电类工程学导论 C 实验报告 6

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：本实验主要使用了迷你型 web 框架 web.py，以及 lucene 中用于网页上下文查找和高亮显示的库 highlight
2. 实验目的
   1) 使用 web.py, 结合前面学习的 HTML, Lucene，中文分词等知识点，根据上次实验爬取的网页，建立一个简单的搜索引擎。

结果界面：

搜索界面：

3. 实验原理：
   本实验通过使用 web.py 框架实现 url 访问、网页请求，并结合上个实验的中文信息索引知识，在网页显示检索结果，以实现一个简单的搜索引擎功能。

二、实验过程

1. 建立索引得到索引文件夹 indexWeb：
   由于之前的实验不要求打印网页内容或获取网页内容上下文，因此没有在索引文件中对网页内容 contents 进行存储。所以这次修改 IndexFilesWeb.py 文件将 contents 对应的 Field 设置为 setStored(True)。并且用 contents2 = soup.get_text()提取网页源代码的文本内容。然后运行

IndexFilesWeb.py 文件得到索引文件夹 IndexWeb

2. 创建 SearchFilesWeb.py 以针对搜索目标返回结果集：

SearchFilesWeb.py 文件基于上次的搜索索引的 py 文件进行修改。这次不再直接打印出搜索结果，而是返回一个结果列表 res_list，里面每一个元素是一个搜索得到的网页信息字典，分别有 title，url，highlight 等信息。如下图：

```
for i, scoreDoc in enumerate(scoreDocs):
    doc = searcher.doc(scoreDoc.doc)
    temp = {}
    temp["url"] = doc.get("url")
    temp["title"]= doc.get("title")
    temp["site"] = doc.get("site")
    print doc.get("contents")
    temp["highlight"]= highlighter.getBestFragment(analyzer, "contents",doc.get("contents"))
    res_list.append(temp)
return res_list
```

为了得到搜索目标在网页中的上下文，我查找了相关文档，找到了 lucene 里的 highlight 库可以实现这一功能。先引入需要用到的库：

```
from org.apache.lucene.search.highlight import Fragmenter,Highlighter,QueryScorer,SimpleHTMLFormatter,SimpleFragmenter
```

然后使用库中函数进行高亮匹配，代码如下，每行代码意义已经写在注释里：

```
#构建Formatter格式化最终显示(将字体颜色设置为斜体红色),_blank_用于后面替换成空格
simpleHTMLFormatter = SimpleHTMLFormatter("<i><font_blank_color='red'>", "</font></i>")
#构造FieldQuery
queryHlt = QueryParser(Version.LUCENE_CURRENT, "contents", analyzer).parse(command_dict['contents'])
#构建Scorer,用于选取最佳切片
scorer = QueryScorer(queryHlt)
#实例化Highlighter组件
highlighter = Highlighter(simpleHTMLFormatter, scorer)
#默认字符为100
highlighter.setTextFragmenter(SimpleFragmenter(100))
```

随后使用 getBestFragment 得到最佳切片即可得到搜索目标在网页中的上下文。

另外由于各个模块需要最终在 web1.py 中运行，因此 SearchFilesWeb.py 仅包含两个函数而不能独立运行，函数参数也有些许改变。

3. 修改网页模板 formtest.html 和 result.html：

formtest.html 与原来相同，没有变化，都是通过 render 传入一个表单 login，然后传递\s，打开网页 result.html

而 result.html 有较大改动。首先 result.html 通过 py 文件 web1.py 传入三个参数 command，res_list,searcher，分别指搜索目标 command，通过 SearchFilesWeb.py 得到的搜索结果列表 res_list，和搜索索引使用的搜索器 searcher。所以针对 res_list 里的每个元素显示出其 title、url 和 highlight 即可。代码如下图：

```
<h1>Search Result for "$command"</h1>
    <br><br>
    $for i in res_list:
        <a href="$i['url']"><font size = '4'><i>$i['title']</i></font></a>
        <br>
        $if i["highlight"] !=None:
            <br>
            $:i["highlight"].replace(' ','').replace('_blank_',' ').split("trydocument")[0].split("windowSTO")[0].strip("function")
            <br>
        <font color="Chartreuse"><i>$:i["url"]</i></font>
        <br><br><br>
```

其中最长的那行代码是为了解决 soup.get_text()不能将某些无用信息剔除的问题，replace('_blank_',' ')替换成空格，使字体能够正常设置。

另外作为创新点，我还在 result.html 里添加了搜索框，避免搜索一次就必须返回 formtest.html 的麻烦。代码如下：

```
<form action="/s" method="GET">
    Google:<input type="text" name="Google">
    <input type="submit" name="search" value="Search">
</form>
```

4. 创建 web1.py 以串联各个模块实现功能：

最后创建 web1.py 串联各个模块实现功能。添加下述代码以启动虚拟机，并在 Class index 和

```
try:
    vm_ev = lucene.initVM(vmargs=['-Djava.awt.headless=true'])
except:
    vm_env = lucene.getVMEnv()
```

Class s 里添加 vm_env.attachCurrentThread() 防止线程报错以及实现在 result.html 里继续搜索。除此之外 Class index 不变，但 Class s 中添加了索引初始设置和获取索引结果的代码，如下图，每行解释已写在注释中：

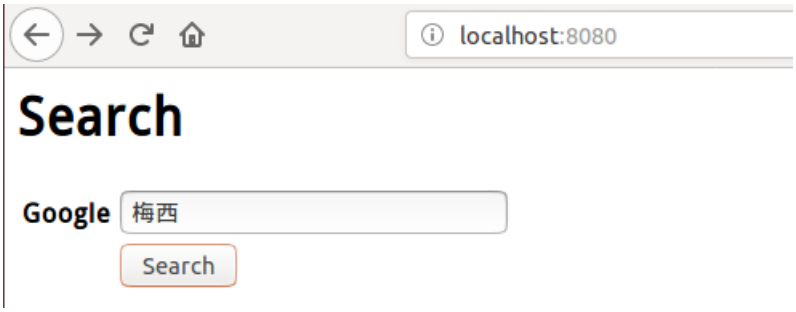
```
class s:
    def GET(self):
        vm_env.attachCurrentThread()
        STORE_DIR = "indexWeb"#索引文件夹位置
        directory = SimpleFSDirectory(File(STORE_DIR))
        searcher = IndexSearcher(DirectoryReader.open(directory))#建立搜索器
        command = web.input().Google#获取索引目标
        #以下两行通过运行SearchFilesWeb.py里的函数得到索引结果command_dict
        command_dict = SearchFilesWeb.parseCommand(command)
        res_list = SearchFilesWeb.run(command,command_dict,STORE_DIR)

        return render.result(command,res_list,searcher)
```

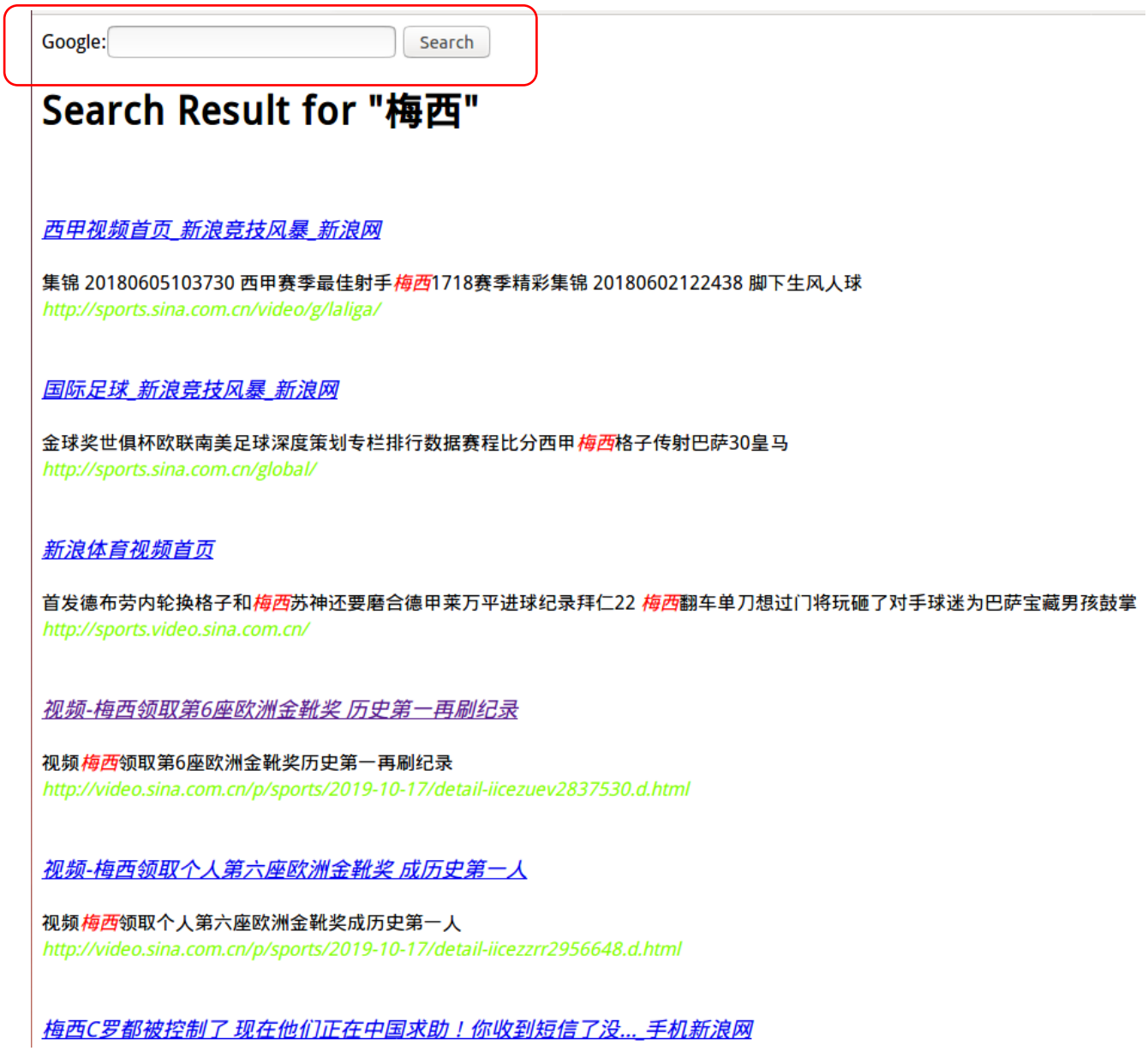
5. 结果展示：

在 terminal 运行 web.py，当出现如下图显示时即可在浏览器输入 http://localhost:8080,得到一个搜

```
(base) zsir@zsir-VirtualBox:/media/sf_UbuntuFile/lab6$ python /media/sf_UbuntuFile/lab6/web1.py
http://0.0.0.0:8080/
```



索首页。当在搜索框中输入足球相关信息时，即可得到结果。比如当我输入“梅西”，可以得到下图：



三、实验总结

1. 实验概述：

   本实验通过使用 web 框架 web.py 创建网页，从而将前面实验的搜索索引以网页的方式呈现。

2. 实验心得

   本次实验，我收获颇丰，主要有以下几点：

   1) 了解了 web 框架 web.py，学会了创建网页的基本方式
   2) 学会了如何在 html 文件里嵌入 python 语句
   3) 再次熟悉了 html 的相关知识，比如标签的书写，字体、颜色等的设置。

3. 实验创新点

   1) 在 result.py 中创建搜索框（即是上图的红色框部分），可以直接在 result.html 页面再次搜索而不用回到 formtest.py。
   2) result.py 中设置了字体的变化，如红色、绿色和斜体等。

最后，衷心感谢实验中老师和各位助教的帮助!