文章编号：2096-1472(2016)-02-43-04

基于Python的动态网页评价爬虫算法

夏火松，李保国
（武汉纺织大学管理学院，湖北 武汉 430073）

摘 要：在大数据获取中面临着如何采集动态评论网页的问题，这篇论文使用静态网页信息构造动态链接，提出了基于Python的动态网页评论爬虫算法。在此基础上实现了评论收集程序。最后将它与通用爬虫算法进行比较，证实了该算法具有针对性强、数据采集速度快、易嵌入开发、简单等优点，为不善于编程的新闻、文学、管理等学科的研究者提供了快速获取评论信息的方法。

关键词：Python语言；静态地址；动态链接；动态网页评论；爬虫算法
中图分类号：TP312 **文献标识码**：A

Crawler Algorithms of Dynamic Web Reviews Based on Python

XIA Huosong,LI Baoguo
(*School of Management*,*Wuhan Textile University*,*Wuhan* 430073,*China*)

Abstract: An issues in big data is:how to get a dynamic comment page? This paper uses information of static pages structure dynamic link and designs a crawler algorithm for dynamic web.On this basis,this paper implements a comment collector.Finally,this paper compares it with the general crawler algorithm.It is proved that this algorithm has the advantages of strong pertinence,fast data acquisition,easy to be embedded,simple and so on.It provides fast access to large data sources for researchers who are not proficient in programming.

Keywords:python language;static address;dynamic link;dynamic web reviews;reptile algorithm

1 引言(Introduction)

大数据具有数据体量巨大(Volume)、数据类型繁多(Variety)、价值密度低(Value)、处理速度快(Velocity)的特点。在大数据获取中面临的一个数据源问题为：如何获取大量的动态评论数据？Python是一门独立的语言，可以直接操作数据库，便于对大规模数据的操作与分析[1]。而且，由于Python包含结巴分词等程序包，可以直接进行分词,适宜于自然语言处理[2]。

现在很多网页通过Ajax动态请求、异步刷新生成数据[3]。Python由于先天局限，它爬取静态网页的方法难于直接提取动态网页。而爬取动态网页的方法虽然有很多，但便于新闻学、语言学、管理等学科的研究者应用的方法却很少。所以这篇论文研究如何用Python语言爬取Ajax动态生成的评论数据。

这篇论文延续前人的方法，通过静态网址信息构造动态链接，并增加了翻页的部分，把各种商品、新闻、社交网站、TV等动态网页评论的爬取方法归结为一套抽象的爬虫算法流程图。在此基础上实现了商品评论收集程序[4]。本文为实时评价数据采集技术的研究提供了新路径[5]。

2 基于Python的爬虫算法 (Reptiles algorithms based on Python)

网络爬虫即数据采集程序。主要有搜索引擎网络爬虫[6]、基于Agent的网络爬虫、迁移的网络爬虫、通用网络爬虫和聚焦爬虫等。其中聚焦爬虫是一种主题网络爬虫，它围绕主题内容采集数据。

静态网页是指不应用程序而直接或间接制作成Html的网页，每一个页面都有一个固定的URL地址，这个URL和相应的Html可以通过Python直接获取。动态网页一般使用脚本语言(Php、Asp等)将网站内容存于数据库中，相应URL动态链接不可以通过Python获取。但是动态URL的变化部分一般可以在相关静态URL及源代码中寻找，所以这篇论文在前人的基础上，利用静态的URL地址和相应的网页源代码构造动态链接，从而实现了Python直接对动态网页的爬取。本研究在前人基础上，通过对各主流商品、新闻、社交网站、TV等动态网页评论分析，提出了基于Python的动态网页爬虫算法流程图，如图1所示。

图1 动态网页评论爬虫算法流程图

Fig.1 Algorithm flowchart of dynamic pages

3 相关操作与爬虫程序(Operation and Reptiles procedure)

为顾及新闻、管理等的学科的研究者，相关操作比较详

基金项目：国家自然科学基金资助项目(71571139)“大数据情景的outlier分析与异类知识管理模式研究”.

细。工具：使用Chrome浏览器的开发人员工具或火狐浏览器的Firebug插件，这篇论文以Firebug插件为例[7]。首先安装火狐浏览器，版本为40.0，并装上Firebug，版本为2.0.13。程序以商品评论为例，具体分五步。

3.1 静态URL构造动态URL

(1)提取某个商品的静态URL网址

该文以商品为例，用浏览器打开某个商品的页面，复制地址栏网址如①所示。

http://item.jd.com/492036.html①

(2)提取对应评论页的静态URL

a.单击“商品评价”；b.复制地址栏的网址如②所示。

http://item.jd.com/492036.html#comment②

(3)提取含有评论数据的Ajax动态链接

这里总结前人获取评论动态链接的方法如下：a.用火狐浏览器在评论页空白部分，右键——使用Firebug查看元素，打开“Firebug工作面板”；b.点击工具面板上的“网络”；c.其子菜单默认在“全部”处；d.单击工作面板左上角的“清除”，以清除已有请求；e.在浏览器窗口中，点击评论第二页的图标；f.在“Firebug工作面板”上，右击"GET p-492036-……"这个动态请求，然后点击“复制地址”；其中，选定"GET p-492036-……"这个请求的原因为：这个请求的“响应”含有评论数据；g.粘贴这个“第二页”评论的动态地址如③所示。

http://club.jd.com/productpage/p-492036-s-0-t-3-p-1.html?callback=fetchJSON_comment98vv216③

通过动态网址③就可以得到第2页的评论。获取动态链接的操作如图2所示。

图2 获取含有评论的Ajax请求链接操作图

Fig.2 Get Ajax request link

动态网址③含有一些时间戳等无用部分，可以进行适当的简化。简化方法为：将网址③粘贴到浏览器地址栏，在保证网页结果保留JSON格式信息的前提下，按照分隔符逐个删除，直到最简，如④所示。

http://club.jd.com/productpage/p-492036-s-0-t-3-p-1.html④

(4)提取存储评论数据的另一页的Ajax动态链接

操作和3.1的第3节相同，但是，在其中的第5部分，应该点击评论的第三页。最后得到第三页评论的动态网址,并简化如⑤所示。

http://club.jd.com/productpage/p-492036-s-0-t-3-p-2.html⑤

(5)提取另外一个商品的已简化的Ajax动态链接如⑥所示

http://club.jd.com/productpage/p-1298665-s-0-t-3-p-1.html⑥

(6)根据静态网页信息构造Ajax动态链接

分析④⑤⑥动态网址的变化部分，找出组成结构如⑦所示

http://club.jd.com/productpage/p-+商品ID+-s-0-t-3-p-+页码标识+.html⑦

“商品ID”唯一标识了这个商品，“页码标识”表示不同的页码。在Ajax动态链接的组成结构中，对于变化部分，一般可以在静态网址①②以及由①②所得到网页的源代码中寻找，其中获取源代码方法：网页空白处右键单击——查看网页源代码。而对于这个网站，“商品ID”可以由静态网址①得到，“页码标识”一般为1…N的自然数。这样就可以由商品的静态网址及网页数据构造出评论的动态链接，从而爬取评论信息。

一般在一种网站中，不同商品对应的评论页动态网址⑦的格式是相同的。所以可以选择某一个商品的动态评论网址，设置为标准动态网址(comment_Norm)，为方便起见，这篇论文把网址④设置为comment_Norm。这样，对于任意一种商品，把comment_Norm的商品ID置换为本商品的ID，就可以得到这个商品的动态网址；置换页数，可以得到2…N页的动态网址(commentsUrl)，因为第1页的动态网址不易获取，所以从第2页开始爬取。

3.2 获取该Ajax请求返回的Json数据

用requests的get/post方法(或urllib、urllib2、beautifuSoup等)发送请求并接收数据：content=requests.get(comments_Url).content。用正则表达式提取标准数据：content='{'+re.findall(r"{(.+)}",content)[0]+'}'。然后转换为Json库函数可以处理的字典格式：dict=json.loads(content,"gbk")，其中“gbk”为这个网页的数据编码方式，Python默认编码方式为“Ascii”，当网站编码方式也为“Ascii”时，直接用json.loads(content)。

3.3 解析Json数据并保存结果

(1)解析Json数据

使用Python IDE即PyCharm解析Json数据，PyCharm版本4.0.5，python版本2.7.10。操作为：a.在dict=json.loads(content,"gbk")这句设置断点；b.点击“Debug/绿色甲虫”图标；c.点击“Step Over”；d.在“Variables”中右击“dict”变量；e.左键单击“add to watches”；f.在“Watches”窗口中点击“dict”变量前的三角符号，就得到了dict的树状结构。操作如图3所示。

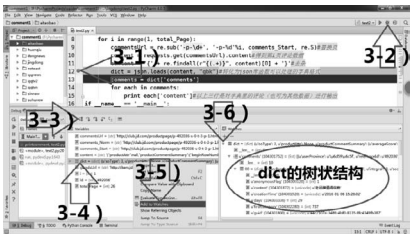

图3 解析Json数据操作图

Fig.3 Json parsing data

也可以使用一般浏览器的“FeHelper”插件解析Json数据，“FeHelper”插件版本v7.5。操作：把3.1的第3节的第6部分的“Response/响应”复制粘贴到FeHelper的“Json串格式化”的窗口中，单击格式化。也可以直接用Firebug插件，操作：在在3.1的第3节的第6部分，点击含有评论数据的响应——单击“JSON”按钮。在“JSON”内，即为Json数据的树状结构，如图2所示。

(2)寻找评论路径

dict['comments'][j]['content']即为评论,j为0—9自然数。

(3)保存结果

用easy_install或者是pip安装相应python包，以及安装对应的数据库软件。结果可以保存到mysql[8]、csv、excel、mongodb等数据库中。

3.4 停止条件

一般通过评论总页数判断，可以直接看有多少页(京东商城、国美在线等)；或则用评论总数除以每页个数得到总页码(天猫网、淘宝网、当当网、亚马逊卓越网、苏宁易购等)。或通过判断动态链接请求的返回值是否为空作为停止条件。

3.5 程序及结果

最后构造程序如图4所示。

```
#-*-coding:utf8-*-
import requests#一个HTTP客户端库，即urllib、urllib2类似
import json#导入json库
import re#导入正则表达式库
def printcomment(goodsUrl, total_Page, comments_Norm):
    id = re.findall('item.jd.com/(.*?).html', goodsUrl, re.S)[0]#本商品的ID
    comments_Start = re.sub('492036', id, comments_Norm, re.S)#替换为本商品ID
    for i in range(1, total_Page):
        commentsUrl = re.sub('-p-\d+', '-p-%d'%i, comments_Start, re.S)#替换页
        content = requests.get(commentsUrl).content#得到第i页评论数据
        content = '{' + re.findall(r"{(.+)}", content)[0] + '}'#标准Json数据
        dict = json.loads(content, "gbk")#转化为json库函数可以处理的字典格式
        for j in range(10):
            print dict['comments'][j]['content']#输出字典dict里的评论
if __name__ == '__main__':
    goodsUrl = 'http://item.jd.com/492036.html'#这个商品页的网址,不同商品有不同Url
    total_Page = 26#这个商品评论页数,不同商品一般不同
    comments_Norm = 'http://club.jd.com/productpage/p-492036-s-0-t-3-p-1.html
    printcomment(goodsUrl, total_Page, comments_Norm)#由静态地址得到动态评论的函数
```

图4 动态网页评论爬虫程序

Fig.4 Crawler of dynamic pages

实验收集的数据样本详见表1。

表1 收集的数据样本

Tab.1 Data samples collected

用户名	等级	评价	评级	回复	赞
jd_556857624	铜牌会员	质量比较轻，使着还可以吧	5	1	7

3.6 特殊情况

(1)自动获取停止爬取的标志

一般需要从含有评论数据的动态网页或其他动态网页中寻找相关数据。a.通过评论总页数：例如淘宝网，dict['maxPage']即为总页数。b.通过评论总数：例如京东商城的dict['productCommentSummary']['commentCount']为评论总个数，再除以每页的个数，即得到总页数。c.通过停止标志：例如腾讯TV、腾讯新闻。它们的停止标志为dict['data']['hasnext']，该值如果为false，则应停止爬取。

(2)页码标志符不是自然数

标志符一般需要从相关动态网页中寻找。例如腾讯新闻、TV。首先选取某个新闻，提取第一页已简化的动态评论网址如⑧所示，提取第二页的如⑨所示，动态网址的结构见⑩。

http://coral.qq.com/article/1267477591/comment?commentid=0&reqnum=10⑧

http://coral.qq.com/article/1267477591/comment?commentid=6081308797779398298&reqnum=20⑨

http://coral.qq.com/article/+新闻ID+/comment?commentid=+pageID+&reqnum=+rNUM⑩

“新闻ID”从评论页静态网址中提取；第1页评论动态网页的“rNUM”为10，第2…N页的“rNUM”为20；第1页的“pageID”为固定值“0”，其他页的“pageID”从前一页的动态网页中找，其中pageID=dict['last']。以此类推，这样就可以得到第1…N页的动态网址了。

其中第一页动态网址的获取方法为：进入评论页，打开“Firebug工作面板”，单击“清除”，然后刷新页面，在请求中逐个寻找。存储评论的请求一般包含在“网络”子菜单的JavaScript或XHR中，可以直接在这里找。

(3)遵守robot协议

在爬取数据的过程中，应严格遵守网络协议规定，经测试，6秒对服务器发起一次请求较为合适。用time.sleep(6)来控制速度。

(4)应对防爬虫方法

a.表头信息：对于一些网站需要表头信息，程序为：content=requests.get(comments_Url,headers=header).content。其中的comments_Url为存储评论信息的动态网址。

Header为表头信息，获取方法为：在3.1的第3节的第6部分点击任意一个请求—头信息—请求头信息—“User-Agent”。b.cookie:对于一些需要登录信息的网站，例如新浪/腾讯微博、twitter、QQ空间、Facebook、朋友网、人人网、网页版微信/来往等，需要Cookie信息。程序为：content=requests.get(comments_Url,cookies=cook).content。Cookie的获取方法为：先用浏览器登录账号，在3.1的第3节的第6部分点击含有评论信息的请求—头信息—请求头信息—Cookie。c.Form Data(表单数据):例如凤凰新闻、TV评论，由评论页动态网址并不能得到评论数据，还得加入Form Data，而且通过更改表单数据中'p'的值来翻页。程序如下：

```
comments_Url='http://comment.ifeng.com/get?job=1&order=DESC&orderBy=create_time&format=json&pagesize=20'
data={'p':'1','docurl':'http://news.ifeng.com/a/20151121/46335318_0.shtml'}
content=requests.post(comments_Url,data=data).content
```

data为表单数据的信息。获取方法为：在3.1的第3节的第6部分点击含有评论信息的请求—Post—参数。

(5)其他

环球新闻用content=re.findall(r"comment_list\((.+)\);",content)[0]语句提取标准Json数据。新浪、腾讯等网站，评论不分页显示，“加载更多/加载更多评论”按钮相当于第2…N页。优酷TV，动态链接返回Html。用正则表达式提取评论信息。对评论部分的字符串(例如：comment="\u559c\u6b22\u59ae")，用comment=comment.decode("unicode-escape")进行反编码后得到对应汉字。

4 对比分析(Comparative analysis)

该研究把本文所设计的爬虫与目前应用广泛的通用爬虫比较：通用爬虫以集搜客和网络神采为例。网络神采通用性最强(采集浏览器看到的)，采集内容范围广(支持登录、跨层、POST、脚本、动态网页)，但需要设置许多参数；基于Python的动态网页评论爬虫专门针对评论，而且爬取过程不依赖于浏览器，因此其效率比集搜客和网络神采快些。在复杂度方面，网络神采考虑的因素比较全面，所以比评论爬虫算法复杂得多；而集搜客，基本不用编写程序，甚至直接使用现成的采集规则。网络神采扩展性强(支持存储过程、插件、二次开发)，集搜客可以导入excel，而Python可操作各种DB。三种爬虫对比分析详见表2。

表2 三种爬虫对比分析

Tab.2 Comparison of three kinds of reptile

指标	网络神采	集搜客	动态网页评论爬虫
通用性	很强	很强	较强(简单更改)
灵活性	很高	很高	一般(随网页变化)
扩展性	强	弱	很强
针对性	一般	一般	高
复杂度	复杂	复杂	一般
速度	一般(约200条/分)	一般	快(约290条/分)

5 结论(Conclusion)

研究在前人的基础上，设计了基于Python的商品、新闻、社交网站、TV评论聚焦爬虫算法。以此为基础，实现了商品评论的收集程序。基于Python的评论爬虫具有一定的高效性、通用性、实时性，所以可以作为实时商品、新闻、社交网站、TV评论采集算法；这种算法基于自然语言处理能力强的Python语言,利于对评论文本的后续分析以及相应爬虫软件[9]的开发。而且这种爬虫比较简单，可以被计算机基础弱的评论挖掘研究者使用。

参考文献(References)

[1] 彭磊,李先国.大数据量Excel数据导入系统的设计与实现[J].计算机应用技术,2014(14):57-59.

[2] 吴宏洲.分词技术的研究与应用——一种抽取新词的简便方法[J].软件工程师,2015,18(12):64-68.

[3] 王佳.支持Ajax技术的主题网络爬虫系统研究与实现[D].北京:北京交通大学,2011:22-27.

[4] 方美玉,郑小林,陈德人.商品评论聚焦爬虫算法设计与实现[J].吉林大学学报(工学版),2012(S1):377-381.

[5] 陈国良.基于商品评论信息的特征挖掘[J].福建电脑,2015(05):106-107.

[6] 刘典型.基于概念聚类的Web数据挖掘搜索引擎的设计与实现[J].软件工程师,2015,18(5):18-20.

[7] Winterto1990.python爬取ajax动态生成的数据以抓取淘宝评论为例子[EB/OL].[2015-08-26]http://www.th7.cn/web/ajax/201508/117293.shtml.

[8] 陈潇.SQL Server2008数据库存储过程的应用[J].软件工程师.2015,18(6):18-19.

[9] 刘正春.基于Carbide.C++的Symbian OS软件开发[J].电脑与电信,2009(01):47-49.

作者简介：

夏火松(1964-)，男，博士，教授.研究领域：决策支持系统.

李保国(1990-)，男，研究生.研究领域：信息管理.