简述利用 Python 网络爬虫实现多下载站软件搜索及下载地址提取

罗楷轩
北京市第四中学 高三（1）班 北京 100088

摘 要 在浩瀚的互联网中如何准确地查找到免费和共享的软件，是一个费时、费力的过程。本文利用Python网络爬虫技术，提供了在常用的下载站点中，快速、有效地过滤大量无用、广告链接，从而准确捕捉真实下载地址的方法。

关键词 Python网络爬虫；下载站；下载地址

1 设计目的及意义

Windows操作系统以其软件数量与种类之丰富著称，然而对于广大网民而言，下载电脑软件是一件颇费心思的事情。究其原因，乃各大下载站广告、弹窗、重定向繁多，且设计出多个类似的下载按钮，迷惑性极强。很多带有“高速下载”字样的下载按钮无一是所需软件的直链，实际会触发下载“下载器”（一个1MB不到的exe文件）[1]。进而一不留神，会带来潜在的被安装捆绑软件的风险。

现利用Python网络爬虫技术，可以实现多个下载站软件“搜索-下载”一体化的操作，解决下载软件的难题。

2 运行流程示例

以搜索关键词“网易”获取“网易邮箱大师”的下载地址为例，程序运行效果如下：

请选择下载站：

[1] 太平洋软件下载中心 [2]pc6下载站 [3]统一下载站

键入：1

请输入软件名：

键入：网易

{‘[1] title’：‘网易UU网游加速器 2.5.5 正式版’，

‘info’：‘网易UU网游加速器是一款免安装、免注册、免登录、完全免费的加速器软件.....’}

{‘[2] title’：‘网易MuMu(安卓模拟器) 1.0.4.0 官方版’，

‘info’：‘网易MuMu安卓模拟器（安卓模拟器），是网易官方推出的精品游戏服务平台.....'}

……

{‘[10] title’：‘网易POPO 2.0.2269 完整版’，

‘info’：‘网易泡泡是由网易公司开发的一款免费的绿色多媒体即时通信工具.....’}

请选择您要下载的软件的数字编号：

键入：3

下载地址如下：

http://dlc2.pconline.com.cn/filedown_61922_8759928/k3AUdmmn/mail.exe

3 实现过程与代码分析

（以太平洋软件下载中心“http://dl.pconline.com.cn/”为例，中文括号中为简单化的Python代码）

3.1 运行环境

Windows 10, Python 3.6, BeautifulSoup4库,requests库, Chrome

3.2 对软件列表页面的操作

（1）通过分析得出软件列表网址为“http://ks.pconline.com.cn/download.shtml?q=*&downloadType=%C8%ED%BC%FE%CF%C2%D4%D8”，其中*处为所输关键词。在关键词中，对于带空格的文字，空格用“+”号代替（.replace(‘ ‘, ‘+’)）；对于中文字符，需使用GB2312编码而非UTF-8编码，并且编码后“\x”字符替换为“%”（.encode(‘gb2312’).replace(‘\\x’,’%’)）。

（2）利用requests和BeautifulSoup爬取网页（soup = BeautifulSoup(requests.get (url_begin).text, ‘lxml’)）。

（3）利用Chrome右键“检查”复制出每一款软件名和软件网址所在标签的selector（soup.select(‘selector’)），将其存入一个list之中。

（4）此处应注意的是，如若软件列表为空（即所搜索软件未收录），那么“#Jwrap>div.main> div > div”标签中应能搜索到“抱歉，没有找到与 ‘’ 相关的结果”字样，此时应重新输入关键词，以避免后续程序报错。

3.3 对软件页面的操作

（1）对3.2中的（2）获取到的软件网址进行爬取。

（2）拷贝软件介绍所在标签的selector，获取其文本（.get_text()），替换\n,\r,\t等特殊字符（.replace(‘\n’,’’).replace(‘\r’,’’).replace(‘\t’,’’)）。

（3）分析页面可知，软件介绍有长有短，考虑到输出时格式的美观与可读性，将每一个软件介绍的前70个字符添加到一个list之中（infos.append(info_select [:69])）。

（4）逐条打印软件名称、简介等信息，注明每一条对应的序号，便于用户筛选所需软件。

（5）分析发现，软件网页与下载网页的网址构成之间存在一定的规则。分割软件网址（.split(‘’)[]），接上下载按钮（其指向软件下载网页）所在标签的相关内容

（‘&linkPage=1.html’，’-1.html’等），存储到一个list当中。至此，完成了下载网址的获取。

3.4 对软件下载页面的操作

（1）对软件下载页面进行爬取。

（2）多次试验发现，有部分软件并不能找到最终的下载链接，这大概是网站数据库本身的问题，但无论如何，应排除此种错误。（if str(soup_choose.select（‘selector’)) != ‘[]’: … else: …）。

（3）下载页面中的下载链接有电信和联通两个选项，URL有所不同，因此笔者在程序中添加了获取用户所属运营商的对应下载地址的代码。仔细研究页面，发现“您的IP是 ***** 建议选择 ** 下载”字样，据此判断网站自身有获取用户IP地址和所属运营商的相关代码。

3.5 对其余页面的操作

（1）鉴于从下载页面的html中查不到IP和运营商的信息，于是利用Chrome“检查”工具进行搜索，发现IP地址和运营商是通过“http://whois.pconline.com.cn/ipJson.jsp?callback=checkComm&callback=checkComm”这一网址获取到的。然而下载页面中只有电信和联通两个运营商的选项，对于移动和网通用户来讲，他们的选择是不清楚的。为此，继续分析下载网页，查找运营商字样，发现在“http://js.3conline.com/pconline/2015/dl/js/client.js”这一JavaScript之中隐含了网通和移动用户的“福音”：

```
if(/网通/.test(location) || /联通/.test(location)){
……
    typeDL=1;}
else{
……
typeDL=2;}
```

这说明网通和联通用户建议使用“本地联通”的链接下载，而除此以外的运营商使用

电信的链接。结合下载页面爬出的xml，便可得出“最终”的下载地址。

（2）之所以最终二字要打引号，因为如这般得到的下载地址仍无法下载。将其与浏览器访问同一网址得到的下载地址做一比对，发现少了一个由八位字符串组成的层级，这个字符串并不能在基本html中找到，而是储存在另一JavaScript“http://dlc2.pconline.com.cn/dltoken/****_genLink.js”当中。直接爬取或用浏览器直接访问此JS，结果只能得到一串固定的字符：“iaMs0RrY”，乍一看没看明白，仔细一瞧，原来是英文“对不起”的意思。看来，下载站开发者有意做了反爬虫的措施，使得直接访问这个至关重要的JS行不通。于是再次利用Chrome，分析浏览器按正常流程访问获得的此JS的RequestsHeaders，发现其中的Referer正是下载页面的网址本身。这下真相大白，将Referer加入到对“_genlink.js”爬虫的Headers之中，如此发送Get指令（requests.get(url_js, headers=headers)），终于得到了梦寐以求的八位字符串。将前期获得的“伪”下载地址切割开，在中间加入从JS中获取的字符串，便可组成最终的下载地址。

以上是获取“太平洋软件下载中心”软件下载地址的基本思路，统一下载站和PC6下载站的思路与之大同小异，不再赘述。

4 部分源代码

```
choose = input(‘请选择您要下载的软件的数字编号：\n’)
downurl_choose = downurl[int(choose)-1]
wb_data_choose = requests.get(downurl_choose)
soup_choose = BeautifulSoup(wb_data_choose.text, ‘lxml’)
    prepared = soup_choose.select(‘#Jwrap> div.area.sc-1>……>span.links-wrap > a’)
carrier_url = ‘http://whois.pconline.com.cn/ipJson.jsp?callback=checkComm&callback=checkComm’
wb_data_carrier = requests.get(carrier_url)
soup_carrier = BeautifulSoup(wb_data_carrier.text, ‘lxml’)
    if str(prepared) != ‘[]’:
        if (str(soup_carrier).find(‘网通’) != -1) or (str(soup_carrier).find(‘联通’) != -1):
downurl_final = str(prepared).split(‘,’)[3].split(‘\”’)[5]
        else:
downurl_final = str(prepared).split(‘,’)[0].split(‘\”’)[5]
    else:
        print(‘暂无下载地址’)
        return
url_js = soup_choose.find(‘script’, {‘src’: re.compile(‘http://dlc2.pconline.com.cn/dltoken/[0-9]+_genLink.js’)}).get(‘src’)
    headers = {
        “Referer”: “{}”.format(urls[int(choose)-1].get(‘href’))
    }
wb_data_js = requests.get(url_js, headers=headers)
soup_js = BeautifulSoup(wb_data_js.text, ‘lxml’)
    code = str(soup_js).split(‘\’’)[1]
downurl_ultimate = downurl_final.rsplit(‘/’, 1)
downurl_ultimate = downurl_ultimate[0]+’/’+code+’/’+downurl_ultimate[1]
    print(‘下载地址如下：\n’ + downurl_ultimate)
```

将写好的三个常用下载站的代码分装在三个Python文件之中，在主程序中，用户输入下载站代号，分别调取相应文件中的函数，以实现不同下载站的搜索和下载功能。

5 需要注意的地方

（1）Python不支持以tag:nth-child(x) 的结构描述selector，应替换为tag:nth-of-type(x)。

（2）注意特殊字符与转义字符的处理。

（3）太平洋软件下载中心的下载地址并非永久直链，每隔一段时间会失效，需重新运行程序获取。

本文提出的快速查找公开免费软件下载地址的方法，在实践中证明是非常实用和有效的。进一步可用pyInstaller将Python程序与它所依赖的库打包成exe文件，便于未安装Python和相应库的广大用户使用。

参考文献

[1] Ryan Mitchell著.陶俊杰,陈小莉译. Python 网络数据采集[M].北京:人民邮电出版社,2016:53-54.

（上接第35页）

间添加Sleep（20）来实现延时。

（3）接收数据，根据通信协议要求，需要对接收的状态命令帧数据进行校验处理，如果接收数据正确则进行数据解析。首先将接收的数据存储于字符型数组中recv Buff中，调用CRC校验函数cal_ccitt16（unsigned char*buffer，unsigned int len）对接收数据的8 byte数据位进行校验，将通过CRC校验算法计算出的两字节的校验位与接收帧数据中的校验位进行比较，如果一致则进行解析处理，否则，丢弃数据。串口接收数据的处理过程都在串口字符接收消息WM_COMM_RXCHAR的响应函数中。

3 结束语

本文基于VC++多线程技术，结合串口通信技术和自定义简单串口通信协议编写了适用于双机通信的串口通信程序，从中可以了解到计算机串口通信程序的编写方法。

参考文献

[1] 李景峰,杨丽娜,潘恒,等.VisualC++串口通信技术详解[M].北京:机械工业出版社,2010:76-105,145-165.

[2] 龚建伟,熊光明.VisualC++/TurboC串口通信编程实践[M].北京:电子工业出版社,2004:87-160,197-217.

（上接第36页）

据多元化服务理念将各种信息及时输送到用户桌面。常见的信息服务内容是为用户提供移动、嵌入式以及云计算等服务，这些服务内容与人们的工作生活密切相关，用户的使用频率较高，能够帮助革新科技情报工作者为需求群体提供更多优质服务，并完善科技情报机构工作体系。从服务方向来看，要想在“互联网+”视野下与时俱进，科技情报系统必须要创新理念、拓展空间、丰富内容并延长其服务链，强化对情报系统的建设，不断提升自身的服务能力[3]。

3 结束语

在信息化浪潮下，科技情报工作的各节点都面临着挑战。在“互联网+”视野下，为了让科技情报工作跟上时代发展步伐，需要相关工作人员树立以服务为中心的工作理念。并根据“互联网+”的思维创新其发展方式，为各类主体提供多元化、高质量的信息情报服务，向支撑科技决策和科技创新的智库方向发展。

参考文献

[1] 胡笑梅,刘帅.大数据环境下中小企业竞争情报服务模式研究[J].湖北经济学院学报(人文社会科学版),2016,13(01):58-59.

[2] 余波.互联网时代基层科技情报工作刍议[J].合作经济与科技,2013,(07):32-33.

[3] 刘军,牛争艳.科技文献共享服务平台科技创新决策分析服务系统的研究与应用[J].情报科学,2013,(8):81-83.