# 基于Python的新浪微博用户数据获取技术

东南大学信息科学与工程学院　罗　咪

【摘要】为了获取用于社交网络研究的新浪微博用户数据，本文改进了传统网络爬虫，设计了一个基于Python的新浪微博爬虫系统。该系统使用scrapy多线程爬虫框架，实现了模拟登陆、动态网页抓取和克服微博反爬虫机制等功能，抓取后数据被存储在MySQL数据库中，便于后续分析。实验结果表明，该爬虫系统获取数据的实行性和效率高，稳定性和准确性较好。

【关键词】微博数据；Python；Scrapy框架；反爬虫机制

DOI:10.19353/j.cnki.dzsj.2018.05.071

## 引言

新浪微博作为我国主流社交媒体，拥有海量数据。自2009年推出以来，新浪微博的使用人数急速上升，带来的是信息量的剧增。每天，人们通过转发、互粉、点赞等行为发表自己的喜恶看法，将个人观点放大到社会空间。在如今这个大数据时代，社交网络分析依赖于海量的数据来探索人类社会关系中的奥秘，而如何获取这些数据至关重要。在此之前，国外的研究学者已经对Twitter、YouTube等社交平台进行了一系列的分析，其研究方法相对成熟。他们获取数据的主要方法是通过网站官方提供的API接口。[1]在国内，由于新浪官方目前限制相关数据接口的使用，所以研究者要想获取数据，需另辟蹊径。[2]

本文基于Python语言提出了一种无需借助官方API接口就能获取用户数据的方法---多线程爬虫技术。该项技术与传统的网络爬虫相比，主要有以下三点改进：首先，使用多线程爬虫取代传统的单线程爬虫，提高了数据获取速率；其次，针对微博的反爬虫机制设计了四种突破策略；最后，成功实现了对于微博评论等动态网页的爬取。

## 1.Scrapy多线程爬虫框架

目前，用python实现多线程爬虫主要有两种方法。一是自行设计多线程函数，二是使用python的scrapy包来包装线程对象。[3] 本爬虫系统主要使用Scrapy框架编写，Scrapy框架是一种引擎和下载器之间的框架,主要功能是处理Scrapy引擎与下载器之间的请求及响应。其基本组件和工作原理如图1所示。

在Scrapy架构下，用户需要编写爬虫部件spiders和数据处理部件item pipeline。

## 2.模拟登陆

模拟登陆是爬虫技术所要攻克的第一个难题。所谓模拟登陆，即让计算机模仿人工操作，以达到欺骗服务器的目的。我们分别尝试了以下四种策略，并比较了它们的优劣性。

（1）手动获取 cookie 登陆：该方法较为简单，但是需要人为参与，自动化程度低。

（2）post 方法登陆weibo.com：该方法难度较高，主要因为以下三点。其一，网页版微博上存在大量的广告图片，javascript代码复杂，从而降低了后续工作中源码分析的效率。[4]其二，官方分别使用了Base64 和 RSA 加密算法对用户账号和密码进行了加密。其三，第一次跳转到的URL在使用正则表达式匹配后才能得到目标主页。

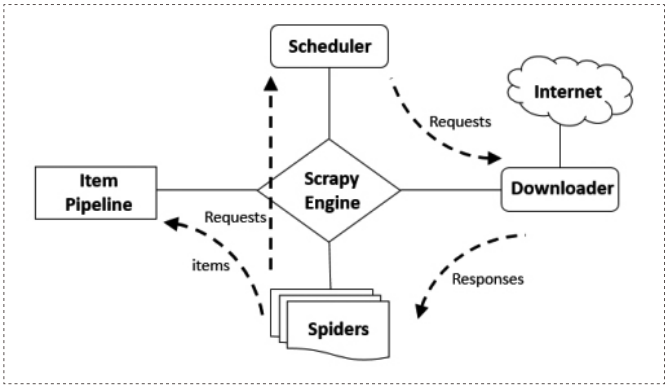


图1 Scrapy多线程爬虫框架原理图

（3）post方法登陆weibo.cn:由于移动端页面相对简洁，网页源码较少，所以相对登陆网页版更加简单，但也存在两个难点。其一，登陆时会出现验证码，本文采用的解决方法是下载到本地然后手动输入。其二，登陆 http://weibo.cn/ 后会有一个重定向，这时必须设定user-agent，否则post完成后会卡在跳转页面。

（4）利用自动化测试工具Selenium: Selenium 是一个自动化测试工具，相当于一个没有界面浏览器，可以完全模拟浏览器行为，所以利用它进行模拟登陆非常方便。[5]

## 3.动态网页抓取

新浪微博与豆瓣、知乎等其他社交网站的区别在于：微博中很多网页都是动态的，爬取有一定的难度，所以需要特殊处理。主要有两种爬动态网站的策略可以选取。第一种方法是使用自动化测试工具selenium进行抓取，这主要是因为chrome driver可以渲染用javascript生成的网站。具体做法是先获取网页源码，再使用 XPath 路径语言解析网页。Selenium 抓取虽然简洁方便，但存在抓取速率

基金项目：江苏省高等学校大学生创新创业训练计划项目（201710286018Y）。

低效，不稳定的缺点，所以还可以使用解析json数据的方法。下面分别讲解微博评论和用户粉丝列表数据的获取办法。

### 3.1 微博评论抓取

首先，抓包寻找到网站ajax请求的接口，接着找寻该的接口url地址规律。例如，我们抓包后发现ajax请求为：

https://weibo.com/aj/v6/comment/big?

ajwvr=6&id=4199411268012827&root_comment_max_id=442152675631872&root_comment_max_id_type=0&root_comment_ext_param=&page=2&filter=hot&sum_comment_number=1753&filter_tips_before=0&from=singleWeiBo&__rnd=1520153792622。

经过测试后，我们可以将该url接口简化为：

https://weibo.com/aj/v6/comment/big?ajwvr=6&id=4199411268012827&page=2

从简化后的url地址可以归纳出新浪微博评论的接口规律为：

https://weibo.com/aj/v6/comment/big?ajwvr=6&id=“这条微博独一无二的id”&page=“指定的页面”。

下面，我们需要通过微博地址获取这条微博独一无二的id。经过网页源码分析我们发现，微博的id出现在网页dom元素属性的很多地方，而且都是以mid开头，这样就能根据网页源码正则匹配出这条微博的id。[6]最后只要改变url接口里的page参数，获取所有评论源码，再使用xpath提取出所需的评论数据即可。

### 3.2 微博粉丝列表抓取

以查看刘亦菲粉丝列表为例，抓包后得到结果如图2所示。

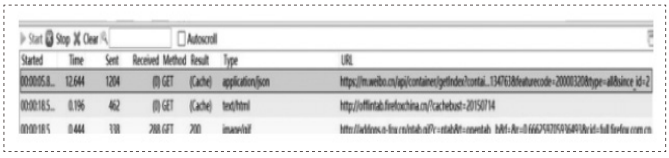

图2 刘亦菲粉丝列表抓包结果

可以得出粉丝列表的url地址为：

https://m.weibo.cn/api/container/getIndex?containerid=231051_-_fans_-_3261134763&type=all&since_id。

其中，每个粉丝拥有不同的containerid参数，而since_id从1到250都可以访问，所以每次访问这个url都可以返回含有20个粉丝信息的json数据。如图3所示。


图3 刘亦菲粉丝列表json数据返回结果

### 4.克服反爬虫机制

为了防止个人盗取微博数据用于非法用途，新浪微博官方对微博爬虫采取抵制态度，并且他们的反爬虫机制也在不断完善。如果爬虫请求过于频繁，账户容易被封禁，降低了数据获取的效率。本系统采用以下4个策略突破微博的反爬虫机制：

（1） 动态更改 user-agent：主要利用python的fake user agent包，让每一次请求都伪造一个用户代理，使服务器误以为是来自不同浏览器的请求。从而降低爬虫被发现的概率。

（2）动态更改 IP：首先获取一串可用的高速代理 IP 列表，然后在每一次请求时更改 IP 地址，让服务器无法锁定具体的访问地址，以达到迷惑服务器的目的。

（3）控制爬取速率：若爬取速率过快，服务器容易检测出异常，所以速率一般控制在 1.5-2s为宜。

（4）建立并维护cookie 池：由于新浪微博会针对一个账户进行速率监控，所以更稳妥的方法是一次多获取几个 cookie，每次请求随机设定一个 cookie。

### 5.总结

本文设计的新浪微博爬虫使用python语言实现，并且可以根据实际需要更改爬取条件和爬取目标。在传统静态网页爬虫的基础上，探讨了针对新浪微博网站需要解决的三大问题: 模拟登陆、动态网页信息抓取、反爬虫机制，并给出了切实可行的解决方案。在动态网页抓取中，需要应用自动化测试工具Selenium来模仿真正用户的操作，以及利用正则表达式来匹配数据。值得一提的是，最终得到的数据往往与所需要的信息有细微差别，需要经过数据清洗才能进入分析的流程。实践证明，该爬虫系统能够实现对微博数据高效稳定的采集，并符合实时性性、健壮性和灵活性等性能指标要求。

### 参考文献

[1]Axel Bruns, Yuxian Eugene Liang. (2012). Tools and methods for capturing Twitter data during natural disasters. Peer-Reviewed Journal on the Internet, 17(4).Retrieved from:http://journals.uic.edu/ojs/index.php/fm/article/view/3937.

[2]吴剑兰.基于Python的新浪微博爬虫研究[J].无线互联科技,2015(06)：93-94.

[3]李俊丽.基于Linux的python多线程爬虫程序设计[J].计算机与数字工程,2015,43(05):861-863+876.

[4]陈珂,蓝鼎栋,柯文德,黎树俊,邓文天.基于Java的新浪微博爬虫研究与实现[J].计算机技术与发展,2017,27(09):191-196.

[5]吴伶琳.基于Selenium的软件自动化测试的研究与应用[J].计算机与现代化,2013(02):65-68.

[6]胡军伟,秦奕青,张伟.正则表达式在Web信息抽取中的应用[J].北京信息科技大学学报(自然科学版),2011,26(06):86-89.