电类工程导论(C 类)实验 6&7 报告

贾萧松 516030910548

一、实验概述

Lab6&7 中，主要结合的 HTML,CSS,lucene 的知识点，根据前面用爬虫爬取过的网页，运用 web.py 制作一个简单的文字加图片搜索引擎

二、实验环境

Ubuntu14.04+python2.7+lucene 4.9.0+jieba+web.py+beautifulsoup

三、实验内容

实验要求制作一个搜索引擎，我主要借鉴了 bing 和百度页面的一些特点来设计我的文字搜索页面，借鉴京东页面的一些特点来设计我的图片搜索页面。

1.网页组织与数据传递

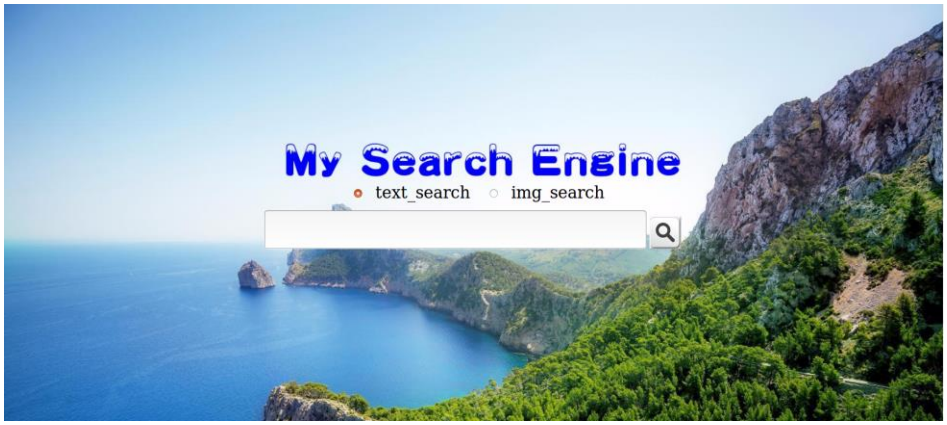
这方面，我的思路比较简单，就是让用户在 index 页面完成搜索内容的输入(keyword)和要搜索文字还是图片(search_kind)，然后跳转到 searching 类，在该类

内对要 keyword 进行分词处理，然后根据 search_kind,调用相应的 searcher 进行查询，最后传入结果生成对应的网页。

```
urls = (
    '/', 'index',
    '/searching', 'searching'
)

render = web.template.render('templates')  # your templates

class index:

    def GET(self):
        return render.index()
```

```
class searching:

    def GET(self):
        # lucene
        user_data = web.input()
        if(user_data.keyword == ""):
            return render.index()
        keyword = user_data.keyword.decode('utf-8', 'ignore')
        command = " ".join(jieba.cut_for_search(
            keyword, HMM=True))

        if(user_data.search_kind == "text"):
            return render.text(text_search(command), keyword, get_recommand(keyword))
        else:
            return render.img(img_search(command), keyword)
```

2. Index.html

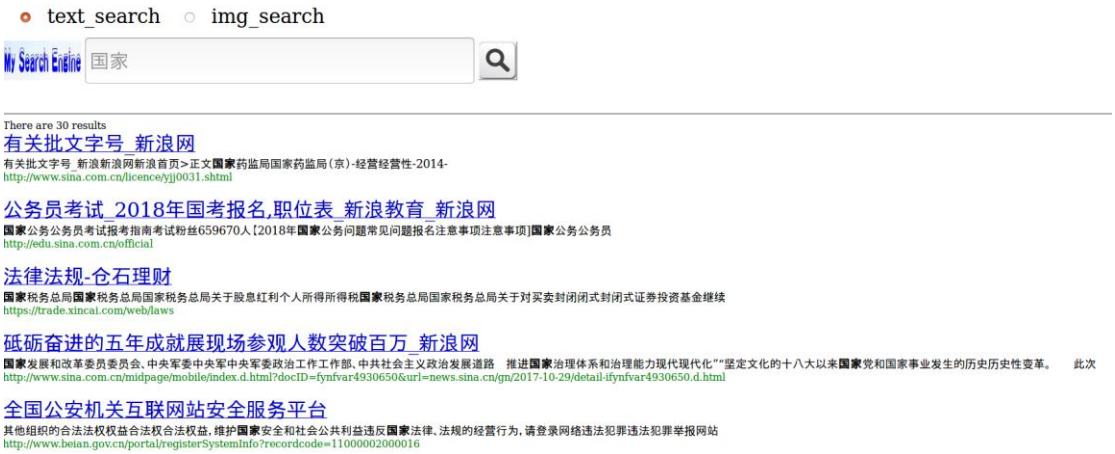


如上图，该页面主要参照了一些 bing 的元素，如搜索框左边的小放大镜，及搜索框上方供用户选择的按钮。在这个网页的设计中，我觉得最费时的一点就是令组件水平垂直都居中，经过网上的学习与测试，找到一种比较简洁的方法，这种方法不需要像其他方法一样需要设置比较多的参数，而是利用 table 标签中有的 align 和 valign 属性直接设置。

```
<table width="100%" height="100%" align="center">
<tr>
<td align="center" valign="middle">
```

至于在设置背景图过程中的遇到的曲折，我将在问题与解决部分细说。

3. text.html



如图，在这个网页的设置中我主要参考了百度搜索的内容，例如：上面的 logo 可以返回 index 页，字体的大小与颜色…

由于需要提取关键词上下文，这个我在 lucene 中找到了便捷的库 highlighter，它可以根据 keyword 来匹配在 content 中最符合的一段文字。关于如何定义合适，我采用了它内置的 QueryScorer 定义。它还可以设定输出的格式，由于内置的 SimpleHTMLFormatter 作为输出已经很好用了，我也就没有修改，代码如下

```
query_highlight = QueryParser(Version.LUCENE_CURRENT, k,
                              envir.analyzer).parse(command_dict["imgtitle"])
myhighlighter = Highlighter(
    SimpleHTMLFormatter(), QueryScorer(query_highlight))
myhighlighter.setTextFragmenter(SimpleFragmenter(50))
for scoreDoc in scoreDocs:
    # find texts which are around the keyword
    doc = envir.img_searcher.doc(scoreDoc.doc)
    text = doc.get("imgtitle")
    key_text = "".join((myhighlighter.getBestFragment(
        envir.analyzer, "imgtitle", text)))
```

除此之外，注意到百度网页还有相关搜索的功能，一开始我查到的 lucene 中有一个叫 suggest 的库可以实现这个功能。由于 lucene 的说明文档是 java 的而且这个库又太过高级远远超过需求，我根据说明文档调试，调了很久也没有实现对应的功能。我决定换一种方法，在网上查询后，找到一种取巧的办法，直接调用百度的 api 来找到相关内容。代码如下(正好用到了关于 html 表单的知识)：

```
def get_recommand(word):
    User_Agent = 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.81 Safari/537.36'
    url = "https://www.baidu.com/s?wd="
    word = urllib.quote(word.encode('utf-8', 'ignore'))
    url += word
    req = urllib2.Request(url)
    req.add_header("User-Agent", User_Agent)
    content = urllib2.urlopen(req)
    soup = BeautifulSoup(content, 'html.parser')
    goal = soup.find('div', {'id': 'rs'})
    res = goal.get_text(" ").split(" ")
    return res[1:]
```

4. img.html

由于在之前的 Lab 中我做的是京东图片的爬取，所以我做这个页面主要参考了京东的搜索界面。

完成这个页面的要点就是，对图片的排版。

令图片可以从左向右排列并自动换行在 html 中有属性 float:left 可以直接完成。但这还不够，因为原图片的大小可能并不一致，其次下方的文字行数可能不一致，这都会造成排版混乱。解决办法就是规定好每个组件应占大小，然后让图片和文字适应标签。最终效果就是这样。

四、问题与解决

1. JVM 报错问题

这个问题就是因为在运行文件时和在网页被打开时，都会运行到这里，解决办法就是另建一个 py 文件储存这些变量，在主文件中 import 这个文件就可以解决。在这里，由于图片和文字搜索的 index 不一样，所以需要创建两个 directory 和 searcher：

```
vm_env = lucene.initVM(vmargs=['-Djava.awt.headless=true'])
text_directory = SimpleFSDirectory(File("text_index"))
img_directory = SimpleFSDirectory(File("img_index"))
text_searcher = IndexSearcher(DirectoryReader.open(text_directory))
img_searcher = IndexSearcher(DirectoryReader.open(img_directory))
analyzer = WhitespaceAnalyzer(Version.LUCENE_CURRENT)
```

到这里，就不得不提一下 Python 中的一个有趣的地方了。在 Python 的使用过程中，有时候会需要模块间的全局变量，这时候的常见做法就是把全局变量放到一个独立的模块中，使用时，导入此全局变量模块即可。这时候就有一个问题，就是如果使用 import 的话这个全局变量就是公共的；而如果使用 from import 的话，就相当于在该模块创建了一个全局变量的副本（初值相同），而修改这个变量，并不会被其他模块知道。这既是一个小坑，又可以帮助我们理解 import 和 from import 的差别。

2. 关于 web.py 的本地文件问题

由于我想为 index 页设置一个背景图，却发现通过相对路径寻找：无论是假设以 code.py 为起点还是假设以 index.html 为起点来设置，始终都无法找到对应的文件。经过思考与猜测，我认为可能由于 web.py 是跑在本地服务器上的，所以它判定的起点可能在 web.py 的代码那里，这个要寻找无疑是很麻烦的。经过上网查询，了解到作者设置了如果在 code.py 目录下建一个名字为 static 的文件夹，那么只需 /static/文件名 就可以找到那个本地文件。

3. 关于京东商品价格的读取

关于京东商品价格的获取，在京东的单个产品页面上，通过查看源码检查 html，发现没有直接给出价格。这是因为价格数据是通过 JS 动态加载的，而 HTML 源码中并不包含动态加载的页面内容。

于是为了解决这个问题，我找到京东关于价格的 api：

```
price_url = "https://p.3.cn/prices/mgets?skuIds=J_" + str(number)
```

然后用 json 加载：

```
price_js = json.loads(urllib2.urlopen(price_url).read())
price = price_js[0]['p']
```

然后就得到了商品价格。不过由于京东商品更新比较快，爬取到的不少商品已经下架，就没有价格了。

五、总结

在 Lab6&7 中，我们完成了对以往学过的 HTML、爬虫、lucene 知识的回顾，同时学习了如何用 div+CSS 来设计一个网页（确实用去了我大量的时间来美化界面），使用 web.py 框架来制作了一个简单的文字、图片搜索引擎。

最后，衷心感谢老师和助教们的精心准备和辛勤付出，你们整理完善的 ppt 和到位的答疑大大提高了我学习的效率，感谢~

贾萧松 516030910548
2017.10.31