**爬取有道词典**

**1.工作概要**

因为之前从来没有写过爬虫，也没有过相关的学习，所以动起手来迟迟没有进展，本

来这个作业已经做好接近一个月了却一直没有提交，究其原因，还是自己做的太简单，跟前面的同学比起来很惭愧。主要就用了beautiful soup和request先去某英语学习网站爬取了单词表，然后再用此表在有道词典检索，并爬取了它的例句，解释等，所以相当于写了两个爬虫，由于单词较少，加之有道词典并没有反爬取措施，所以像多线程、IP代理池、数据库我都没用emmmm（- -；），不过目前也在跟实验室里做另一个网站的爬虫，后面再慢慢改进吧

**2.爬取单词表**

2.1爬取LIST的url集合

首先我确定了要爬取的目标六级高频词汇，于是找到某网页：

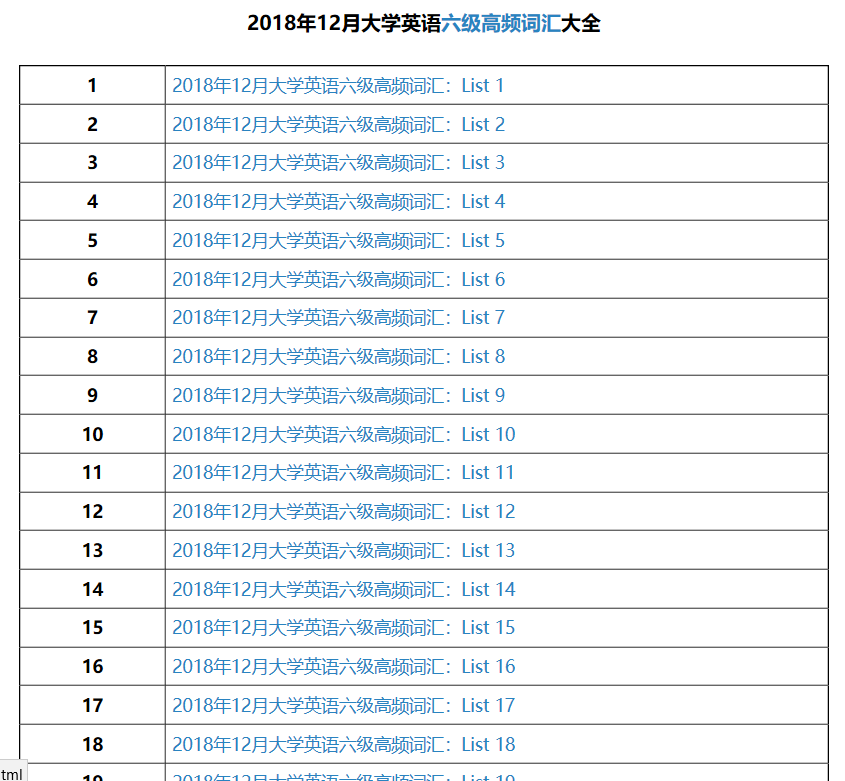


图2-1 六级词汇表下载截图

它一共有40多个LIST，每个LIST大概50个单词，正好用它组成词汇表

分析它的网页结构：入口是上图，然后我要先爬取所有LIST的url，再根据url去对应的LIST表里面下载相应数据：

首先查看HTML文件，发现需要的内容在td标签中截取了部分内容：

|  |
| --- |
| <td height="30" width="18%"><div align="center"><strong>1</strong></div></td>, <td height="30" width="82%"><a href="http://cet6.koolearn.com/20180927/822885.html" target="\_blank">2018年12月大学英语六级高频词汇：List 1</a></td>, <td height="30"><div align="center"><strong>2</strong></div></td>, <td height="30"><a href="http://cet6.koolearn.com/20180927/822886.html" target="\_blank">2018年12月大学英语六级高频词汇：List 2</a></td>, <td height="30"><div align="center"><strong>3</strong></div></td>, …… |

所以提取里面的url使用如下代码：

|  |
| --- |
| **""" 初始化，得到单词列表的url """ def** getListUrl():  url = **r'http://cet6.koolearn.com/20180927/822884.html'** html = getWordHtml(url)  wordSoup = BeautifulSoup(html, **"lxml"**)  *#找到<td height="30" width="82%"><a href="http://cet6.koolearn.com/20180927/822885.html" target="\_blank">  #他返回一个匹配td height="30"的列表，其中有我们要的href* a = wordSoup.find\_all(name=**"td"**, height=**"30"**)  LIST = []  i = 0  *#并不是每个td height="30"都包含一个href  # 每隔一行是一个url所以用line.a['href']取值* **for** line **in** a:  **if** ((i % 2) == 1 **and** i < 98):  LIST.append(line.a[**'href'**])  i += 1  **return** LIST |

其中getWordHtml(url)方法是根据url返回一个下载的相应html文件

|  |
| --- |
| **""" 去查找一个单词列表并下载它的单词 输入：单词列表 返回：XML格式的文件 """ def** getWordHtml(url):  *#修改head信息，模仿成手机浏览器* headers = {  **'User-Agent'**: **r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '  r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3'**,  **'Referer'**: **r'http://www.lagou.com/zhaopin/Python/?labelWords=label'**,  **'Connection'**: **'keep-alive'** }  *#发出请求并下载得到HTML文件* req = request.Request(url, headers=headers)  page = request.urlopen(req).read()  **return** page.decode(**'utf-8'**) |

2.2爬取每一个LIST列表里面的单词：

打开对应网页：

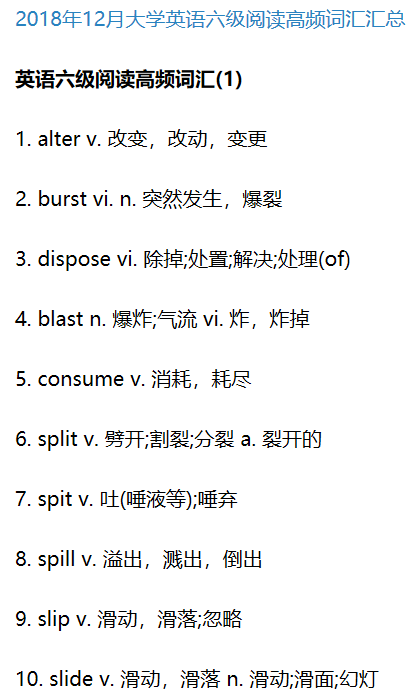


图2-2 单词列表界面

查看源码节选知道内容在<div class="mt40">标签里面：

|  |
| --- |
| <div class="show\_l2">  <div class="mt40">  <p>　　<a class="link" href="http://cet6.koolearn.com/" target="\_blank">新东方在线英语六级</a>频道为大家整理了</a></p><p><strong>　　英语　account 帐(目、户);叙述，说明</p><p>　　claim 声称，断言;索取</p><p>　　decline 下降;谢绝</p><p>　　potential 潜在的;潜力</p><p>　　promote 促进;提升</p><p>　　avoid 避免</p><p>　　challenge 挑战</p><p>　　current 流;当前的;潮流，趋势</p><p>　……　maintain 维持;保养</p><p>　　range 范围;山脉</p><p>　　survey 眺望;调查</p><p>　　aware (of)意识到的</p><p>　　community 共同体;社区，团体</p><p>　　concern 关心，挂念;涉及</p><p>　　contribute 贡献;有助于;投稿</p><p>　　feature 特征;面貌</p><p>　　ignore 不顾，不理，忽视</p><p>　　impact 冲击，碰撞;影响，效果</p><p>　　pursue 追踪;从事;追求</p><p>　 </div> |

于是爬取内容：

|  |
| --- |
| **""" 收集单词 输入：单词列表的全部url 返回：单词列表的List集合 """ def** reWordList(List):  RelistWord = []  *#将LIST列表的url依次取出，并通过上面的getHTML函数来得到LIST列表里面的内容* **for** url **in** List:  list1 = []  html = getWordHtml(url)  *# 创建BeautifulSoup对象* wordSoup = BeautifulSoup(html, **"lxml"**)  *# 搜索文档树,找出div标签中class为mt40的所有子标签，每一个子标签里面含有一个单词的结构* a = wordSoup.find\_all(name=**"div"**, class\_=**"mt40"**)  **for** line **in** a:  *#将div标签里所有的字符串遍历* **for** string **in** line.stripped\_strings:  list1.append(repr(string)) *#["'新东方在线英语六级'", "'频道为大家整理了'","'英语六级高频词汇'",  # ..."'：List 1'", "'comprehension 理解'","'access 接近，进入;通道，入口'", "'account 帐(目、户);叙述，说明'"...]* list1 = list1[10:-1] *#减去前面10个和后面1个多余的  #再将单词和解释切分开来如access 接近，进入;通道，入口，只要access* **for** elem **in** list1:  elem = elem.split(**" "**)  a = elem[0][1:]  RelistWord.append(a)  **return** RelistWord |

3.3对文本进行清洗

一般来说爬下来的单词是【apple 苹果】这样可以直接切分的，但在运行中频繁报错发现有一些是【apple苹果】（这里e和苹之间没有空格）或【apple①苹果】所以对数据还要再清洗一遍并写入单词表：

|  |
| --- |
| ListUrl = getListUrl() a = reWordList(ListUrl) *#写入单词到words.txt* file = open(**'words.txt'**, **'w'**, encoding=**'utf-8'**) pat = **'[a-zA-Z]+'** *#正则表达式全英文* **for** i **in** a:  *#由于a里的单词爬下来有些单词和汉字在一起没切开，所以使用re.findall匹配所有英文* new\_str = re.findall(pat, i) *#找到全英文的列表* **if**(new\_str != []):  a = new\_str[0]  a = re.sub(**'[^\w\u4e00-\u9fff]+'**, **''**, a)*#将使用正则表达式过滤非中英文字符因为比如apple①后面带个①* file.write(str(a)+**"\n"**) print(**"完成"**) |

**3.爬取有道词典**

有了单词表后，就可以去有道词典爬注释等了，即依次取单词表中的所有单词得到他们的HTML文件，然后选取有需要的内容：

首先是单词查询器：没查一个单词返回对应xml文件

|  |
| --- |
| **""" 去有道词典查找一个单词并下载它的HTML 输入：需要查询的单词 返回：XML格式的文件 """ def** getYoudaoHtml(word=**'bless'**):  url = **r'http://dict.youdao.com/search'** data = {**'le'**: **'eng'**, **'q'**: word, **'keyfrom'**: **'dict'**}  *#加上head，模仿成手机浏览器* headers = {  **'User-Agent'**: **r'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) '  r'Chrome/45.0.2454.85 Safari/537.36 115Browser/6.0.3'**,  **'Referer'**: **r'http://www.lagou.com/zhaopin/Python/?labelWords=label'**,  **'Connection'**: **'keep-alive'** }  data = parse.urlencode(data).encode(**'utf-8'**)  req = request.Request(url, headers=headers, data=data)  page = request.urlopen(req).read()  **return** page.decode(**'utf-8'**) |

接着分析返回回来的文件：

发现单词的解释在**'div'**, class\_=**"trans-container"**这个标签下的<li>标签下，例子在**'div'**, class\_=**"examples"**这个标签下，所以依次提取并写入就ok了。

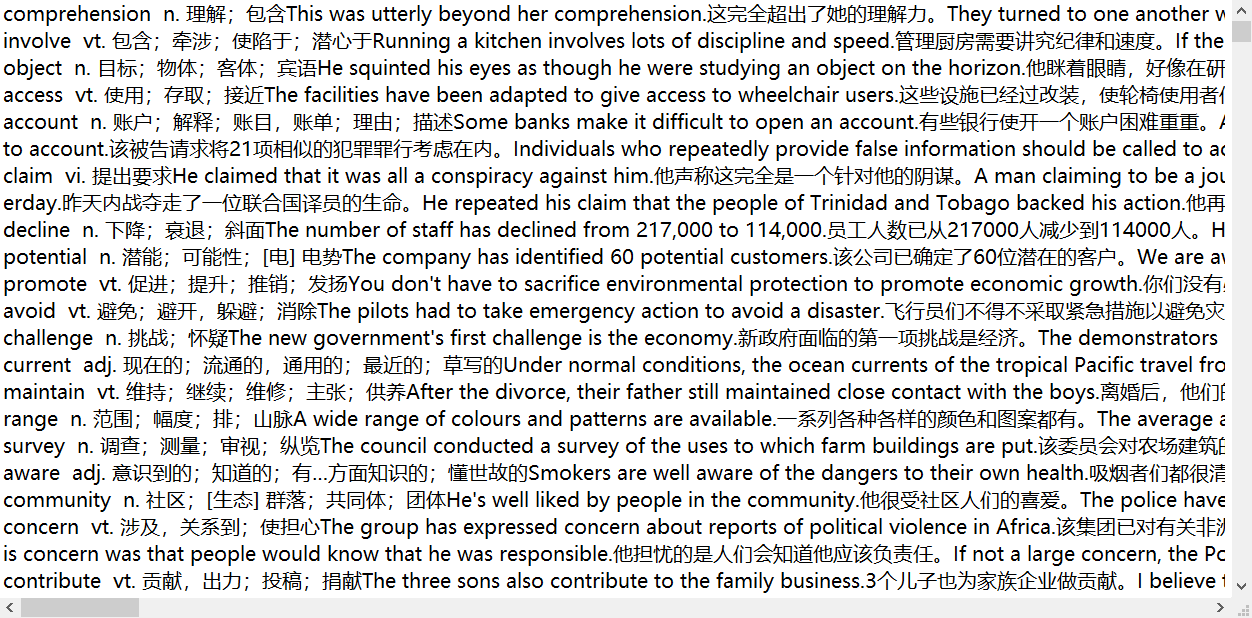
|  |
| --- |
| **""" 去查找并写入一个单词的爬取内容，并记录到DownData.txt 输入：word - 需要查询的单词 """ def** WriteOneWord(word=**'application'**,file=**''**):  wordSoup = BeautifulSoup(getYoudaoHtml(word),**"lxml"**)  writestr = word+**" "** exsitif = wordSoup.find(**'div'**, class\_=**"trans-container"**) *#先判断能不能查到单词，查不到会返回None* **if**(exsitif **is not None**):  meaning = wordSoup.find(**'div'**, class\_=**"trans-container"**).li.string  writestr = writestr + str(meaning)  example = wordSoup.find\_all(**'div'**, class\_=**"examples"**)  **for** line **in** example:  **for** string **in** line.stripped\_strings:  writestr = writestr +str(string)  file.write(writestr + **"\n"**) |

执行main函数：

|  |
| --- |
| List = loadWords(**'words.txt'**) file = open(**'DownData.txt'**, **'w'**, encoding=**'utf-8'**) i = 1;size = len(List); **for** word **in** List:  WriteOneWord(word,file)  **if**(1):  print(**"i:{} word:{} 进度:{:.2%}"**.format(i,word,i/len(List)))  i += 1 print(**"完成"**) |

其中loadWords(**'words.txt'**) 是加载第二部下载好的单词表。

**4.结果及改进**



程序本身爬取的数据量比较少，如果数据量大的话可以加多线程，然后爬的这两个网站都没有反爬虫机制，所以后面还可以加入动态IP代理池或其他同学提到的一些反爬虫机制等，然后自己的数据处理也比较简单，不利于大量数据的分析和管理后面可以加上mongDB或redis等等一些数据库。

总之，我也是还在学习爬虫中，期望后面有时间了再多多改进一下吧。

**5.参考**

<https://cuiqingcai.com/1319.html>

<https://blog.csdn.net/c406495762/article/details/71158264?utm_source=blogxgwz0>