> 同济大学数据采集与集成实验报告04

《爬取天气预报数据 》

- 学院：电子信息工程学院
- 专业：数据科学与大数据专业
- 学号：1851804
- 姓名：苗成林
- 指导教师：郭玉臣
- 时间：2021.11.02

实验报告正文

实验目的及要求

1. 了解 HTML 文档结构
2. 了解深度优先算法和广度优先算法
3. 掌握网站遍历和数据采集方法
4. 掌握 BeautifulSoup 用法

实验原理

BeautifulSoup

1. Beautiful Soup提供一些简单的、python式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。
2. Beautiful Soup自动将输入文档转换为Unicode编码，输出文档转换为utf-8编码。你不需要考虑编码方式，除非文档没有指定一个编码方式，这时，Beautiful Soup就不能自动识别编码方式了。然后，你仅仅需要说明一下原始编码方式就可以了。

爬虫广度优先算法

整个的广度优先爬虫过程就是从一系列的种子节点开始，把这些网页中的”子节点”(也就是超链接)提取出来，放入队列中依次进行抓取。被处理过的链接需要放 入一张表(通常称为Visited表)中。每次新处理一个链接之前，需要查看这个链接是否已经存在于Visited表中。如果存在，证明链接已经处理过， 跳过，不做处理，否则进行下一步处理。

初始的URL地址是爬虫系统中提供的种子URL(一般在系统的配置文件中指定)。当解析这些种子URL所表示的网页时，会产生新的URL(比如从页面中的http://www.admin.com ”中提取出http://www.admin.com 这个链接)。然后，进行以下工作：

1. 把解析出的链接和Visited表中的链接进行比较，若Visited表中不存在此链接，表示其未被访问过。
2. 获取子链接。
3. 处理完毕后，将链接直接放入Visited表中。
   针对这个链接所表示的网页，继续上述过程。如此循环往复。

广度优先遍历是爬虫中使用最广泛的一种爬虫策略，之所以使用广度优先搜索策略，主要原因有三点：

1. 重要的网页往往离种子比较近，例如我们打开新闻网站的时候往往是最热门的新闻，随着不断的深入冲浪，所看到的网页的重要性越来越低。

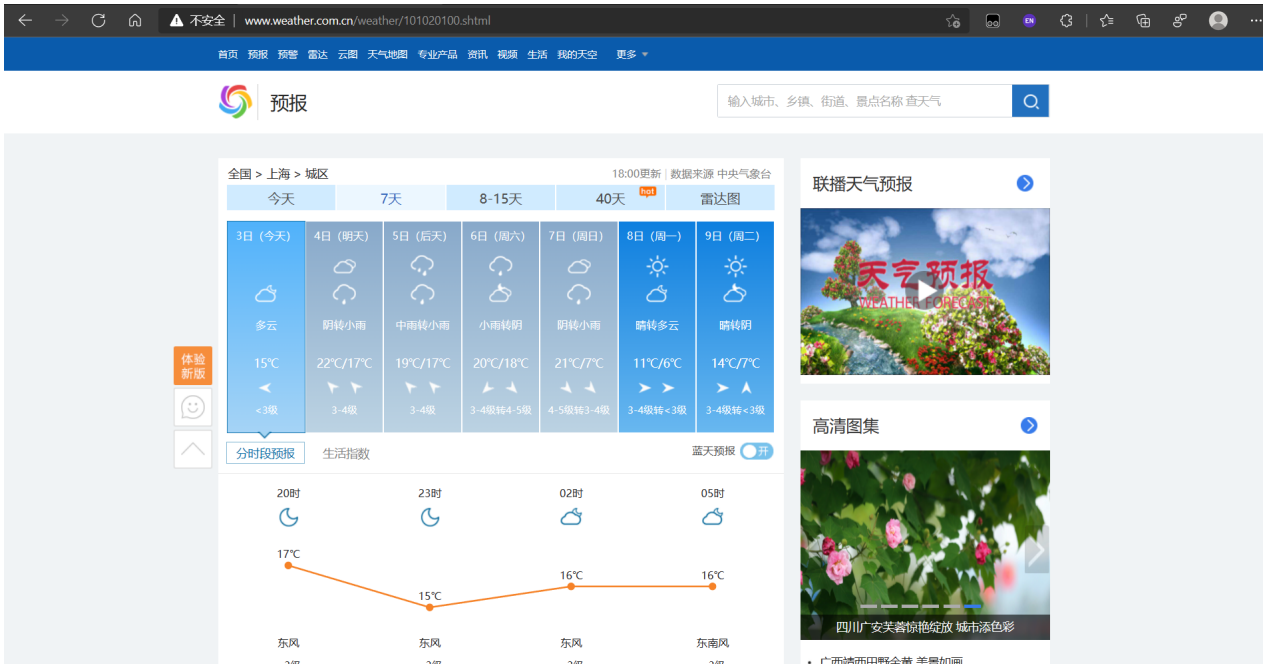
2. 万维网的实际深度最多能达到17层，但到达某个网页总存在一条很短的路径。而广度优先遍历会以最快的速度到达这个网页。
3. 广度优先有利于多爬虫的合作抓取，多爬虫合作通常先抓取站内链接，抓取的封闭性很强。

爬虫深度优先算法

深度优先搜索是一种在开发爬虫早期使用较多的方法。它的目的是要达到被搜索结构的叶结点(即那些不包含任何超链的HTML文件) 。在一个HTML文件中，当一个超链被选择后，被链接的HTML文件将执行深度优先搜索，即在搜索其余的超链结果之前必须先完整地搜索单独的一条链。深度优先搜索沿着HTML文件上的超链走到不能再深入为止，然后返回到某一个HTML文件，再继续选择该HTML文件中的其他超链。当不再有其他超链可选择时，说明搜索已经结束。优点是能遍历一个Web 站点或深层嵌套的文档集合；缺点是因为Web结构相当深，，有可能造成一旦进去，再也出不来的情况发生。

实验过程

中国天气网上海页面



网页部分源代码如下

```
<ul class="t clearfix">
<li class="sky skyid lv3 on">
<h1>9日（今天）</h1>
<big class="png40 d02"></big>
<big class="png40 n07"></big>
<p title="阴转小雨" class="wea">阴转小雨</p>
<p class="tem">
<span>26</span>/<i>18℃</i>
</p>
<p class="win">
<em>
<span title="南风" class="S"></span>
<span title="西南风" class="SW"></span>
</em>
<i><3级</i>
</p>
```

```
<div class="slid"></div>
</li>
<li class="sky skyid lv1">
<h1>10日（明天）</h1>
<big class="png40 d01"></big>
<big class="png40 n02"></big>
<p title="多云转阴" class="wea">多云转阴</p>
<p class="tem">
<span>27</span>/<i>18℃</i>
</p>
<p class="win">
<em>
<span title="东风" class="E"></span>
<span title="南风" class="S"></span>
</em>
<i><3级</i>
</p>
<div class="slid"></div>
</li>
<li class="sky skyid lv2">
<h1>11日（后天）</h1>
<big class="png40 d02"></big>
<big class="png40 n07"></big>
<p title="阴转小雨" class="wea">阴转小雨</p>
<p class="tem">
<span>31</span>/<i>20℃</i>
</p>
```

爬虫基本流程

1. 用到的库：requests，BeautifulSoup，re。其中，requests库用于获取网页内容，BeautifulSoup用于网页解析，re正则表达式库用于对爬取内容进行匹配和搜索。
2. 正常情况下BeautifulSoup就可以完成网页解析，但是中国天气网现在的风向条目变成了两种风向，是在不知道怎么用BeautifulSoup去解析，只好调用re库把两个风向给检索出来。
3. 不同城市有不同城市的代码，可以根据城市的不同替换URL就好。

网页请求

```
def get_page(url):
    try:
        kv = {'user-agent':'Mozilla/5.0'}
        r = requests.get(url,headers = kv)
        r.raise_for_status()
        r.encoding = r.apparent_encoding
        return r.text
    except:
        return '错误'
```

正则表达式网页解析

```
def parse_page(html, return_list):
    soup = BeautifulSoup(html, 'html.parser')
    day_list = soup.find('ul', 't clearfix').find_all('li')
    for day in day_list:
        date = day.find('h1').get_text()
        wea = day.find('p',  'wea').get_text()
        if day.find('p', 'tem').find('span'):
                hightem = day.find('p', 'tem').find('span').get_text()
        else:
                hightem = ''
        lowtem = day.find('p', 'tem').find('i').get_text()
        #win = re.search('(?<= title=").*?(?=")',
str(day.find('p','win').find('em'))).group()
        win = re.findall('(?<= title=").*?(?=")', str(day.find('p','win').find('em')))
        wind = '-'.join(win)
        level = day.find('p', 'win').find('i').get_text()
        return_list.append([date, wea, lowtem, hightem, wind, level])
    #return return_list
```

数据格式化存储

```
def print_res(return_list):
    tplt = '{0:<10}\t{1:^10}\t{2:^10}\t{3:{6}^10}\t{4:{6}^10}\t{5:{6}^5}'
    print(tplt.format('日期', '天气', '最低温', '最高温', '风向', '风力',chr(12288)))
    for i in return_list:
        print(tplt.format(i[0], i[1],i[2],i[3],i[4],i[5],chr(12288)))
```

EXCEL表格展示

上海未来七天天气数据						
3日（今天）	阴	9℃		东风	<3级	
4日（明天）	多云转小雨	10℃	19℃	北风-东北风	<3级	
5日（后天）	多云	8℃	20℃	东北风-西风	<3级转4-5级	
6日（周六）	小雨转雨夹雪	0℃	9℃	西风-西风	4-5级转3-4级	
7日（周日）	晴转多云	-2℃	6℃	西风-东南风	3-4级转<3级	
8日（周一）	多云转晴	-1℃	9℃	西南风-西风	<3级转3-4级	
9日（周二）	多云转晴	0℃	11℃	西风-西风	3-4级转<3级	

提取当前网页全部链接

```
def getlink(url):
    headers = ("User-Agent", "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36")
    opener = urllib.request.build_opener()
    opener.addheaders = [headers]
    urllib.request.install_opener(opener)
    file = urllib.request.urlopen(url).read()
    file = file.decode('utf-8')
    pattern = '(https?://[^\s)";]+(\.(\w|/)*))'
    link = re.compile(pattern).findall(file)
    #去重
    #link = list(set(link))
    return link
```

提取到的链接包括

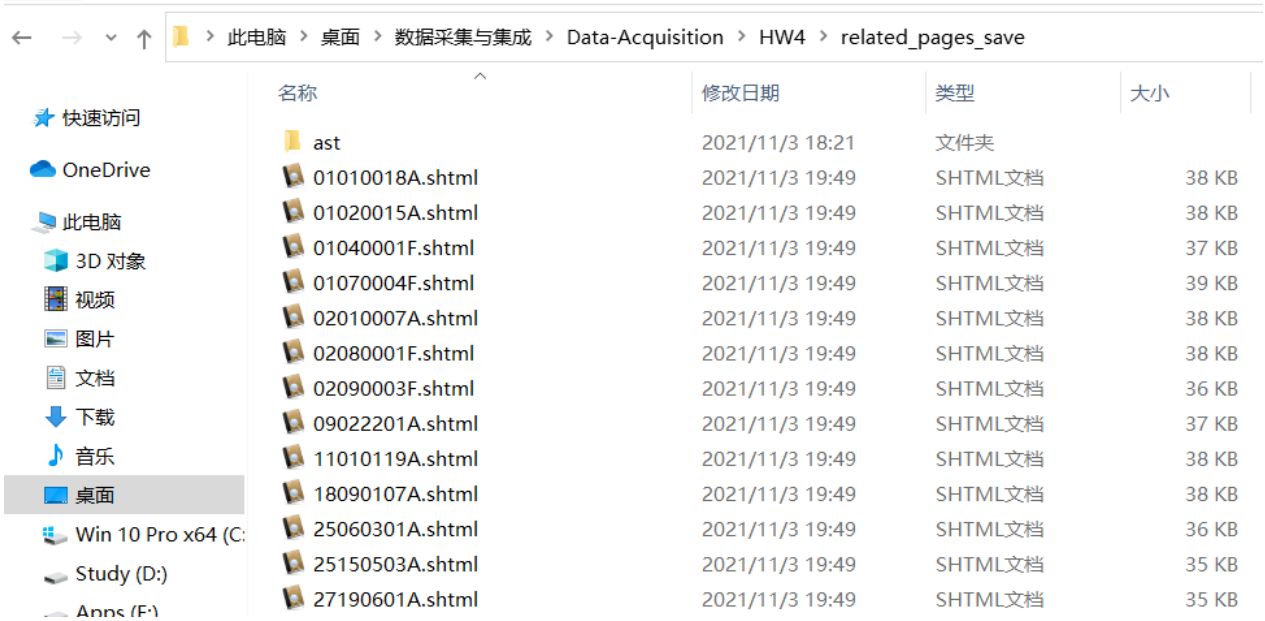
```
http://www.weather.com.cn/weather1d/101010100.shtml
http://www.weather.com.cn/weather1d/101020100.shtml
http://www.weather.com.cn/weather1d/101270101.shtml
http://www.weather.com.cn/weather1d/101210101.shtml
http://www.weather.com.cn/weather1d/101190101.shtml
http://www.weather.com.cn/weather1d/101030100.shtml
http://www.weather.com.cn/weather1d/101280601.shtml
http://www.weather.com.cn/weather1d/101040100.shtml
http://www.weather.com.cn/weather1d/101110101.shtml
```

```
http://www.weather.com.cn/weather1d/101280101.shtml
http://www.weather.com.cn/weather1d/101120201.shtml
http://www.weather.com.cn/weather1d/101200101.shtml
http://www.weather.com.cn/weather1d/10101010018A.shtml
http://www.weather.com.cn/weather1d/10130051008A.shtml
http://www.weather.com.cn/weather1d/10118090107A.shtml
http://www.weather.com.cn/weather1d/10109022201A.shtml
http://www.weather.com.cn/weather1d/10101020015A.shtml
http://www.weather.com.cn/weather1d/10127190601A.shtml
http://www.weather.com.cn/weather1d/10102010007A.shtml
http://www.weather.com.cn/weather1d/10125150503A.shtml
http://www.weather.com.cn/weather1d/10111010119A.shtml
```

获得以上链接对应页面

```python
for link in linklist:
    if link[1]=='.shtml':
        url=link[0]
        response=requests.get(url=url)
        page_text=response.text
        # encode编码，将ISO-8859-1编码成unicode
        page_text = page_text.encode(response.encoding)
        # decode解码,将unicode解码成utf-8
        page_text = page_text.decode("utf-8")

        with open ('related_pages_save/'+link[0][-15:],'w',encoding='utf-8') as f:
            f.write(page_text)
```



问题改进

爬虫得到的数据存储后打开发现是乱码，需要更改存储的编码格式，进行Encode和Decode编码转换

```
response=requests.get(url=url)
page_text=response.text
# encode编码，将ISO-8859-1编码成unicode
page_text = page_text.encode(response.encoding)
# decode解码,将unicode解码成utf-8
page_text = page_text.decode("utf-8")
```

参考文献

(44条消息) 爬虫之广度优先&深度优先_飞星恋的博客-CSDN博客

(44条消息) BeautifulSoup详解_大数据 小白学习录-CSDN博客

Python爬虫从入门到放弃（十）之 关于深度优先和广度优先 - syncd - 博客园 (cnblogs.com)

(44条消息) python+pandas 保存list到本地excel_liuzh的博客-CSDN博客_python将list写入excel

(44条消息) python爬虫的BeautifulSoup库详解*一行玩python-CSDN博客*beautifulsoup详解