$ ，官方文档中的默认实现是取 **freq** 的平方根即 **sqrt(freq)**作为 **tf** 值。

tf

```
public float tf(float freq)
```

Implemented as `sqrt(freq)`.

Specified by:

tf in class `TFIDFSimilarity`

Parameters:

freq - the frequency of a term within a document

Returns:

a score factor based on a term's within-document frequency

我们的方式是基于官方文档给出的建议进行实现的。（因为该方法尽管在文档有介绍，但实际上并不能在此版本的 **Lucene** 上直接调用，所以我们可以直接按照文档建议的设置复现该函数）

```
def tf(self, freq):
    return math.sqrt(freq)
```

②idf(inverse document frequency)

该函数被称作倒频率，表示词项 **T** 在所有文档中出现的频率。若它在所有

文档中出现的次数越多，表明这个词项 **T** 越不重要。

官方文档中的默认实现方式是 **inverse document frequency smooth** 的方式，我们也基于此进行实现。

idf

```
public float idf(long docFreq,  
                long docCount)
```

Implemented as $\log((\text{docCount}+1)/(\text{docFreq}+1)) + 1$.

Specified by:

```
idf in class TFIDFSimilarity
```

Parameters:

docFreq - the number of documents which contain the term

docCount - the total number of documents in the collection

Returns:

a score factor based on the term's document frequency

```
def idf(self, docFreq, numDocs):  
    return math.log((numDocs+1)/(docFreq+1))+1
```

它的不平滑版本是 **inverse document frequency**，即 $\log(N/n_t)$ ，其他常见的 **idf** 权重变体如下

Variants of inverse document frequency (idf) weight

weighting scheme	idf weight ($n_t = \{d \in D : t \in d\} $)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log \left(\frac{N}{1 + n_t} \right) + 1$
inverse document frequency max	$\log \left(\frac{\max_{t' \in d} n_{t'}}{1 + n_t} \right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

③lengthNorm

作为文档长度归一化因子，在实际公式中体现为 **lengthNorm**，在文档添加到索引中的时候根据该字段在文档中出现次数计算。

官方文档默认的实现方式是：

lengthNorm

```
public float lengthNorm(int numTerms)
```

Implemented as $1/\sqrt{\text{length}}$.

Specified by:

lengthNorm in class `TFIDFSimilarity`

Parameters:

numTerms - the number of terms in the field, optionally discounting overlaps

Returns:

a length normalization value

WARNING: This API is experimental and might change in incompatible ways in the next release.

基于此我们的实现为：

```
def lengthNorm(self, numTerms):  
    return math.sqrt(numTerms)
```

④sloppyFreq

官方文档默认的实现方式是：

sloppyFreq

```
public float sloppyFreq(int distance)
```

Implemented as $1 / (\text{distance} + 1)$.

Specified by:

sloppyFreq in class `TFIDFSimilarity`

Parameters:

distance - the edit distance of this sloppy phrase match

Returns:

the frequency increment for this match

See Also:

`PhraseQuery.getSlop()`

但再次查看最新文档，发现该函数已被弃用（**deprecated**），故无需重写。

⑤idfExplain

这个是用来输出文档评分过程的具体信息的函数，实际上如果不需要细致

了解评分信息组成的话就不需要实现这一部分，因为这对于我们最终的检索结果没有实质帮助，因此这里我们不做处理。

至此关于文档相似度函数 **TFIDFSimilarity** 的重写完毕，将我们重写后的相似度函数作为 **IndexWriter** 的设置项之一传入即可。

```
config = IndexWriterConfig(self.analyzer)
config.setSimilarity(SimpleSimilarity()) # index和search的similarity应保持一致
# config.setSimilarity(BM25Similarity(1.2, 0.75)) # BM25是默认方式
config.setOpenMode(IndexWriterConfig.OpenMode.CREATE)
# Writer
self.writer = IndexWriter(self.store, config) # 在Directory对象上实例化一个IndexWriter对象
```

然后，创建四个域用于存储前面提取得到的四类信息，由于任务要求只对 **text** 域进行关键词匹配搜索，故 **text** 域的属性设置与其他三个域的属性设置有所不同。

```
def init_fields(self):
    '''Initialize the fields in an array(the indices will be matching the data indices)
    For later use in the data writer'''
    fields = []

    # Define doc_no field
    doc_no_f = FieldType()
    doc_no_f.setStored(True)
    doc_no_f.setTokenized(False)
    fields.append(doc_no_f)

    # Define doc_type field
    doc_type_f = FieldType()
    doc_type_f.setStored(True)
    doc_type_f.setTokenized(False)
    fields.append(doc_type_f)

    # Define text_type field
    text_type_f = FieldType()
    text_type_f.setStored(True)
    text_type_f.setTokenized(False)
    fields.append(text_type_f)

    # Define text field
    text_f = FieldType()
    text_f.setStored(True)
    text_f.setTokenized(True)
    text_f.setIndexOptions(IndexOptions.DOCS_AND_FREQS_AND_POSITIONS) # documents, term frequencies and positions are indexed
    fields.append(text_f)

    self.fields = fields
    return fields
```

在建立好的四个域内写入数据的操作也十分简洁。

```

def write_data(self):
    '''Get an array of array with 4 fields
    Write data into the lucene writer'''
    # Idea was to commit regularly(every 10 docs) but commit are computationally expensive
    # printo = True
    self.init_writer()
    for d in self.data:
        # Insert array fields into lucene fields
        f0 = Field(self.fields_n[0], d[0], self.fields[0])
        f1 = Field(self.fields_n[1], d[1], self.fields[1])
        f2 = Field(self.fields_n[2], d[2], self.fields[2])
        f3 = Field(self.fields_n[3], d[3], self.fields[3])
        # Add lucene fields to a lucene document
        doc = Document()
        doc.add(f0)
        doc.add(f1)
        doc.add(f2)
        doc.add(f3)
        # Write the documents into the lucene index
        self.write_doc(doc)
    # Close writer
    self.get_fields_name()
    self.close_writer()

```

3.查询

首先我们要明确 **lucene** 的查询语句格式为 “**field : keywords**”，按照任务要求我们的查询域只是 **text**，所以 **field** 已经固定为 **TEXT** 了，那么另外的参数只有要查询的关键词 **keywords** 了。

我们先初始化查询器的相关参数信息，和建立索引时的设置保持一致即可，包括我们自定义的相似度函数、分析器、存储索引位置信息等。

```

if __name__ == "__main__":
    lucene.initVM(vmargs=['-Djava.awt.headless=true'])
    # searcher model
    path = os.path.expanduser('~/.Search_engine/index1')
    info = [StandardAnalyzer(), SimpleFSDirectory(Paths.get(path))]
    searcher_class = searcher(path, info)

```

```

def init_searcher(self):
    '''Initializes the lucene searcher'''
    self.isearcher = IndexSearcher(DirectoryReader.open(self.store)) # 实例化一个IndexSearcher对象，它的参数为SimpleFSDirectory对象
    self.isearcher.setSimilarity(SimpleSimilarity()) # index和Search的similarity应保持一致

```

因为要求以命令行形式输入查询，格式如 “**search --hits=27 new york**”，所以我们对命令行输入部分进行了相应处理，包括错误提醒等功能，最终将需要返回的结果数 **hits** 和关键词 **keywords** 传给查询函数中。

```

print("-----Search engine for TDT3 based on pylucene-----")
running = True
while running:
    print("\n***query example: search --hits=27 new york***")
    query_statement = input('#')
    if query_statement == 'exit!!!':
        print("Goooooooood Bye!!!")
        break
    if ' ' in query_statement:
        query_split = query_statement.split(' ', 1)
        operation = query_split[0]
        if operation != 'search':
            print("Wrong operation, please input again.")
            continue
        query_split2 = query_split[1]
        if ' ' in query_split2:
            query_split3 = query_split2.split(' ', 1)
            hits = query_split3[0]
            if hits[:7] != '--hits=':
                print("Wrong arguments, please input again.")
                continue
            hit = hits[7:]
            if not hit.isdigit():
                print("Wrong arguments, please input again.")
                continue
            keywords = query_split3[1]
            searcher_class.query('text', keywords, int(hit))
        else:
            print("Wrong input format, please input again.")
            continue

```

然后我们需要实例化一个 **QueryParser** 对象，它描述查询请求，解析 **Query** 查询语句，将由 **keywords** 和 **filed** 组成的查询语句传入对象中进行查询。

```

self.init_searcher()
arg = self.format_query(field, param)
qp = QueryParser('text', self.analyzer) # 实例化一个QueryParser对象，它描述查询请求，解析Query查询语句
query = qp.parse(str(arg)) # 以查询语句为参数

```

我们使用查询器的 **search** 方法获得前 **100** 个相关的文档。

```

result = self.isearcher.search(query, 100).scoreDocs

```

再根据 **hits** 值输出前 **top k** 个相关的文档信息，打印其顺序、得分、文档序号、文档类型、文本内容等信息，同时采取只显示前两行文本内容作为 **summary snippets** 来节省显示空间。

```

def print_results(self, result, hit):
    '''Display results in organized way'''
    i = 0
    for r in result[:hit]: # 返回1~top_k结果
        i = i + 1
        doc = self.isearcher.doc(r.doc)
        print("-----")
        print(str(i) + ':\t score:' + str(r.score) + '\t DOCNO:' + str(doc.get('doc_no')) + '\t DOCTYPE:' + str(doc.get('doc_type')) + '\t TEXT:')
        # print('text: ' + doc.get('text'))
        # 对长文本做summary snippets提取
        text = doc.get('text')
        cnt = 0
        for j in range(len(text)):
            if cnt == 3:
                break
            if text[j] == '\n':
                cnt = cnt + 1
        print('text(only show the first two lines): ' + text[:j] + '.....')

```

至此实验的关键步骤已叙述完毕，下面为实验结果展示。

三、实验结果

下面为文档要求的 5 条查询语句的实际演示情况。

Q1: search --hits=10 hurricane

```

Run: query_test.py
/home/wzp/anaconda3/envs/lucene/bin/python /home/wzp/Search_engine/query_test.py
-----Search engine for TDT3 based on pylucene-----

***query example: search --hits=27 new york***
#search --hits=10 hurricane
#EXECUTING SIMPLE QUERY#
query: text: hurricane
##NEW QUERY: field=text; param=hurricane ##
Time taken = 0.031235933303833008s

-----
1:  score:851.8217163085938  DOCNO:NYT19981108.0132  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Jose Antonio Amaya Garcia has lived long enough to survive three hurricanes,
but the devastation wrought on Central America by the gargantuan maelstrom
.....
-----
2:  score:851.8217163085938  DOCNO:NYT19981108.0095  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Jose Antonio Amaya Garcia has lived long enough to survive three hurricanes,
but the devastation wrought on Central America by the gargantuan maelstrom
.....
-----
3:  score:570.0611572265625  DOCNO:NYT19981113.0421  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
All that remains of the cruise ship are two life rafts, seven life
jackets, part of a wooden staircase, crew members' photographs clutched
-----

```

```

-----
4:  score:485.24053955078125  DOCNO:NYT19981003.0052  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
In the days before Hurricane Georges struck the Gulf Coast last week,
storm forecasters faced one of their most trying responsibilities:
.....
-----
5:  score:485.24053955078125  DOCNO:NYT19981005.0056  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
In the days before Hurricane Georges struck the Gulf Coast last week,
storm forecasters faced one of their most trying responsibilities:
.....
-----
6:  score:435.39141845703125  DOCNO:NYT19981026.0341  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
In many ways, the capital of Puerto Rico does not look strikingly
different from cities in the 50 states like Miami, Houston and parts
.....
-----
7:  score:410.4629821777344  DOCNO:NYT19981020.0178  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
For 11 days last month, Hurricane Georges plowed across the Caribbean,
killing at least 500 people and inflicting property damage estimated
.....
-----
-----
8:  score:381.7594299316406  DOCNO:VOA19981201.0500.0794  DOCTYPE:NEWS  TEXTTYPE:TRANSCRIPT
text(only show the first two lines):
The hurricane season has officially ended, although one storm is still
pushing its way through the Atlantic Ocean. Hurricane Nicole is no
.....
-----
9:  score:356.56011962890625  DOCNO:NYT19981225.0038  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
He had done time in jail and been through the Army and he was closing
in on the middleweight boxing crown when race riots convulsed the
.....
-----
10: score:353.0643310546875  DOCNO:APW19981028.1138  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Hurricane Mitch paused in its whirl through the western Caribbean
on Wednesday to punish Honduras with 120 mph (205 kph) winds, sweeping
.....
-----

```

Q2: search --hits=10 mitch george（多关键词查询）


```
***query example: search --hits=27 new york***
```

```
#search --hits=10 mitch george
```

```
#EXECUTING SIMPLE QUERY#
```

```
query: text: mitch george
```

```
###NEW QUERY: field=text; param=mitch george ###
```

```
Time taken = 0.1508498191833496s
```

```
-----  
1:  score:585.3491821289062  DOCNO:NYT19981103.0407  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
George W. Bush set down his peanut butter and raspberry jelly sandwich  
and wrinkled his nose, as though the aide on the other side of the
```

```
.....
```

```
-----  
2:  score:516.120849609375  DOCNO:NYT19981105.0262  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
When the American tenor Anthony Dean Griffey portrayed the hulking,  
half-witted Lennie in Carlisle Floyd's opera ``Of Mice and Men'' at
```

```
.....
```

```
-----  
3:  score:499.86614998234375  DOCNO:NYT19981127.0213  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
``Curious George'' is a best-selling children's book series. Furious  
George is a struggling punk rock band. You wouldn't necessarily confuse
```

```
.....
```

```
-----  
4:  score:437.10992431640625  DOCNO:NYT19981231.0319  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
One is an Argentine-born preacher who appeals to both Hispanic and  
Anglo audiences. Another is a young megachurch pastor in Southern
```

```
.....
```

```
-----  
5:  score:410.4134826660156  DOCNO:NYT19981108.0132  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
Jose Antonio Amaya Garcia has lived long enough to survive three hurricanes,  
but the devastation wrought on Central America by the gargantuan maelstrom
```

```
.....
```

```
-----  
6:  score:410.4134826660156  DOCNO:NYT19981108.0095  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
Jose Antonio Amaya Garcia has lived long enough to survive three hurricanes,  
but the devastation wrought on Central America by the gargantuan maelstrom
```

```
.....
```

```
-----  
7:  score:406.80426025390625  DOCNO:NYT19981119.0278  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
```

```
text(only show the first two lines):
```

```
Sometimes, from some angles, Milwaukee can fool you. Is this Chicago?  
There's the lake: the same immense lake, Lake Michigan, stretching
```

```
.....
```

```

-----
8:  score:402.1751403808594      DOCNO:PRI19981102.2000.0373      DOCTYPE:NEWS      TEXTTYPE:TRANSCRIPT
text(only show the first two lines):
Mitch is now a mere tropical storm but as a Hurricane it was devastating.
At one point during its destructive sweep through the Carribean last
.....
-----
9:  score:371.0003662109375      DOCNO:APW19981109.0776      DOCTYPE:NEWS      TEXTTYPE:NEWSWIRE
text(only show the first two lines):
As disease and starvation threatened to raise the death toll from
Hurricane Mitch, Central American leaders were flying Monday to El
.....
-----
10: score:365.7735290527344      DOCNO:NYT19981219.0275      DOCTYPE:NEWS      TEXTTYPE:NEWSWIRE
text(only show the first two lines):
The QE2 was docked at Casablanca in 1989 when Jeanie Bowers Scott,
then performing on board as an opera singer, received a call from
.....

```

Q3: search --hits=10 bill Clinton Israel

```

***query example: search --hits=27 new york***
#search --hits=10 bill Clinton Israel
#EXECUTING SIMPLE QUERY#
query: text: bill Clinton Israel
###NEW QUERY: field=text; param=bill Clinton Israel ###
Time taken = 0.20850181579589844s

-----
1:  score:1400.121826171875      DOCNO:NYT19981220.0116      DOCTYPE:NEWS      TEXTTYPE:NEWSWIRE
text(only show the first two lines):
It appeared to be the ultimate comeback in a career marked by seemingly
miraculous political resurrections. The night of Tuesday, Nov. 3,
.....
-----
2:  score:1253.8978271484375      DOCNO:NYT19981127.0249      DOCTYPE:NEWS      TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Following are President Clinton's answers to the 81 questions put
to him by the House Judiciary Committee as part of its impeachment
.....
-----
3:  score:1037.32470703125      DOCNO:NYT19981018.0155      DOCTYPE:NEWS      TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Shortly after he took office in 1993, President Clinton traveled to
Silicon Valley to lay out his vision of a robust U.S. economy buoyed
.....

```

4: score:1037.32470703125 DOCNO:NYT19981018.0144 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

Shortly after he took office in 1993, President Clinton traveled to
Silicon Valley to lay out his vision of a robust U.S. economy buoyed

.....

5: score:1037.32470703125 DOCNO:NYT19981018.0235 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

Shortly after he took office in 1993, President Clinton traveled to
Silicon Valley to lay out his vision of a robust U.S. economy buoyed

.....

6: score:978.6189575195312 DOCNO:NYT19981213.0146 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

Following are excerpts from weekend editorials published in newspapers
across the United States on the impeachment proceedings against President

.....

7: score:844.3718872070312 DOCNO:NYT19981024.0007 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

After nine days and nights of tortuous negotiations, Yasser Arafat,
the Palestinian leader, and Prime Minister Benjamin Netanyahu of Israel

.....

8: score:816.1744384765625 DOCNO:MNB19981203.2100.2413 DOCTYPE:NEWS TEXTTYPE:CAPTION

text(only show the first two lines):

The current fight over impeachment, what has been a string of misdeeds
by the chief executive are all but certain to become the chief legacy

.....

9: score:780.1347045898438 DOCNO:NYT19981114.0164 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

Lisa Smart's presence on the operating table at Beth Israel Medical
Center on a Friday afternoon last November began in an utterly unremarkable

.....

10: score:753.8434448242188 DOCNO:NYT19981210.0357 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

An Egyptian gas pipeline is snaking its way east across the Sinai
desert, laying the groundwork for a regional power network that could

.....

Q4: search --hits=10 "newt gingrich" down (双引号多关键词查询)

query example: search --hits=27 new york

#search --hits=10 "newt gingrich" down

#EXECUTING SIMPLE QUERY#

query: text: "newt gingrich" down

###NEW QUERY: field=text; param="newt gingrich" down ###

Time taken = 0.2686271667480469s

1: score:1547.8936767578125 DOCNO:NYT19981107.0032 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

House Speaker Newt Gingrich, who orchestrated the Republican takeover
of Congress in 1994 and presided this year over what at times seemed

.....

2: score:1406.1376953125 DOCNO:NYT19981106.0531 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

House Speaker Newt Gingrich, who orchestrated the Republican takeover
of Congress in 1994 and pressed the impeachment inquiry into President

.....

3: score:1186.095458984375 DOCNO:NYT19981116.0350 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

Hours after his swift emergence last week as the presumptive new Speaker
of the House, Rep. Bob Livingston telephoned an old friend, former

.....

4: score:1097.7572021484375 DOCNO:NYT19981105.0509 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

A struggle for control of the House is under way, with Rep. Robert
Livingston conducting a telephone campaign that could lead to him

.....

5: score:1091.6597900390625 DOCNO:NYT19981106.0479 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

It seemed somehow the ultimate irony of this strange political year,
the year stigmatized by President Clinton's affair with Monica Lewinsky.

.....

6: score:1091.6597900390625 DOCNO:NYT19981106.0549 DOCTYPE:NEWS TEXTTYPE:NEWSWIRE

text(only show the first two lines):

It seemed somehow the ultimate irony of this strange political year,
the year stigmatized by President Clinton's affair with Monica Lewinsky.

.....

7: score:1068.1207275390625 DOCNO:NBC19981106.1830.0071 DOCTYPE:NEWS TEXTTYPE:CAPTION

text(only show the first two lines):

Good evening. Major news tonight, when the congressional election
results came in on tuesday night, a huge disappointment for the republicans,

.....

```

-----
8:  score:1050.5810546875  DOCNO:NYT19981107.0084  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
A day after House Speaker Newt Gingrich's abrupt resignation, the
race for House speaker was transformed into a spirited free-for-all,
.....
-----
9:  score:1012.3668823242188  DOCNO:NYT19981107.0119  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Their wish finally came true: Democrats helped topple House Speaker
Newt Gingrich. But rather than celebrate, many of Gingrich's toughest
.....
-----
10:  score:995.115478515625  DOCNO:NYT19981106.0497  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
All year long, Rep. Robert Livingston made no effort to mask his lean
and hungry look. ``I'm running whether they like it or not,' the
.....

```

Q5: search --hits=10 nba strike closed-door

```

***query example: search --hits=27 new york***
#search --hits=10 nba strike closed-door
#EXECUTING SIMPLE QUERY#
query: text: nba strike closed-door
###NEW QUERY: field=text; param=nba strike closed-door ###
Time taken = 0.22744250297546387s

-----
1:  score:973.6868896484375  DOCNO:NYT19981231.0243  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
In terms of material things, there's no place like New York and no
time like the Christmas holidays. The rituals of buying, giving and
.....
-----
2:  score:846.916259765625  DOCNO:NYT19981023.0368  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Yvette Weilacker sits in the kitchen of a small house in Buffalo waiting
for the contractions to begin. Her baby, already named Nicholas Lee
.....
-----
3:  score:846.916259765625  DOCNO:NYT19981024.0057  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Yvette Weilacker sits in the kitchen of a small house in Buffalo waiting
for the contractions to begin. Her baby, already named Nicholas Lee
.....
-----

```

```

-----
4:  score:846.916259765625  DOCNO:NYT19981110.0218  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Yvette Weilacker sits in the kitchen of a small house in Buffalo,
N.Y., waiting for the contractions to begin. Her baby, already named
.....
-----
5:  score:806.4852294921875  DOCNO:NYT19981031.0165  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
Four short years ago, hockey was hot. The Rangers won the Stanley
Cup and rap artists wore National Hockey League jerseys, even if they
.....
-----
6:  score:797.1123657226562  DOCNO:NYT19981108.0125  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
On Oct. 30, children, teen-agers and adults packed the balcony overlooking
the gymnasium floor at the Indiana Institute of Technology to watch
.....
-----
7:  score:793.3074340820312  DOCNO:NYT19981117.0284  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
When Andrei Sokolov came to this nuclear city more than 30 years ago
it was a bastion of privilege for the Soviet Union's scientific elite.
.....
-----

```

```

-----
8:  score:763.0249633789062  DOCNO:NYT19981229.0365  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
As the prospect of the National Basketball Association season being
canceled increases, so has talk that a rival league will be formed.
.....
-----
9:  score:662.2989501953125  DOCNO:NYT19981026.0247  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
One of the most important things travelers buy with their vacation
dollars is time _ the precious weeks, days or even hours in which
.....
-----
10: score:613.2070922851562  DOCNO:NYT19981027.0281  DOCTYPE:NEWS  TEXTTYPE:NEWSWIRE
text(only show the first two lines):
As the interim director of the Mississippi Gaming Commission, Bruce
Nourse authorized the opening of the state's first casino in 1992.
.....

```

当输入格式不正确时，会提示错误要求重新输入；当输入“**exit!!!**”时，程序正常退出。

```

***query example: search --hits=27 new york***
#search --hits=27 new york
Wrong operation, please input again.

***query example: search --hits=27 new york***
#search --hits=27 new york
Wrong arguments, please input again.

***query example: search --hits=27 new york***
#search --hits=27 new york
Wrong arguments, please input again.

***query example: search --hits=27 new york***
#search
Wrong input format, please input again.

***query example: search --hits=27 new york***
#exit!!!
Goooooooood Bye!!!

Process finished with exit code 0

```

另外可以验证，如果不使用我们自定义的相似度计算函数而是使用默认的 **BM25Similarity**，则查询结果会有区别，以此证明我们自定义的相似度计算函数发挥了作用。

```

***query example: search --hits=27 new york***
#search --hits=10 hurricane
#EXECUTING SIMPLE QUERY#
query: text: hurricane
###NEW QUERY: field=text; param=hurricane ###
Time taken = 0.04712986946105957s

1: score:8.121366500854492   DOCNO:CNN19981019.1600.0586   DOCTYPE:NEWS   TEXTTYPE:CAPTION
text(only show the first two lines):
Hurricane Madeline continues to steam toward Mexico's western pacific
coast, while hurricane Lester is weakening and moving out to sea.
.....

2: score:8.054352760314941   DOCNO:CNN19981008.1130.0447   DOCTYPE:NEWS   TEXTTYPE:CAPTION
text(only show the first two lines):
Hurricane georges is now the third costliest hurricane on record.
Actuaries estimate georges caused $2.5 billion in insured losses throughout
.....

3: score:8.045164108276367   DOCNO:V0A19981201.0500.0794   DOCTYPE:NEWS   TEXTTYPE:TRANSCRIPT

```

此外，任务文档的其他要求均在代码及上述实验结果图中有所体现，至此

ps: 项目完整代码可见压缩包或 **github** 库（暂为私人库，**ddl** 后设为公开）
https://github.com/xiaopeng-whu/Search_engine

通过此次实验，我对信息检索的相关概念和技术有了更深入的了解，尤其对如何使用 **Lucene** 构建一个基础的搜索引擎有了更深的掌握，我对数据的处理与存储方式也有了更加清晰的认识，为日后的其他与信息检索有关的工作打下了基础、积累了经验。

指导教师评语及成绩

批阅日期:

实验报告说明

- 24

4. 实验方案设计（思路、步骤和方法等）：这是实验报告极其重要的内容。包括概要设计、详细设计和核心算法说明及分析，系统开发工具等。应同时提交程序或设计电子版。

对于**设计型和综合型实验**，在上述内容基础上还应该画出流程图、设计思路和设计方法，再配以相应的文字说明。

对于**创新型实验**，还应注明其创新点、特色。

5. 结论（结果）：即根据实验过程中所见到的现象和测得的数据，做出结论（可以将部分测试结果进行截屏）。

6. 小结：对本次实验的心得体会，所遇到的问题及解决方法，其他思考和建议。

7. 指导教师评语及成绩：指导教师依据学生的实际报告内容，用简练语言给出本次实验报告的评价和价值。