电类工程学导论 C 实验报告 4，5

518030910406　郑思榕

一、实验准备

1. 实验环境介绍
   1) 环境：在 windows 系统中使用 VirtualBox 5.2.18 安装 Ubuntu14.04 虚拟机，从而在 UNIX 系统环境下进行本次实验。
   2) 语言：　python 2.7
   3) 工具：本实验主要使用了基于 Java 的全文检索库 Lucene 和中文分词库 finalseg
      - 库 Lucene 的 analysis 模块主要负责词法分析及语言处理而形成 Term。
      - 库 Lucene 的 index 模块主要负责索引的创建，里面有 IndexWriter。
      - 库 Lucene 的 store 模块主要负责索引的读写。
      - 库 Lucene 的 QueryParser 主要负责语法分析。
      - 库 Lucene 的 search 模块主要负责对索引的搜索。
      - 库 Lucene 的 similarity 模块主要负责对相关性打分的实现。
      - 中文分词库 finalseg 支持 GBK,UTF8,Unicode 编码，使用 seg_list = finalseg.cut(command, find_new_word=True)将待分词的语句 command 分解成含一个个词语的列表。再结合" ".join(seg_list)语句即可得到词语间以空格相隔的字符串。
      - 分析器 WhitespaceAnalyzer 用来对查询语句进行词法分析和语言处理，其以空格字符作为分割符号。再本次实验中结合中文分词库 finalseg 即可对中文语句分词。
2. 实验目的
   1) 实现一个中文网页索引与搜索程序（ex1）

      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。运行 SearchFiles.py 如下图所示：

```
print 'path:', doc.get("path"), 'title:', doc.get("title"), \
      'url:', doc.get('url'), 'name:', doc.get("name")
```

```
Hit enter with no input to quit.
Query:战争游戏

Searching for: 战争 游戏
10 total matching documents.
path: C:/Users/Alpha/Desktop/py/baidu_sim/html\httpwf.qq.com title: 战争前线-WarF
ace-官方网站-腾讯游戏-孤岛危机系列射击巨作 url: http://wf.qq.com/ name: httpwf.qq.com
path: C:/Users/Alpha/Desktop/py/baidu_sim/html\httpwww.pcgames.com.cnkzztpcgameG
OW title: 战争机器PC_战争机器_太平洋游戏网战争机器专题 url: http://www.pcgames.com.cn
/kzzt/pcgame/GOW/ name: httpwww.pcgames.com.cnkzztpcgameGOW
path: C:/Users/Alpha/Desktop/py/baidu_sim/html\httpwww.7k7k.comflash_fl491_1.htm
 title: 战争小游戏_战争小游戏大全_战争小游戏全集_7k7k战争小游戏 - 7k7k小游戏 url: http:
//www.7k7k.com/flash_fl/491_1.htm name: httpwww.7k7k.comflash_fl491_1.htm
Hit enter with no input to quit.
Query:战争 NOT 游戏

Searching for: 战争 NOT 游戏
10 total matching documents.
path: C:/Users/Alpha/Desktop/py/baidu_sim/html\httpbaike.baidu.comview14949.htm
title: 越南战争_百度百科 url: http://baike.baidu.com/view/14949.htm name: httpbaik
e.baidu.comview14949.htm
path: C:/Users/Alpha/Desktop/py/baidu_sim/html\httpbaike.baidu.comview67404.htm
title: 普法战争_百度百科 url: http://baike.baidu.com/view/67404.htm name: httpbaik
e.baidu.comview67404.htm
```

2) 模拟实现搜索引擎的“site:”功能（对搜索的网站进行限制）(ex2)

示例：搜索sina.com.cn和baike.baidu.com上包含“国家”的网页。

```
Hit enter with no input to quit.
Query:国家

Searching for: 国家
10 total matching documents.
-------------------------
path: html\httpgb.533.com
title: 英国留学|英国留学费用|英国留
url: http://gb.533.com/
name: httpgb.533.com
-------------------------
path: html\httpwww.wlaap.com
title: 世界文艺网
url: http://www.wlaap.com/
name: httpwww.wlaap.com
-------------------------
```

```
Hit enter with no input to quit.
Query:国家 site:sina.com.cn

Searching for: 国家 site:sina.com.cn
10 total matching documents.
-------------------------
path: html\httpmatch.2012.sina.com.c
untry_t00001
title: 法国奖牌榜_2012伦敦奥运会_新浪网
url: http://match.2012.sina.com.cn/m
_country_t00001
name: httpmatch.2012.sina.com.cnmeda
_t00001
-------------------------
path: html\httpmatch.2012.sina.com.c
untry_t00001
title: 韩国奖牌榜_2012伦敦奥运会_新浪网
url: http://match.2012.sina.com.cn/m
_country_t00001
name: httpmatch.2012.sina.com.cnmeda
```

```
Hit enter with no input to quit.
Query:国家 site:baike.baidu.com

Searching for: 国家 site:baike.baid
10 total matching documents.
-------------------------
path: html\httpbaike.baidu.comview
title: 美国_百度百科
url: http://baike.baidu.com/view/2
name: httpbaike.baidu.comview2398.
-------------------------
path: html\httpbaike.baidu.comview
title: 英国_百度百科
url: http://baike.baidu.com/view/3
name: httpbaike.baidu.comview3565.
-------------------------
path: html\httpbaike.baidu.comview
```

3) 实现一个图片索引(ex3)

新建一个索引，输入文本，输出相关的图片地址，图片所在网页的网址，图片所在网页的标题。示例如下：

```
Hit enter with no input to quit.
Query:男装
Searching for: 男装
10 total matching documents.
imgurl: http://img01.taobaocdn.com/
url: http://list.taobao.com/market/
d=all&atype=b&style=grid&ppath=140
urltitle:
薄款-夹克-男装-淘宝网

-----------------------------------
imgurl: http://img01.taobaocdn.com/
url: http://list.taobao.com/market/
ype=0&random=false&viewIndex=1&yp4
urltitle:
中老年专区-男装-淘宝网
```

3. 实验原理：

分析网页源代码，提取其中的有关信息。对网页内容建立倒排索引表。利用 Lucene 建立索引文件，添加合适的 Field 构建索引。利用中文分词库 jieba 或 finalseg 对中文进行分词。

二、实验过程

在实验前先利用 lab3 的脚本爬取一定量网页。本次实验的 ex1 和 ex2，我爬取的根网址 seed 是新浪的英超板块：http://sports.sina.com.cn/g/premierleague/ 。ex3 爬取的根网址 seed 是京东的商品页面：https://item.jd.com/100007717032.html。接下来的实验过程分为两个部分，索引创建和搜索索引。

1) 索引创建：

- 创建一个 IndexWriter 用来写索引文件；
- 创建索引文档 Document，在 Document 中，针对不同类型的 Field,创立不同的 FieldType，并将所需信息填入 Field 加入到 Document 中；
- IndexWriter 将索引写到索引文件夹。

2) 搜索索引：

- 创建 IndexSearcher 准备进行搜索；

- 创建 analyzer 来对查询语句进行词法分析和语言处理；
- 创建 QueryParser 用来对查询语句进行语法分析；
- QueryParser 调用 parser 进行语法分析，形成查询语法树，放到 Query 中；
- IndexSearcher 调用 search 对查询语法树 Query 进行搜索，得到结果

1. 第一题 ex1：

- 涉及的文件：

html_v1：存放爬取的 5000 多个网页源代码文件

index_v1：运行 IndexFiles_v1.py 后产生的索引文件夹

IndexFiles_v1.py：实现索引的创建，需要在 python terminal 输入该文件路径才能运行

SearchFiles_v1.py: 实现搜索功能，需要在 python terminal 输入该文件路径才能运行

index_v1.txt: 存放网页文件名 filename 和对应的 url，每行格式为 url +'\t'+ filename

1) 创建文件名->url 的字典：

本实验需要将网页对应的 url 加入到 field 中。即便实现已经创立了文件名 filename 与 url 关联的文件 index6.txt，但如果对于每个文件建立索引时，都用正则表达式在 txt 文件里查询用时太长。因此在索引建立前根据 index6.txt 文件创建 filename->url 的字典 files。建立索引时，只需 files[filename]即是文件对应的 url。该字典在 main()函数里建立，代码如下：

```
f = open(indextxt,'r')
files = {}
for line in f.readlines():
    temp = line.strip().split('\t')
    if len(temp)<=1:
        continue
    files[temp[1]]=temp[0]
f.close()
```

index6.txt 每行内容为"url+\t+filename"，因此 filename=temp[1],url=temp[0]。注意应使用 line.strip()去掉首位的转义字符\n，split('\t')将字符串转换成数组。判断 len(temp)<=1 略去特殊情况，避免报错而使程序终止。

2) 获取网页的编码方式 charset

不同网页的编码方式有 utf-8,gbk,gbk2312 等情况，不尽相同。但网页的编码方式都写在网页源码里，由 charset 指明。如下图所示：

```
<meta charset="UTF-8"/>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

我用正则表达式找出源码里的 charset 信息。charset 的代码格式有上图的两种情况：charset="utf-8"或 charset=utf-8"，相差一个引号。因此我用下图的正则表达

```
charset1 = re.search("charset(?:=\"|=)([^\"]*)", contents)
charset = charset1.groups()[0]
```

式进行筛选：。其中`\"`表示引号本身，`?:`和或运算`|`配合，表示查找开头是 charset="和 charset=的情况。`[^\"]`匹配任何非引号的字符，后面的`*`表示匹配任意多次。下一句`.groups()[0]`指的是`([^\"]*)`，即是我们要找的编码方式。

3) 将网页的 name,path,content,title,url 加入到 document

这部分是代码的重点，涉及到 FieldType 的设计和中文分词。

本次实验涉及两种 FieldType。一种是网页内容 content，需要用来建立索引和创建倒排索引表，需要分词，但最后搜索时不需要打印出来，因此不用完全存储。另一种是文档的文件名 name,路径 path,标题 title,网址 url 的 FieldType。不用来索引，因此不用分词也不用建立倒排索引表。但最后搜索时需要将内容完整呈现出来，因此需要完全存储。两种 FieldType 如下图所示：

```
t1 = FieldType()#文档的文件名name,路径path,标题title,网址url的FieldType
t1.setIndexed(False)
t1.setStored(True)
t1.setTokenized(False)

t2 = FieldType()#文档内容contents2相关的FieldType
t2.setIndexed(True)
t2.setStored(False)
t2.setTokenized(True)
t2.setIndexOptions(FieldInfo.IndexOptions.DOCS_AND_FREQS_AND_POSITIONS)
```

接下来将信息加入到 Field 中。

首先是较为简单的 name 和 path，由于二者在上面的代码已经求出，因此直接用

```
doc.add(Field("name", filename, t1))
doc.add(Field("path", path, t1))
```

即可。

然后加入网页内容。由于网页源码中有许多标签，标签里的内容对于内容查询没有帮助，需要借助 BeautifulSoup 的函数将其去除。如下图所示：

```
soup= BeautifulSoup(contents,"html.parser")
contents2 = ''.join(soup.findAll(text=True))#unicode
```

处理完之后 contents2 是 unicode 类型。但还不能加入到 Field 中，还需要对其中的中文进行分词处理。这里我是用的是中文分词库 finaltag。如下图，第一行

```
seg_list = finalseg.cut(contents2, find_new_word=True)
contents3 = " ".join(seg_list)
```

将待分词的内容 contents2 分解成含一个个词语的列表，再结合" ".join(seg_list)语句即可得到词语间以空格相隔的字符串。此时 contents2 是 unicode 类型，直接用`doc.add(Field("contents",contents3, t2))`将其加入 Field 即可。另外由于中文分词库的使用，中文词语以空格相隔，因此修改用 WhitespaceAnalyzer 代替 StandardAnalyzer。

最后用 soup.find 找到 title,将其设置为 charset 编码，加入到 Field 即可，如下图所示：

```
i1 = soup.find('title')
title = i1.string.encode(charset,"ignore")
doc.add(Field('title',title,t1))
```

至于 SearchFiles_v1.py 和结果展示，与下一题相差不大，待会一起描述描述。

2. 第二题 ex2：

➢ 涉及的文件：

html_v1：存放爬取的 5000 多个网页源代码文件

index_v1：运行 IndexFiles_v1.py 后产生的索引文件夹

IndexFiles_v1.py：实现索引的创建，需要在 python terminal 输入该文件路径才能运行

SearchFiles_v1.py: 实现搜索功能，需要在 python terminal 输入该文件路径才能运行

index_v1.txt: 存放网页文件名 filename 和对应的 url, 每行格式为 url +'\t'+ filename

1) 代码见 IndexFiles_v1.py：

为了实现限制域名 site 的联合搜索功能，我添加了包含 site 的 Field。不同于前两个 FieldType，site 需要建立索引和创建倒排索引表，需要分词，最后搜索时也需要打印出来，因此要完全存储，因此创建第三种 FieldType 如下：

```
t3 = FieldType()#site相关的FieldType
t3.setIndexed(True)
t3.setStored(True)
t3.setTokenized(False)
t3.setIndexOptions(FieldInfo.IndexOptions.DOCS_AND_FREQS_AND_POSITIONS)
```

为了获取域名，我查阅了 python2.7 的官方文档，找到了 urlparse 库的 urlparse 函数具有分析 url 的功能，调用 urlparse.urlparse(url)返回一个字典，其中键 netloc 对应的值即是所求 url 的域名。因此获取域名 site 的代码如下：

```
domain = urlparse.urlparse(files[filename]).netloc
doc.add(Field('site',domain,t3))
```

2) 代码见 SearchFiles_v1.py:

为了实现联合搜索，需要对用户输入的 command 进行分析。因此创建 parseCommand( )函数如下：

```
def parseCommand(command):

    command_dict = {}
    opt = u"contents"
    for i in command.split(' '):
        if ':' in i:
            opt, value = i.split(':')[:2]
            opt = opt.lower()#大写转小写
            if opt ==u"site" and value != '':
                command_dict[opt] = ' '+value#get取回键对应的值
        else:
            command_dict[opt] = command_dict.get('contents', '') + ' ' + i
    return command_dict
```

opt = u'contents' 表示默认用户输入的是对网页内容 contents 的搜索，用户对不同内容的联合搜索需要以空格为间隔。最终返回 opt-->value 的字典。

当接下来的查询就是将每个键值对加入到 BooleanQuery()中。需要注意的是，如果查询的是网页内容 contents，需要对查询的内容进行中文分词，这里同样借助中文分词库 finalseg 实现，代码如下图所示：

```
command_dict = parseCommand(command)
querys = BooleanQuery()
for k,v in command_dict.iteritems():
    if k == u"contents":#如果是对内容搜索，需要分词
        seg_list = finalseg.cut(v, find_new_word=True)
        v = " ".join(seg_list)
    query = QueryParser(Version.LUCENE_CURRENT, k,
                        analyzer).parse(v)
    querys.add(query, BooleanClause.Occur.MUST)
```

- 结果展示：

当在 python terminal 运行 SearchFiles_v1.py，输入“梅西 site:2018.sina.com.cn”，获得部分结果如下图：

```
(base) zsir@zsir-VirtualBox:/media/sf_UbuntuFile/lab4$ /media/sf_UbuntuFile/lab4/SearchFiles_v1.py
lucene 4.9.0

Hit enter with no input to quit.
Query:梅西 site:2018.sina.com.cn
Searching for: 梅西 site:2018.sina.com.cn
50 total matching documents.
-----------------------------------
path: html6/http2018.sina.com.cnarg
title: 阿根廷队_世界杯球队_2018俄罗斯世界杯_新浪体育
url: http://2018.sina.com.cn/arg/
name: http2018.sina.com.cnarg
site: 2018.sina.com.cn
-----------------------------------
path: html6/http2018.sina.com.cn
title: 2018俄罗斯世界杯_新浪体育_新浪网
url: http://2018.sina.com.cn/
name: http2018.sina.com.cn
site: 2018.sina.com.cn
-----------------------------------
path: html6/http2018.sina.com.cnzt_dforefront
title: 俄罗斯最前线_2018世界杯_新浪体育_新浪网
url: http://2018.sina.com.cn/zt_d/forefront
name: http2018.sina.com.cnzt_dforefront
site: 2018.sina.com.cn
-----------------------------------
path: html6/http2018.sina.com.cnnga
title: 尼日利亚队_世界杯球队_2018俄罗斯世界杯_新浪体育
```

3. 第三题(ex3)

- 涉及的文件：

htmlImg_v2：存放爬取的 2000 多个京东商品网页

indexImg_v2：运行 IndexFilesImage_v2.py 后生成的索引文件夹

IndexFilesImage_v2.py：实现索引的创建，需要在 python terminal 输入该文件路径才能运行

SearchFilesImage_v2.py：实现搜索功能，需要在 python terminal 输入该文件路径才能运行

indexImg_v2.txt：存放网页文件名 filename 和对应的 url，每行格式为 url +'\t'+ filename

imgdata_v2.txt：存放每个网页里商品图片的信息，包括图片链接 imgurl、网页链接 url、网页标题 title，每行格式为 imgurl+'\t'+url+'\t'+title+'\n'

IndexFilesImgData2.py：对 htmlImg_v2 文件夹里每个网页提取出商品图片和包含的信息，存放在 imgdata_v2.txt 文件里。这是为了避免每次运行 IndexFilesImage_v2.py 都调用多次的 BeautifulSoup()，提高了运行速度。

1) 代码见 IndexFilesImgData2.py：

为了避免每次运行 IndexFilesImage_v2.py 都调用多次的 BeautifulSoup()，提高运行速度，我们先把图片的信息提取出来，长期存储在 imgdata_v2.txt 文件中，因此运行 IndexFilesImage_v2.py 只需逐行读取该文件，将信息添加到索引文件 Document()即可。

与前两题类似，先逐行读取包含文件名和 url 的 indexImg_v2.txt 文件，得到 filename --> url 的字典，快速获取 url 信息，代码如下。

```
f = open(indextxt, 'r')
files = {}
for line in f.readlines():
    temp = line.strip().split('\t')
    if len(temp) <= 1:
        continue
    files[temp[1]] = temp[0] # filename --> url
f.close()
```

然后遍历 htmlImg_v2 文件夹下的所有网页，用 BeautifulSoup 提取 title、imgurl 和 alt。获取 title 直接使用 title = soup.title.text.strip()即可。若 title==None，该网页不是我们想要的，直接 continue 略过即可。分析商品网页的代码，得到下图所示：

其中可以看见 id 为 spec-img 的即为每个网页的商品图片。因此使用下图代码即可得到 alt 和 imgurl：

```
for linkline in soup.findAll('img',{"id":re.compile('spec-img')}):
    imgurl = linkline.get('data-origin')
    alt = linkline.get("alt")
    if imgurl == None or alt == None:
        continue
```

最后将所得信息添加到 imgdata_v2.txt 即可：

```
with open("imgdata_v2.txt",'a') as f2:
    line = imgurl+'\t'+url+'\t'+alt+"\t"
    f2.write(line.encode('utf-8','ignore')+'\n')
```

2) 代码见 IndexFilesImage_v2.py：

由于已经提取出了所有需要的信息，因此不必再遍历 htmlImg_v2 里的网页，直接逐行读取 imgdata_v2.txt 即可。FieldType 有两种，t1 和 t2，与前两题相同。因此 txt 文件的每行 line 进行 linelist = line.split('\t')操作，imgurl = linelist[0]，url = linelist[1]，alt = linelist[2]。需要注意的是，alt 需要支持搜索，FieldType 为 t2，也需要中文分词，此处借助中文分词库 jieba，代码如下：

```
seg_list = jieba.cut(alt)
altTokened = " ".join(seg_list)
```

而 imgurl，url，title 设置 FieldType 为 t1 类型即可。

3) SearchFilesImage_v2.py：

需要对 command 进行中文分词，其他代码和前两题几乎一致。

➢ 结果展示

当在 python terminal 运行 SearchFilesImage_v2.py，输入“雀巢”，获得部分结果如下图：

```
(base) zsir@zsir-VirtualBox:/media/sf_UbuntuFile/lab4$ /media/sf_UbuntuFile/lab4/SearchFilesImage_v2.py
Building Trie...
Trie has been built succesfully.
lucene 4.9.0

Hit enter with no input to quit.
Query:雀巢
Searching for: 雀巢
雀巢
40 total matching documents.
-------------------------------------
imgurl: //img11.360buyimg.com/n1/jfs/t22150/38/1669742126/193148/49160aee/5b305cb7N3b66c572.jpg
url: https://item.jd.com/1501844651.html
urltitle: 雀巢（Nestle） 雀巢Nestle 超级能恩超启能恩婴幼儿奶粉 800g/ 3段
-------------------------------------
imgurl: //img11.360buyimg.com/n1/jfs/t15028/7/2234985019/372274/8b6c45fd/5a7d1dafN6aa7f986.jpg
url: https://item.jd.com/255751.html
urltitle: 雀巢（nestle）AL110婴幼儿无乳糖营养配方粉
-------------------------------------
imgurl: //img14.360buyimg.com/n1/jfs/t1/56310/4/8613/163668/5d60a927Ec07f10ae/76bb48a44b2ce5a7.jpg
url: https://item.jd.com/1404987399.html
urltitle: 雀巢（Nestle） 腹泻配方粉雀巢能恩AL110无乳糖营养配方粉400g/克 *1罐
-------------------------------------
imgurl: //img14.360buyimg.com/n1/jfs/t1/35398/24/14986/157816/5d299107E286c7724/ca4a4987e6c2f0cc.jpg
url: https://item.jd.com/1565839109.html
urltitle: 雀巢（Nestle） 雀巢超级能恩超启能恩婴儿奶粉德国原装进口婴儿幼儿奶粉（新老包装随机发货） 1段800g
-------------------------------------
```

三、实验总结

1. 实验概述：

通过分析网页提取所需信息，利用 Lucene 的使用对所需信息建立倒排索引表，并借助中文分词库和实现中文信息检索和联合检索的功能。

2. 实验心得

本次实验，我收获颇丰，主要有以下几点：

1) 熟悉了建立索引，搜索索引的过程，学会了利用 lucene 建立倒排索引表并检索所需信息。
2) 学会了如何利用 lucene 实现联合检索功能
3) 学会了如何将爬虫爬取的网页进行分析、提取信息，以实现信息检索功能

3. 实验创新点

1) 在 ex1，ex2，ex3 中事先读取存有 filename 和 url 的 txt 文件，将其建立一个具有 filename--->url 的字典，避免每次获取某个网页的 url 都要用正则表达式搜索 txt 文件，提高了文件运行速率。
2) 新建了 IndexFilesImgData2.py 文件，对 htmlImg_v2 文件夹里每个网页提取出商品图片和包含的信息，存放在 imgdata_v2.txt 文件里。避免了每次运行 IndexFilesImage_v2.py 都要对某一网页源代码文件调用多次的 BeautifulSoup()，反而只需逐行读取 imgdata_v2.txt 文件，将信息添加到索引文件 Document() 即可，大大提高了运行速度。

最后，衷心感谢实验中老师和各位助教的帮助!