基于 Python 的新浪微博数据爬虫程序设计

◆陈 琳 任 芳

摘要：为了快速地获取到海量微博中的数据，根据微博网页的特点，提出了一种基于Python爬虫程序设计方法。通过模拟登录新浪微博，实时抓取微博中指定用户的微博正文等内容；该工具利用关键词匹配技术，匹配符合规定条件的微博，并抓取相关内容；最后使用该工具对部分微博数据作了一个关于雾霾问题的分析。实验结果表明：本程序具有针对性强、数据采集速度快、易嵌入开发、简单等优点，为不善于编程的研究者提供了快速获取微博的方法，有利于对微博的后续数据挖掘研究。

关键词：Python；爬虫；新浪微博；雾霾

引言

过去几十年里，Web 的迅速发展，大量的数据通过 Web 发布，使其成为世界上规模最大的公共数据源。随着网络的高速发展，互联网成为海量信息的载体，如何有效地提取并利用这些信息成为研发人员一个巨大的挑战。为了更好地吸引开发者，以及和开发者更好的交互，微博等社交网络平台提供了一些数据访问编程接口（OpenAPI）供研发人员获取数据，但是，由于各方面的考虑，利用 OpenAPI 进行数据抓取时总是有各种各样的限制[1][2]。以新浪微博为例，新浪提供的微博 API 对普通用户的权限和抓取频率都进行了较为严格地限制，而且无法对微博内容进行搜索[3]。为此，本文提出了一款基于 Python 的新浪微博数据爬虫程序，为微博数据获取提供技术支持。

本文提出的程序通过爬虫模拟登录移动端新浪微博并获取相关微博数据，并将这些数据保存到本地，方便进一步的数据挖掘与分析。同时，本文爬虫还集成了关键词匹配功能，利用该匹配功能可以实现指定关键词的数据获取。使用本文爬虫程序能够节省数据分析人员的开发程序的时间，使得他们可以将更多的精力放在数据分析上面，同时也可以对海量数据起到过滤作用。

一、相关概念

1.1Python 语言。Python 语言是一种功能强大、语法简洁清晰的开源编程语言，几乎能够在目前所有的操作系统上运行；Python 是高效率的完全面向对象的语言，能有效而简单地实现面向对象编程[4][5]。Python 解释性语言的本质，再加上其简洁的语法和对动态输入的支持，使得它在大多数操作系统平台上都是一个较为理想的脚本语言，特别适用于快速的应用程序开发[6]。Python 提供了针对网络协议标准库，对网络协议的各个层次进行了抽象封装，程序员就可以集中精力处理程序逻辑。其次，Python 非常擅长处理字节流的各种模式，具有很快的开发速度[7][8]。

1.2 网络爬虫。网络爬虫[9]（Web Crawler），是一种按照一定的规则，自动提取 Web 网页的应用程序或者脚本，它是搜索引擎抓取数据系统的重要组成部分，并为搜索引擎从互联网上下载 Web 页面。爬虫的目的是将互联网上的网页下载到本地形成互联网内容的备份。爬虫是从一个或多个初始页面的 URL，通过分析页面源文件的 URL，抓取新的 Web 链接，通过这些 Web 链接，再继续寻找新的 Web 链接，如此不断循环，直到抓取和分析所有页面。当然这是理想情况下的执行情况，根据现在公布的数据，最好的搜索引擎也只爬取整个互联网不到一半的网页。

1.3 模拟登录。不同于以前传统 Web 网站不需要登录，现在的大部分社交网站需要登录才能进入个人主页，不登录访问将会自动跳转到登录页面。所以需要设计一种适用于社交网站爬虫程序。该程序既可以支持登录，而且可以获取大量用户的信息。上述过程就需要采用 Session 机制来解决。Session 机制通过 Cookie 和 URL 重写来实现用户登录。通过 Cookie 可以实现 Session 会话，但如果客户禁用了 Cookie，那么就只有使用 URL 重写[10]。

二、程序设计

爬取新浪微博，大致有两种方法，一种是用纯爬虫，还有一种是用新浪提供的 OpenAPI 接口。虽然新浪微博的 API 接口便于开发者开发，但其限制也很大：比如开发者必须经过新浪网站的授权，且授权有有效期；OpenAPI 接口访问频次也有限制等，所以本文采用纯爬虫进行程序设计。因为 PC 端的新浪微博是 Ajax 的动态加载，爬取数据技术上有难度，而移动端的微博可以通过分页爬取的方式来一次性爬取所有微博内容，所以本程序改为对移动端的微博进行爬取。

程序设计主要包括 3 个模块，首先通过已注册的账户和密码登录移动端新浪微博，分析移动端新浪微博的网站源码，找到获得该账户的 Cookie；获得想要爬取用户的 user_ID，从 Request URL 获得登陆页面 html 代码，利用 Python 的 lxml 库分析 html 代码；根据关键字筛选符合关键词条件的微博，并保存在本地；对爬取的数据进行数据分析的实验：与雾霾相关的微博与时间关系的分析。程序设计流程如图 1 所示。

图 1 程序设计流程

2.1 获得账户 Cookie 并获得登陆页面代码

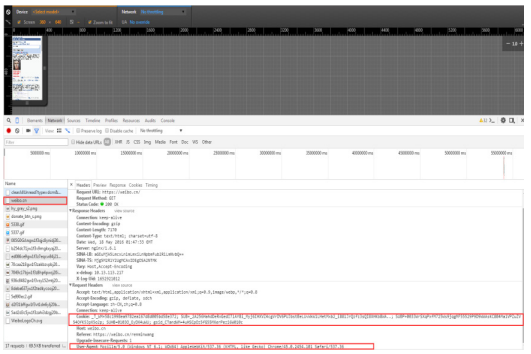

图 2 网站“开发者工具”界面

Cookie 英文含义指饼干，Cookie 是一小段的文本信息。用户客户端请求服务器，服务器记录该客户端状态，把账号按照一定的规则加密后，连同账号一块保存到 Cookie 中。使用 response 向客户端浏览器颁发一个 Cookie。客户端浏览器把 Cookie 保存起来。当浏览器再对该网站请求时，浏览器会把请求的网址和 Cookie 一同提交给服务器。服务器检查该 Cookie，以此来辨认客户端状态。

首先用 chrome 浏览器打开新浪微博移动端 option+command+i 调出开发者工具，点开 Network，将 Preserve log 选项选中，输入账号密码，登录移动版新浪微博 。找到 m.weibo.cn->Headers->Cookie 和 headers，如图 2 所示。

统一资源定位符（URL）是对可以从互联网上得到的资源的访问方法和位置的一种简洁的表示，是互联网上网页的地址。互联网上的每个文件都有一个唯一的 URL，URL 使用数字和字母按一定顺序排列以确定一个地址，而且它包含的信息能让浏览器知道怎么处理它。本文爬取的 URL= 'https://weibo.cn/user_id'，爬取的用户 user_id = "renminwang"。

2.2 分析网页。在 Python 语言众多 HTTP 客户端库中，除了自带库 Urllib、Urllib2 库，还有一个功能更为强大操作更为简单的第三方库 requests。requests 库提供一系列用于操作 URL 的函数，使开发者可以像读取本地文件一样读取互联网和 ftp 上的数据。requests 库可以非常方便地抓取 URL 内容，也就是发送一个 GET 请求到指定的页面，因为 requests 自带 Json 格式解析，最后返回不需要进行对 Json 格式数据进行转换。html =requests.get(url, cookies = cookie, headers=headers).content 返回的是页面二进制的数据 Requests 库获取了新浪微博的页面源代码，通过 Python 的 lxml 库的 etree.HTML 来处理网页源代码，从而生成一个可以被 lxml 库的 xpath 方法解析的对象。而 xpath 方法是用一种与目录树类似的方法，用来描述源代码中的结构在 HTML 文档中的路径。用“/”作为上下层级路径的分隔。定位某一个 HTML 标签，可以使用类似文件路径里的相对路径，例如微博文本 wbContent 保存在 HTML 标签 的元素组 class 中，class 属性为“ctt”；微博发布时间 wbDate 保存在 HTML 标签 的元素组 class 中，class 属性为“ct”；微博总页数 pageNum 保存在 HTML 标签 <input> 的元素组 name 中，class 属性为“mp”的多维字典中，键为 value，pageNum 为其值。函数 getWB() 获取微博文本和发布时间，分别返回包含文本和发布时间的列表 list。

上述流程关键代码如下：

```
pageNum = (int)(selector.xpath('//input[@name="mp"]')[0].
attrib['value'])
for page in range(1,pageNum+1):
  url = 'https://weibo.cn/renminwang?&page=%d'%(page)
  lxml = requests.get(url, cookies = cookie).content
  selector = etree.HTML(lxml)
  wbContent = selector.xpath('//span[@class="ctt"]')
 wbDate= selector.xpath('//span[@class="ct"]')
  listtxt=getWB(wbContent)
  listdate=getWB(wbDate)
def  getWB(Content)
  for each in Content:
    text = each.xpath('string(.)')
    list.append(text)
return list
```

2.3 关键词匹配。据权威机构统计，截止 2015 年底，新浪微博月活跃用户数已经达到 2.12 亿人，新浪微博作为基于用户关系的信息分享、获取平台，其微博文已成海量信息源。针对科研研究而言，一个用户的全部微博中肯定含有很多无意义的内容。例如一个用户的微博中关注雾霾与时间关系的研究就需要该用户在所有的微博中找到与雾霾相关的微博，并保存这些内容。而简单的爬虫并不能区分哪些微博有用，哪些微博没有用，如果把全部微博都保存到本地磁盘上就会耗费大量的存储空间，同时还会花费更多的时间从这些数据中再次筛选与雾霾相关的微博，这样做既浪费了资源又浪费了时间。因此在爬虫需含有关键字匹配模块，这样既方便了根据关键字筛选符合条件的微博，又提高了爬虫效率，节省了资源。用户为“renminwang”，关键词为雾霾，下载到本地的文件，如图 3 所示：

以下就是微博匹配关键字的关键代码：

```
def  findKey(list,str)
  for i in list
    index = i.find(str)
    if  index == -1
      continue
    listindex.append(index)
                    return listindex
```

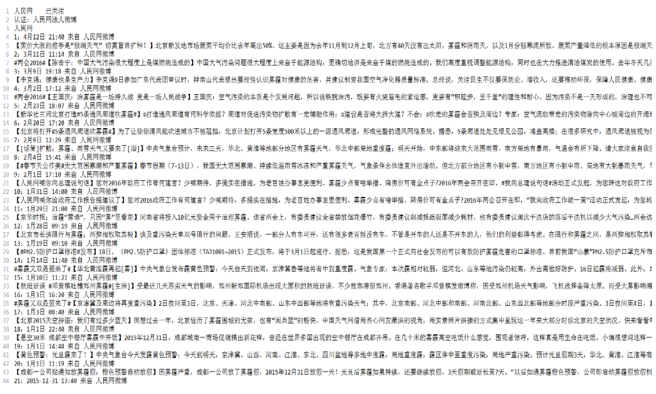

图 3 下载到本地的微博

三、数据分析

对已抓取的数据进行数据分析：一个是与雾霾相关的微博与时间关系的分析。在该实验中，针对人民网从 2010 年 1 月到 2016 年 5 月这段时间内，新浪微博账户人民网上的共计 87502 条微博，进行了关键字雾霾的匹配分析，并且记录下了 344 条符合匹配条件的微博。图 4 显示的是从 2010 年的 2016 年之间不同年份与雾霾相关的微博数量随时间的变化曲线。

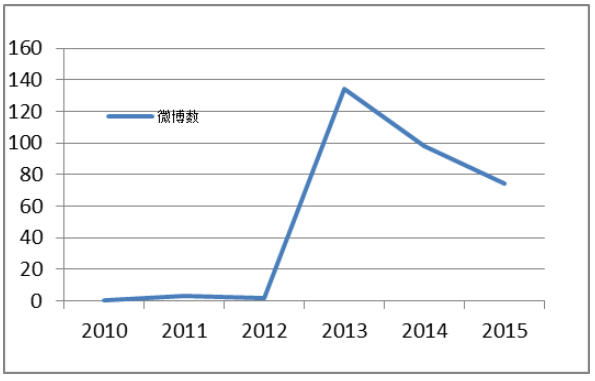

图 4 不同年份与雾霾相关的微博总数量随时间的变化曲线

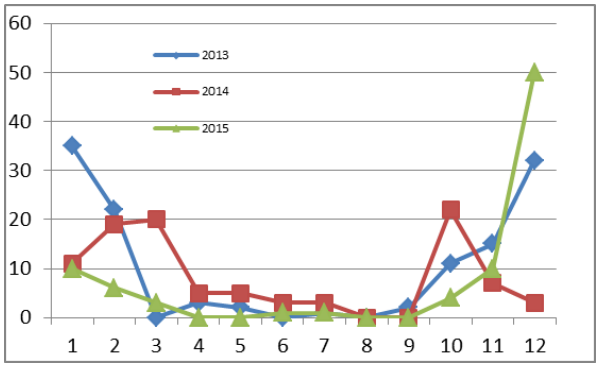

图 5 不同年月与雾霾相关的微博数量随时间的变化曲线

从图 4 可看出，在 2013 年度人们对于雾霾的关注度总数在 2010 年到 2015 年间最高，这说明 2013 年雾霾天气的频发重发成为一个民众关注的热点问题，2013 年也被大众称之为“雾霾年”。随着国家对环境的治理力度不断加大，空气质量问题得到改善，人们对雾霾关注度开始呈逐渐下降的趋势。从图 5 可看出，雾霾的高关注度主要集中在全年的 1–3 月和 10–12 月，这与气象专家分析雾霾易发生在每年秋冬季节的判断较为一致的。

四、总结

用 Python 丰富的标准库以及快速开发的特长，本文设计了基于 Python 的新浪微博数据爬虫程序，为社交网络研究者们提供了较为简单快捷的新浪微博数据获取工具。工具使用者只需提供新浪微博账号就能利用爬虫抓取新浪微博中的相关数据；使用者通过关键词，本文爬虫就能自动搜索匹配相关内容并将符合条件的微博保存到磁盘之上。实验结果表明：本程序基于自然语言处理能力强的 Python 语言，利于对微博的后续挖掘研究。

参考文献

[1] 郭涛，黄铭钧．社区网络爬虫的设计与实现 [J]．智能计算机与应用 ,2012,2(4):65–67.

[2] 廉捷，周欣，曹伟，等．新浪微博数据挖掘方案 [J]. 清华大学学报：自然科学版，2011,51(10):1300–1305.

[3] 刘艳平，俞海英，戎沁 .Python 模拟登录网站并抓取网页的方法 [J]. 微型计算机应用 ,2015,31(1):58–60.

[4]Mark Lutz.Learning Python[M]. 北京：机械工业出版总社 ,2009.

[5] 刘志凯，张太红，刘磊．基于 Web 的 Python3 编程环境 [J]. 计算机系统应用 ,2015,24(7):236–239.

[6] 王大伟．基于 Python 的 Web API 自动化测试方法研究 [J]. 电子科学技术 ,2015,2 (5):573–581.

[7]Magnus Lie Hetland, 司维，曾军葳，等．Python 基础教程 [M]. 北京：人民邮电出版社 ,2014:243–245.

[8] 高森．Python 网络编程基础 [M]. 北京：电子工业出版社 ,2007:326–327.

[9] 周立柱，林玲．聚焦爬虫技术研究综述 [J]. 计算机应用 ,2005,25(9):1965–1969.

[10] 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(5):861–863.

（作者单位：陈琳，四川省气象台；任芳，陕西省气象服务中心）