**电工导实验报告3**

F1403023 5140309534 韩坤言

**一、实验目的**

1. **了解**索引的创建和搜索，中文分词
2. 了解组合查询，更新索引中的文档，简单的图片搜索

**二、实验内容**

1. 了解Lucene，索引创建和搜索索引，实现中文的全文检索
2. 组合查询来对搜寻的结果进行限制，更改目录中的文档以及搜索网页中出现的图片。

**三、实验环境**

1. **Firefox + Firebug插件或Chrome**
2. **Python 2.7 + easy\_install + BeautifulSoup**
3. **JCC + PyLucene**

**四、实验步骤**

Lucene是一个高效的，基于Java的中文检索库。全文索引主要分为两个部分，索引的创建和利用索引来进行搜索。原文先被分词，得到的词元传给语言处理组件，最后将一个个term传给索引组件，得到倒排索引。而用户查询时的关键词先被分词，创建语法树，进而搜索。对中文而言，先用中文分词软件分好词再用WhitespaceAnalyzer或者SimpleAnalyzer进行分析是比较方便而且可行的。

例如Google等搜索引擎，都能在搜索时加上限定，比如site:某某网站，更好符合用户的需求。通过使用BooleanQuery可以将不同的查询组合成复杂的查询方式。BooleanQuery本身是一个布尔子句的容器，这个子句可以是逻辑“或”、“与”或“非”。可以用BooleanQuery的add方法将一个查询子句添加到某个BooleanQuery对象中，这样能够优化搜寻的结果，得到更令人满意的搜索结果。图片搜索要依据图片周围的文字来获取关于图片的信息，用这些信息来建立索引，最后依据这进行查找。

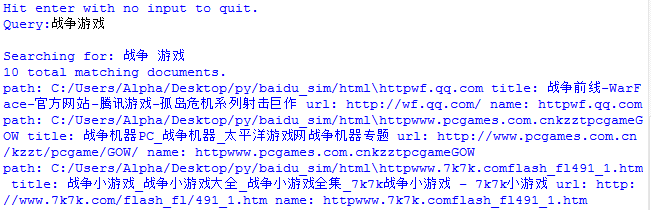
**五、问题及其解决**

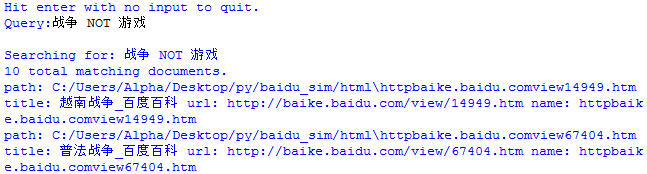
**1. 实现一个中文网页索引与搜索程序**

爬取一定数量（>5k）的中文网页，修改IndexFiles.py和SearchFiles.py，对这些中文网页建立索引。

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





原以为这并不是什么难事，但是问题接踵而至。

首先，我发现我之前的crawl\_thread爬如此大量的网站时经常会爬着爬着就突然停了下来，既没有终止程序，也没有任何的报错信息。有时候甚至到499X的时候停下来，让人心烦不已。我不得不进一步修改我的程序。考虑到很多文件比如pdf,doc这些都不是需要的，而且这些文件一般爬的很慢，我于是将把得到的url后缀是这些的都过滤掉了，爬取网页的速度提升了不少。我还加了一些异常捕获，尽量减少发生错误的可能性。

其次是汉字的编码问题，虽然我查阅了大量关于编码转换和机理，但是遇到有的网站还是无能为力，毫无办法，乱码还是乱码，但是大部分的网站都能正常处理，这个问题有待之后解决。

随着网页越爬越多，最后速度会越来越慢，我原以为是crawled里的元素多了以后，判断在不在crawled中是占用时间的元凶，但是就算我加上了hash来优化，依旧没有好转，到最后100至200左右速度会大幅下降，反而用了哈希以后，平均速度反而降低了，我还是保留了原来的方法。

爬5000个网页大约用时如下：

C:\Users\DELL-PC\Desktop\QQ截图20151028145904.png

大约26分钟，可以接受。

修改index的创建程序不是很难，无非改一改路径，我运用了第三方的库chardet来识别网页的编码方式，把其解码成unicode的格式，因为unicode可以跨平台运用，实用最方便，也不容易出错。bs自带的识别编码方式的函数不太理想，安装了chardet库以后，遇到不确定的情况还会调用chardet模块，所以我直接用了chardet模块。但又是仍然有无法处理的情况，毕竟少数，加个异常捕获跳过就行了。读取爬网页时的index.txt，找到对应url的文件名，然后再打开，读取，分词，把html的tag去除。开始我打算实用nltk库，但是高版本貌似不支持这个函数了，我于是选择了使用bs自带的get\_text()函数。虽然效果不是非常好，但是减少了查询时所不必要的时间。Search的程序也是相应改动，改动不大。5000个做成index还是花了不少时间。查询时效果不错。

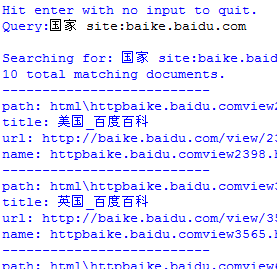
C:\Users\DELL-PC\Desktop\QQ截图20151028154839.png

用时挺长，或许不把title print出来时间会短一些。

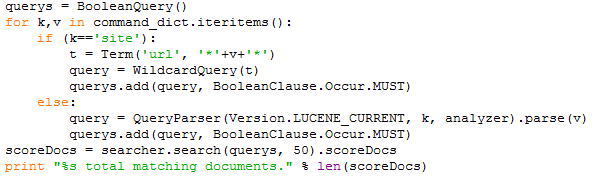


测试图片

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

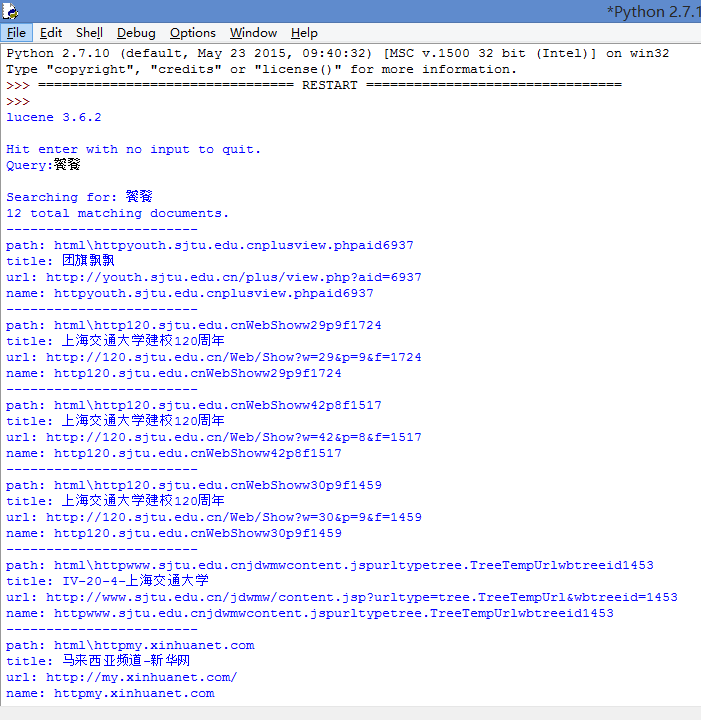


开始我是用tld的第三方模块来得到一个url的域名，把域名加到index里之后进行匹配。但是这样得到的域名只是顶级的域名，遇过搜索的时候用下级的域名变无法搜索到。后来我修改search的程序



通过WildcardQuery可以部分来进行匹配，顺利解决了分级域名无法匹配的问题。

我的测试：



选了一个比较少见的词，全网站搜索有12个符合的

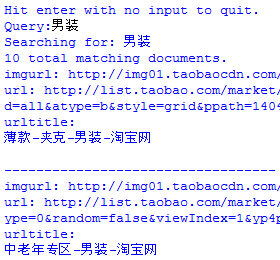


限定了sjtu后就只有5个了，site的限制颇有成效。

**3. 实现一个图片索引**

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：

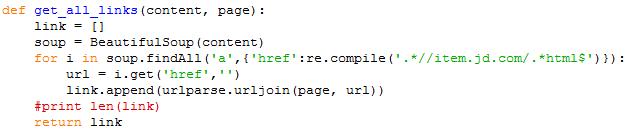


提示：图片周围的文本可能会用到parser实验中的parent, nextSibling，previousSibling等函数。

做图片索引时最好选定某个网站爬取。比如只对淘宝网站上的图片进行索引，这样可以对特定网站的结构进行分析，让搜索结果更精确。

我是选择京东，因为我发现爬京东比淘宝快多了。首先要修改crawler，因为我需要爬取那些商品的界面，并不需要别的没有用的网页。

我添加了正则匹配，因为商品的url都有共同的部分



图片好找，但是我还需要关于图片的描述用来匹配搜索。我剖析了京东商品页面的html树

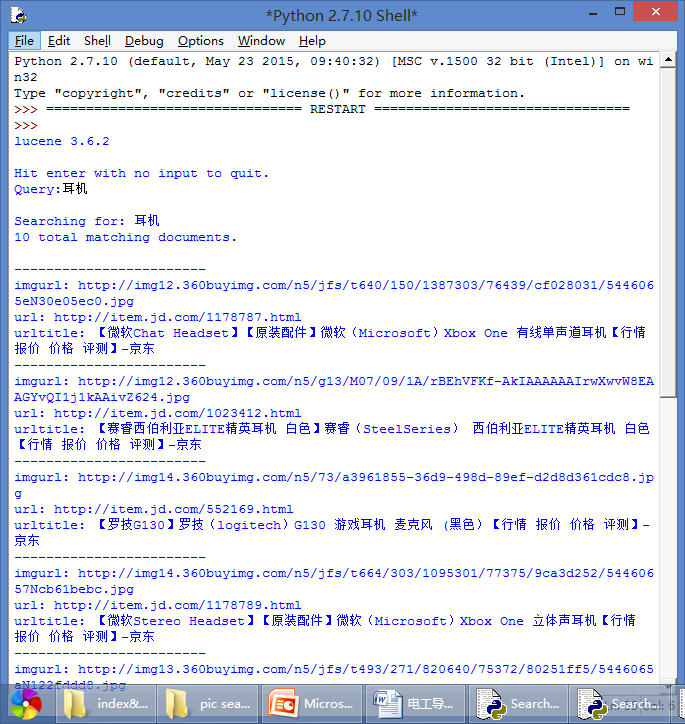


主要相关的图片和描述都在这个叫p-box的tag里，虽然还有很多商品的图片，那些图片反正爬取的时候会爬到是它们的界面，所以就不处理了。下面商品介绍虽然有很多图片，但没有相应的描述，而且图片相关性参差不齐，所以就算了，我主要把左上角一副大图和下面数量不等的小图存下来，这些都是相关性极高的图片，也有详尽的描述。

这部分代码我是这样处理的



由于京东商品的页面结构是固定的，这样能大幅减少出错的概率，尝试很多次基本没有问题。做成index时描述是要分析的，别的信息只要存储即可。



小样本进行测试，结果令人满意，如果样本再大一些，产品更加多样的话，效果会更好（这些网页都和数码产品相关，所以大部分都是这些商品）

**六、实验总结**

这次在上次作业的基础上，难度提升不少，但是实用性也越来越有所提升。index的建立和关键词的搜索大大提升了搜索的效率，也能对大量爬到的网站进行处理，为我所用（否则爬了一堆网站什么都做不了并没有什么用）。之后的site的限制来是搜索结果更能让人满意，以及对图片的搜索，都有一种一步步在做搜索引擎的感觉，总而言之，通过这两周的学习让我受益匪浅，对此也有了兴趣。