

实验 4-Lucene

姓名：王煌基

学号：519030910100

班级：F1903004

一、实验概览

基于对 Java 的全文检索库 Lucene 基本原理的学习（包括创建、搜索索引等操作），应用适当的分词器与 Analyzer 实现网页索引与搜索程序。

二、实验环境

1. 个人笔记本电脑
2. 操作系统：windows10 专业版
3. 使用软件：Visual Studio Code; Docker Desktop

三、实验过程

练习：实现一个中文网页索引与搜索程序

爬取一定数量（>5k）的中文网页（可利用之前实验爬取的网页），修改 `IndexFiles.py` 和 `SearchFiles.py`，对这些中文网页建立索引并进行搜索，搜索时需要打印出检出文档的路径、网页标题、url。

doc 的 Field 中需要有 `name` (文件名)，`path` (文件路径)，`title` (网页标题)，`url` (网页地址)，`contents` (索引的文件内容)

搜索时显示出相关信息

【解】

根据习得的相关知识，可以将该练习分成以下几个小任务，方便我们的完成：

①基于 lab2 中的多线程爬取网页程序来爬取一定数量（>5k）的中文网页，并保存其网页链接、内容及文件名于 `index.txt` 中方便我们后续操作中直接调用。

②利用恰当的分词器、Analyzer 处理所有已知网页，将其内容建立相应索引，并存放于 `index` 文件夹中以备搜索索引时的直接调用。

③利用恰当的分词器、Analyzer 处理搜索内容，并进行相应的搜索过程。

将本练习分成三部分之后将会比较容易完成。

①基于 lab2 中的多线程爬取网页程序来爬取一定数量（>5k）的中文网页，并保存其网页链接、内容及文件名于 `index.txt` 中方便我们后续操作中直接调用。

此处直接利用 lab2 的代码 `crawler_multi_thread.py` 并进行一定的修改以适用于该练习，下面说明修改的地方：

1. 对 add_page_to_folder 函数的文件输出名进行了修改，确保都是以.html 结尾（方便直接在网页中打开以核实是否爬取成功，）：

```
1.     try:
2.         if not os.path.exists(folder): # 如果文件夹不存在则新建
3.             os.mkdir(folder)
4.             #确保所有文件均以.html 结尾
5.             f = open(os.path.join(folder, filename+'.html'), 'wb')
6.             f.write(content) # 将网页存入文件
7.             f.close()
8.     except:
9.         pass
```

并且均以.html 结尾的文件可方便后面对文件夹内文件的访问：

```
1. if not filename.endswith('.html'):
2.     continue
```

2. 对 working 函数的爬取网页的数据规模进行了修改，从 count=300->count=5000：

```
1.         varLock.acquire()
2.         count += 1
3.         if(count>=5000):
```

3. 爬取网页的选择，此处我选择了 17yy 小游戏网站，其中数据较为丰富，便于索引：

```
1. q = queue.Queue()
2. #q.put("https://www.sjtu.edu.cn/")
3. q.put("https://www.17yy.com")
```

从而我们的①任务就完成了。

②利用恰当的分词器、Analyzer 处理所有已知网页，将其内容建立相应索引，并存放于 index 文件夹中以备搜索索引时的直接调用。

首先我先确定了本次练习需要利用 jieba 分词器分词，并且所有词之间以空格间隔，从而可以方便地利用 WhitespaceAnalyzer 进行索引的建立，因为 WhitespaceAnalyzer 本身就是基于空格作为间隔符的词汇分割分词器：

```
1. import jieba
2. analyzer = WhitespaceAnalyzer()
```

其次，对于任务①中得到的 index.txt，我们需要对每个网页进行预处理，得到其 path、title、url、name、contents 的属性并为此建立索引。由于在 index.txt 中，我们的网页链接与文件名之间间隔为\t，因此可以利用字符串的处理方式将其分开，并建立从文件名到网页链接的字典关系以便后续调用：

```
1. File = open("index.txt", "r")
2. dic = {}
3. for url in File.readlines():
4.     front = url.find('\t')
5.     dic[url[front+1:-1]+' .html']=url[:front]
```

然后，由于网页源代码中包含 HTML tag（例如<html>,<body>等），在加入 lucene 前，可以用 BeautifulSoup 等库过滤文档中的 HTMLtag，然后利用正则表达式清除其余额外字符以方便分词操作：

```
1. contents = file.read()
2. soup = BeautifulSoup(contents)
3. contents = ''.join(soup.findAll(text=True))
4. #contents 内仅保留汉字进行分词
5. contents = re.sub("[^\u4e00-\u9fa5]", "", contents)
```

使用正则表达式前，存在很多的空格以及无效字符：

```
_location=window.location.href;if(_location.indexOf("/m/")!=-1){var ua=navigator.userAgent,ua=ua.toLowerCase().replace(/-/g,"");if(ua.match(/(Android)/i)){if(confirm('是否跳转到手机游戏频道? ')){location.href='http://h.17yy.com/';}}if(ua.match(/(iPhone|iPod)/i)){if(confirm('是否跳转到手机游戏频道? ')){location.href='http://h.17yy.com/';}}}  
  
[if IE 6]>  
<script src="http://css.17yy.com/js/DD_belatedPNG_0.0.8a-min.js"></script>  
<script>  
    DD_belatedPNG.fix('#jingdian ol li a img');  
</script>  
<![endif]>  
  
body{font-family:'Microsoft Yahei';}  
.sub_con_body a{margin: 2px 10px 2px 5px;}  
.sub_con .sub_con_body{white-space: nowrap;}  
#commentcontent a{margin:0px!important;}  
#al_nv_c{width:1200px;}
```

26仙侠放置中文版[策略]
27王国的崛起中文版[动作]
28奥特曼大作战4[冒险]

更多游戏
最新好玩小游戏无敌版射击动作冒险双人

使用正则表达式之后只剩下中文，可以更方便分词操作：

经典小游戏大全经典小游戏是否跳转到手机游戏频道是否跳转到手机游戏频道净化网络环境关于本站首页动作益智体育射击冒险养成策略敏捷休闲换装经营双人三人综合专题无敌版页游盒子网页游扮演养成塔防武侠小游戏经典拳皇解谜挂机植物大战僵尸超级玛丽消消看连连看愤怒的小鸟祖玛版泡超人俄罗斯方块攻略冒险金庸群侠传冒险王火影忍者火影对战冒险岛传功小游戏大全传功勇士三国刚火柴人街头霸王中文版中文无敌版军迷合金弹头魂斗罗闪客快打打枪战争空战飞机坦克二战前线战枪战无敌版女生结婚换装美容美发公主古装宫廷做蛋糕做饭后宫炫舞恋爱美女测试画画钢琴橙光酷跑钓鱼乐高小火车射箭跑酷汽车赛车飞车卡丁车摩托越野大卡车开车学车倒车驾驶停车自行车挖客梦魔龙诀变态版神曲变态版嗜魂神魔传说暗黑世界超变版西游伏妖变态版三国群英传核金弹头满版炎黄大陆变态版唐门六道变态版之封神霸业真龙主宰之盛世遮天新莽荒纪斗罗大陆裁决战歌侠客开服航海家中文版帝国中文版后门火车中文版中世纪放置中文版僵尸任务无敌版像素城市之战萨拉文版冒险王之神兵传奇终极无敌速升版金庸群侠传终极无敌版经典连连看雷索的战争初章中文版便重制版版最后的战役之联合之城中文版英雄大作战终极无敌版爱德华王子中文版仙侠放置中文版被地牢大师中文版唐门六道变态版之封神霸业放置立方体中文版武器收集中文版一切皆可当武器战火钮造世界中文版游戏开发模拟器中文版死神火影武侠浮生记中文版老爹饼干圣代店中文版我的世界

对于我们得到的中文，我们需要利用 jieba 来进行分词操作，此处利用精准分词：

```
1. contents = jieba.cut(contents, cut_all=False)
2. contents = ' '.join(contents)
```

最后将我们处理过后的相关数据均存入 doc 内再写入 writer 即可完成该任务：

```
1. title = soup.find('title').string
2. doc = Document()
3. doc.add(Field("path", path, t1))
4. doc.add(Field("title", title, t1))
5. doc.add(Field("url", dic[filename], t1))
6. doc.add(Field("name", filename, t1))
7. #对 contents 利用 jieba 进行分词操作
8. if len(contents) > 0:
9.     doc.add(Field("contents", contents, t2))
10. else:
11.     print("warning: no content in %s" % filename)
12. writer.addDocument(doc)
```

③利用恰当的分词器、Analyzer 处理搜索内容，并进行相应的搜索过程。

首先，我们需要对搜索的内容进行分词处理：

```
1. print ("Hit enter with no input to quit.")
2. command = input("Query:").replace(" ", "")
3. command = jieba.cut(command, cut_all=False)
4. command = " ".join(command)
```

然后利用 QueryParser 对 Analyzer 的内容进行关键词为 command 的检索并记录输出：

```
1. query = QueryParser("contents", analyzer).parse(command)
2. scoreDocs = searcher.search(query, 50).scoreDocs
3. print ("%s total matching documents." % len(scoreDocs))
4. for scoreDoc in scoreDocs:
5.     doc = searcher.doc(scoreDoc.doc)
6.     print ('path:', doc.get("path"), 'title:', doc.get("title"), 'url:', doc.get("url"), '
    name:', doc.get("name"))
```

进而，我们的练习得到了解决。

四、实验问题及解决

问题：在实验过程中出现了如下的错误方式，为什么会出现，如何修改呢？

```
1. Failed in indexDocs: 'gbk' codec can't decode byte 0xa1 in position 444: illegal multibyte sequence
2. adding httpwww.muzhiwan.com.html
3. Failed in indexDocs: 'gbk' codec can't decode byte 0xa9 in position 57: illegal multibyte sequence
4. adding httpwww.niba.com.html
5. Failed in indexDocs: 'gbk' codec can't decode byte 0x91 in position 262: illegal multibyte sequence
6. adding httpwww.veryhuo.com.html
7. Failed in indexDocs: 'gbk' codec can't decode byte 0x80 in position 83: illegal multibyte sequence"
```

【解】

对于网页的编码处理，为了方便处理，基于相关知识¹，我均使用 gbk 解码方式处理文件：

```
1. path = os.path.join(root, filename)
2. file = open(path, encoding='gbk')
3. contents = file.read()
```

但事实上，在爬取过程中，存在部分网页的编码并不是 gbk 的编码方式，而是 utf-8 的：

```
▼<head>
  <meta charset="UTF-8">
  <title>最火软件站_提供免费软件下载|绿色软件下载|手机软件下载_最火软件站</title>
  <meta name="keywords" content="最火软件站,免费软件,绿色软件,软件下载,手机软件,游戏下载,电脑教程,源码下载,网页特效,手机教程,游戏攻略,IT资讯">
  -
▼<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  <title>经典小游戏大全 - 17yy经典小游戏</title>
  <meta name="keywords" content="17yy, 小游戏大全, 经典小游戏, 免费小游戏, 双人小游戏大全">
  <meta name="description" content="17YY经典小游戏, 为您提供包括动作、体育、益智、射击、冒险、策略、装扮、敏捷等各种类型经典小游戏大全。17yy小游戏, 致力做国内最优秀的小游戏网站。17yy, 一起玩玩吧。">
```

从而出现了编码解码的错误，错误类型为：**UnicodeDecodeError**，因此可以通过特判的方式将无法进行 gbk 解码的网页文件用 utf-8 的方式进行解码即可，即默认以 gbk 方式解码，当出现问题时再进行 utf-8 的方式解码：

```
1. file = open(path, encoding='gbk')
2. try:
3.     contents = file.read()
4. except UnicodeDecodeError:
5.     file = open(path, encoding='utf-8', error='ignore')
6.     contents = file.read()
```

从而，我们能够规避掉由于编码导致的问题。

¹ GBK 编码方式的编码是以中国国情而创造的，在国际上的兼容性不好，这也是为什么大多数的网页是使用 UTF-8 编码而不是 GBK。

from: https://blog.csdn.net/qq_25408423/article/details/80649432



















title: GBK 和 UTF-8 文字编码有什么区别？GBK 有什么好处？

五、实验结果

运行 crawler_multi_thread.py 爬取超过 5000 个的网页：

```
5501
5502
5503
5504
5505
5506
5507
285.39349007606506
```

并且在 \html 文件夹中产生对应的网页文件：

名称	修改日期	类型	大小
 httpgame.51.com.html	2020/10/14 11:30	Chrome HTML D...	77 KB
 http.17yy.com.html	2020/10/14 11:29	Chrome HTML D...	12 KB
 httppc.tgbus.com.html	2020/10/14 11:30	Chrome HTML D...	149 KB
 httpzdf.17yy.com.html	2020/10/14 11:30	Chrome HTML D...	75 KB
 httpswww.17yy.com.html	2020/10/14 11:30	Chrome HTML D...	126 KB
 httpswww.17yy.comabout.html.html	2020/10/14 11:30	Chrome HTML D...	8 KB
 httpswww.17yy.comahsjcbb.html	2020/10/14 11:30	Chrome HTML D...	9 KB
 httpswww.17yy.combusiness.html.h...	2020/10/14 11:30	Chrome HTML D...	8 KB
 httpswww.17yy.comcjzg.html	2020/10/14 11:29	Chrome HTML D...	9 KB
 httpswww.17yy.comcontact.html.html	2020/10/14 11:30	Chrome HTML D...	8 KB
 httpswww.17yy.comdeclaration.ht...	2020/10/14 11:30	Chrome HTML D...	11 KB
 httpswww.17yy.comddl2d.html	2020/10/14 11:29	Chrome HTML D...	10 KB
 httpswww.17yy.comemembermy.ht...	2020/10/14 11:30	Chrome HTML D...	10 KB
 httpswww.17yy.comememberregist...	2020/10/14 11:30	Chrome HTML D...	12 KB
 httpswww.17yy.comepayapi.html	2020/10/14 11:30	Chrome HTML D...	1 KB
 httpswww.17yy.comgamecenter.ht...	2020/10/14 11:30	Chrome HTML D...	10 KB
 httpswww.17yy.comhqgczb.html	2020/10/14 11:29	Chrome HTML D...	9 KB
 httpswww.17yy.comjsxwbt.html	2020/10/14 11:29	Chrome HTML D...	9 KB


并且在 index.txt 中保存网页链接以及文件名：

```
index.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
https://www.17yy.com      httpswww.17yy.com
https://www.17yy.com/jsxwbthttpswww.17yy.comjsxwbt
https://www.17yy.com/wdcq httpswww.17yy.comwdcq
https://www.17yy.com/wshbt httpswww.17yy.comwshbt
https://www.17yy.com/tmld httpswww.17yy.comtmld
https://www.17yy.com/hqgczb
httpswww.17yy.comhqgczb
https://www.17yy.com/zlzzsszt
httpswww.17yy.comzlzzsszt
https://www.17yy.com/yhdlbthttpswww.17yy.comyhdlbt
https://www.17yy.com/tmldb
httpswww.17yy.comtmldb
https://www.17yy.com/xkmeng
httpswww.17yy.comxkmeng
https://www.17yy.com/mbi2 httpswww.17yy.commbi2
第 1 行, 第 1 列 100% Unix (LF) UTF-8
```

运行 IndexFiles.py 建立索引，运行结果：

```
adding httpwww.wangye.cn.html
adding httpwww.wanyx.com.html
adding httpwww.yxbao.com.html
adding httpwww.yzz.cn.html
adding httpxiaoyouxi.2345.com.html
commit index
....done
0:07:12.909998
```

其中建立的索引保存在\index 文件夹中：

<input type="checkbox"/> 名称	修改日期	类型	大小
 _c.cfe	2020/10/14 18:43	CFE 文件	1 KB
 _c.cfs	2020/10/14 18:43	CFS 文件	4,365 KB
 _c.si	2020/10/14 18:43	SI 文件	1 KB
 segments_9	2020/10/14 18:43	文件	1 KB
 write.lock	2020/10/13 17:25	lenovo lock file	0 KB

运行 SearchFiles.py 进行索引的检索：

首先搜索：战争游戏，分词为：战争 游戏：

```
root@00d769e9c5cb:/workspaces/lab4-Lucene# python SearchFiles.py
lucene 8.3.0
Hit enter with no input to quit.
Query:战争游戏
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 0.917 seconds.
Prefix dict has been built successfully.

Searching for: 战争 游戏
50 total matching documents.
path: html/httpwww.17yy.comszhanzhengjinhuaishi.html title: 【战争进化史小游戏大全】经典战争进化史小游戏 - 17yy小游戏 url: http://www.17yy.com/s/zhanzhengjinhuaishi name: httpwww.17yy.comszhanzhengjinhuaishi.html
path: html/httpwww.17yy.comswangguozhanzheng.html title: 【王国战争小游戏大全】经典王国战争小游戏 - 17yy小游戏 url: http://www.17yy.com/s/wangguozhanzheng name: httpwww.17yy.comswangguozhanzheng.html
path: html/httpswww.17yy.comszhanzhengshidai.html title: 【战争时代小游戏大全】经典战争时代小游戏 - 17yy小游戏 url: https://www.17yy.com/s/zhanzhengshidai name: httpswww.17yy.comszhanzhengshidai.html
```

再者搜索：战争游戏 NOT 游戏，分词为：战争 游戏 NOT 游戏：

```
path: html/httpwww.17yy.comf232625.html.html title: 【战争进化史2中文终极无敌版】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/232625.html name: httpwww.17yy.comf232625.html.html
Hit enter with no input to quit.
Query:战争游戏 NOT 游戏

Searching for: 战争 游戏 NOT 游戏
0 total matching documents.
Hit enter with no input to quit.
```

再者搜索：战争 AND 老爹 AND 汉堡 OR 公主 NOT 猴子 NOT 气球（分词一样）：

```
root@00d769e9c5cb:/workspaces/lab4-Lucene# python SearchFiles.py
lucene 8.3.0
Hit enter with no input to quit.
Query:战争 AND 老爹 AND 汉堡 OR 公主 NOT 猴子 NOT 气球
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 1.020 seconds.
Prefix dict has been built successfully.

Searching for: 战争 AND 老爹 AND 汉堡 OR 公主 NOT 猴子 NOT 气球
14 total matching documents.
path: html/httpswww.17yy.comsjingyingcanting.html title: 【经营餐厅类小游戏大全】经典经营餐厅类小游戏 - 17yy小游戏 url: https://www.17yy.com/s/jingyingcanting name: httpswww.17yy.comsjingyingcanting.html
path: html/httpwww.17yy.comsjingyingcanting.html title: 【经营餐厅类小游戏大全】经典经营餐厅类小游戏 - 17yy小游戏 url: http://www.17yy.com/s/jingyingcanting/ name: httpwww.17yy.comsjingyingcanting.html
path: html/httpwww.17yy.comf239860.html.html title: 【泡泡兔汉堡店中文终极无敌版】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/239860.html name: httpwww.17yy.comf239860.html.html
path: html/httpswww.17yy.comsjingying.html title: 【经营小游戏大全】经典经营小游戏 - 17yy小游戏 url: https://www.17yy.com/s/jingying name: h
```

再者搜索：精英游戏 OR 公主 AND 僵尸 NOT 大冒险（分词将精英与游戏分开）：

```
Hit enter with no input to quit.
Query:精英游戏 OR 公主 AND 僵尸 NOT 大冒险
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 0.898 seconds.
Prefix dict has been built successfully.

Searching for: 精英 游戏 OR 公主 AND 僵尸 NOT 大冒险
50 total matching documents.
path: html/httpwww.17yy.comf233236.html.html title: 【盟军战争英雄中文版】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/233236.html name: httpwww.17yy.comf233236.html.html
path: html/httpwww.9game.cn.html title: 九游手机网游_手游下载门户_好玩的手机游戏排行榜 url: http://www.9game.cn/ name: httpwww.9game.cn.html
```

最后搜索完整语句：植物大战僵尸喜欢森林冰火人：

```
Hit enter with no input to quit.
Query:植物大战僵尸喜欢森林冰火人

Searching for: 植物 大战 僵尸 喜欢 森林 冰火 人
50 total matching documents.
path: html/httpwww.17yy.comf208208.html.html title: 【你没玩过的植物大战僵尸4】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/208208.html name: httpwww.17yy.comf208208.html.html
path: html/httpwww.17yy.comf216572.html.html title: 【植物大战僵尸2山寨无敌版】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/216572.html name: httpwww.17yy.comf216572.html.html
path: html/httpwww.17yy.comf44100.html.html title: 【大富翁魔兽世界版】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/44100.html name: httpwww.17yy.comf44100.html.html
path: html/httpwww.17yy.comf10324.html.html title: 【小小可爱QQ堂】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/10324.html name: httpwww.17yy.comf10324.html.html
path: html/httpwww.17yy.comf133148.html.html title: 【超级祖玛】小游戏 - 17yy经典小游戏 url: http://www.17yy.com/f/133148.html name: httpwww.17yy.comf133148.html.html
```

六、实验体会

这是本课程的第三次实验作业，这次的任务难度并不算太大，但是对于初学者而言，初学者容易被代码中出现的“高大上”的库给弄晕从而降低做任务的主动性和积极性，因而完成起来仍旧具有一定的挑战性，但是当任务完成后，可以发现其实任务并不会太难。

在本次实验中，我再一次加深了对正则表达式的运用以及对之前 lab1、lab2-3 所学知识的巩固，这样避免了我学了新的知识就将旧的知识抛之脑后的后果，并且，对于练习中遇到的问题，我仍旧积极主动地去查阅相关资料，养成一个良好的学习习惯，收获匪浅！

当然，本次实验也存在不足之处，主要是由于选取的网站虽然数量较大但是具体的内容重复性高（热门游戏在每个网页均有体现），因而对选取的网站也具有一定的要求。