文章编号: 2095-2163(2018)04-0112-04 中图分类号: TP319 文献标志码: A

基于 Python 的反反爬虫技术分析与应用

余豪士, 匡芳君
(温州商学院 信息工程学院, 浙江 温州 325035)

摘 要: 爬虫软件是现今互联网环境下,高效准确地获取数据的重要方式之一。针对传统的初级爬虫技术易于被目标网站拦截访问的问题,简述爬虫的工作原理和方式,讨论爬虫、反爬虫与反反爬虫之间的相互关系。分析应对目标网站的反反爬虫机制,包括伪装用户代理,设置 IP 地址代理、使用自动化测试工具调用浏览器等技术要点,并分析了基于 Python 语言中 Requests 库,构建了对网页的多种请求方式和数据获取方法的解决方案。结合反反爬虫机制与数据分析技术,以哔哩哔哩视频网为案例,分析其网页基本结构与调用的应用程序接口,使用 Python 与 Requests 库抓取网站所有视频的相关数据。数据清洗后分析播放量最高视频的相关信息,并将结论以数据可视化的方式呈现,实现对数据的获取、挖掘与分析。

关键词: 网络爬虫; 反爬虫; 反反爬虫; 大数据; 数据分析

Crawler and anti-anti-crawler technology based on Python

YU Haoshi, KUANG Fangjun
(School of Information Engineering, Wenzhou Business College, Wenzhou Zhengjiang 325035, China)

Abstract: Crawler software is one of the most important ways to obtain data effectively and accurately in the current Internet environment. In view of the traditional crawler technology which is prone to be intercepted by target website, the paper explains how the crawler appears to work, discusses about the relationship between crawler, anti-crawler and anti-anti-crawler, and analyzes the mechanism of anti-anti-crawler for the target website, including fake user agents, setting IP proxy address, calling browser using automated testing tools. Furthermore, multiple requests and data acquisition methods for web pages are built based on Requests Library in Python language and its solution is analyzed. Combined with the mechanism of anti-anti crawler and data analysis technology, the paper takes the Bilibili website as a case, analyzing its basic structure, as well as its API called. On the one side, all relevant data of video on the Bilibili website is captured using Python and Requests Library and the related information of the video, in which the highest click rate is analyzed after data cleaning. On the other side, the conclusion is presented in the way of data visualization, and the data acquisition, mining and analysis are also realized.

Key words: Web crawler; anti-crawler; anti-anti-crawler technology; big data; data analysis

引言

大数据时代下的数据来源和获取尤为重要[1],爬虫技术作为一项获取数据的工具而被广泛应用。已超过 60%的互联网流量来自爬虫(Spider),各大搜索引擎门户网站以及新闻网站的文章都与爬虫息息相关。爬虫技术已成为当今的研究热点,目标网站对爬虫软件所做的各方面防范,给出了不同的拦截方式[2]。开发者与开发者之间通过爬虫、反爬虫、反反爬虫技术进行较量,一方面开发者想通过爬虫脚本获取数据,另一方面开发者又想拦截爬虫,防止爬虫脚本妨碍本网站的正常运营,对正常用户的访问造成了负面影响。

1 反反爬虫概述

1.1 反反爬虫技术

爬虫软件是一种模拟浏览器的行为,是从指定网站抓取和保存网络数据的应用软件。爬虫软件提取出存在于网页上的数据,并以结构化的方式存储。主要活动于计算机网络通信模型中的传输层与应用层。传输层使用 TCP/IP 协议与目标 Web 服务器进行数据传输; 应用层使用 HTTP 或 HTTPS 协议与目

基金项目: 国家自然科学基金(61402227)。

作者简介: 余豪士(1997-),男,本科生,CCF 会员(90298G),主要研究方向: 数据分析、前后端开发; 匡芳君(1976-),女,博士,教授,CCF 会员(14005G),主要研究方向: 群智能与多目标优化、模式识别、信息安全等。
通讯作者: 匡芳君 Email: kfjztb@126.com

收稿日期: 2018-05-11

标 Web 服务器通信[3]。

由于传统的初级爬虫不使用任何隐藏伪装手段，在对站点发送大量请求时，会加重目标 Web 服务器的负担，且容易被服务器侦测。在大中型网站中，开发者会针对传统的初级爬虫制定一系列的反爬机制，如针对爬虫软件所处终端进行 IP 限制；针对请求报文中 Header 属性拦截爬虫软件；通过分析网站流量和日志统计分析过滤爬虫。爬虫开发者针对反爬虫机制，开发了一套反反爬虫机制，在爬取数据的过程中防止被目标站点拦截，开发者需最大限度地将爬虫模拟成真人行为，获取真实可靠的数据。初级爬虫、反爬虫、反反爬虫的关系如图 1 所示。

图 1　爬虫、反爬虫、反反爬虫关系

Fig. 1　The relationship between crawler, anti-crawler and anti-anti-crawler

1.2　反反爬虫策略

1.2.1　降低访问频率

对目标站点连续访问不同网页，如果不限制爬虫的请求频率，爬虫的效率只会受到所处终端的处理能力和带宽的限制，因此爬虫的访问频率会非常高。通过增加线程的休眠时间，降低访问频率，实现模仿人为浏览的行为。具体代码如下：

```
import time
time.sleep(0.5)
```

1.2.2　伪装用户代理

用户代理(User-Agent)是一种代表用户行为的属性，用于发送 HTTP 请求描述用户系统和浏览器信息。站点服务器通过获取报文中的 User-Agent 属性，给不同操作系统与浏览器发送不同页面。通常爬虫软件在请求数据时不会携带此属性字段，目标站点也因此可侦测与进行拦截。所以，爬虫脚本在请求时需在头部加入类似浏览器的 User-Agent 属性[4]。例如：

```
headers = {'User-Agent': 'Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36'}
data = requests.get(url, headers=headers).text
```

1.2.3　IP 代理

爬虫脚本在访问请求的过程中，TCP 报文会携带客户端的 IP 地址，站点服务器也因此可获取到客户端的 IP 地址。爬虫软件访问频率过高，站点服务器可对此 IP 地址进行暂时性的封禁。开发者在编写脚本时需要设置 IP 代理池。在多进程下，多个进程间使用不同的 IP 代理访问目标网站，绕过站点服务器 IP 地址字段的检测，加快爬取数据的效率。例如：

```
proxies = {'http': 'XX.XX.XX.XX:XXXX',
'https': 'XX.XX.XX.XX:XXXX'}
data = requests.get(url, proxies=proxies).text
```

1.2.4　使用自动化测试工具 Selenium

Selenium 是一个用于 WEB 开发自动化测试的软件，其本身用于从用户角度使用终端测试 Web 应用，加载浏览器驱动对网页进行操作。爬虫开发者使用 Selenium，并设置适应的浏览器，例如 Chome Driver 或无头浏览器 PhantomJS，最大限度模拟真人行为。应用代码如下：

```
from selenium import webdriver
driver = webdriver.Chrome()
```

1.2.5　访问移动端站点

网站根据终端浏览器的用户代理相应不同的页面，其中终端分为移动端和 PC 端。移动端站点地址通常以 WAP 开头，且对爬虫软件的限制不如 PC 端强。如果目标站点有移动端页面且数据可抓性高，可以对移动端页面进行抓取[5-6]。

2　基于 Requests 库编写爬虫

Python 中的第三方 HTTP 库、Requests 库被爬虫开发者广泛应用。Requests 集成了定制请求头、发送请求、传递 URL 参数、获取相应内容等多种函数[7]。

2.1　发送请求

在发送请求上，Requests 集成了多种请求方式，例如最普遍的 get 和 post 请求，还有其他 HTTP 协议中的请求类型。具体实现过程如下：

```
response = requests.get('https://httpbin.org/get')
response = requests.delete("http://httpbin.org/delete")
response = requests.options("http://httpbin.org/get")
```

2.2　传递 URL 参数

在浏览器地址输入栏，输入目标网址的地址后，可输入以键值对形成的参数，最终形成一个完整的 URL 地址跳转至目标网页。同理在 Requests 库也有此功能，以字典的形式构建。实现过程如下：

```
params = {'key1':'value1','key2': 'value2'}
response = requests.get('http://yhslib.com', params= params)
```

若要查看构建后的完整地址，也可输出查看。

2.3 定制请求头

HTTP 请求头，Requests 库也给出了定制方式，以字典的形式构建。实现过程如下：

```
headers = {'content-type': 'application/json'}
response = requests.get('http://yhslib.com', headers=headers)
```

2.4 获取相应内容

通常所需的数据会显示在网页上，这也说明数据包含在 HTML 或者 JavaScript 等文本类型的文件中，通过获取其文本信息经过筛选即可获得数据。Requests 库中可以通过获取 text 获得其文本：

```
r = requests.get('http://httpbin.org/get')
print(r.text)
```

有些情况下，所需数据以二进制的文件存在，例如图片、音频、视频等。在 Requests 中可通过获得二进制数据，通过解码和编码得到最终数据文件。

JSON 数据在数据交换和 API 接口领域中广泛应用。Requests 中，对 JSON 类型数据有独立的获取方式：

```
r = requests.get('https://XXX.XXX')
print(r.json())
```

3 案例分析

哔哩哔哩视频网是中国的弹幕视频分享网站，此网站的特色是悬浮在视频上方实时地评论社交功能[8]。哔哩哔哩网主打动漫视频，吸引了大量年轻用户，具有音乐、舞蹈、科技、生活等板块。据统计，此网站注册用户已超过 1.5 亿，其中 24 岁以下用户占总用户数的 75%，每日视频播放量已超过 1 亿。分析网站中各个视频的播放次数等关键数据，得出用户对此网站视频的喜好。

3.1 分析网页

打开哔哩哔哩弹幕网中任意视频详情页，分析 HTML 代码[9]，可以发现每一个视频页中都有其相应的播放量，用户发送的弹幕数、捐赠投币数和收藏数等关键数据，如图 2 所示。检查其元素属性和网页元数据可以发现，各个数值并非存在于网页源码中，而是通过 AJAX[10] 方式进行异步交互[11] 最终显示在页面中，因此需要从加载资源寻找。

图 2 目标站点的关键数据

Fig. 2 Key data of target website

3.2 获取数据与分析接口

进入调试模式，点击 Network 选项，可以搜索到相关 API 接口[12]。API 接口分析如图 3 所示，得到请求头部信息，信息包括目标地址(GET)、主机域名(Host)、用户代理(User-Agent)、上一级网页(Referer)、Cookie 信息(Cookie)等信息。通过 Get 方式传递参数，其中包含视频编号(aid)。在编写爬虫脚本时，需要伪造请求头部信息，防止被站点拦截。

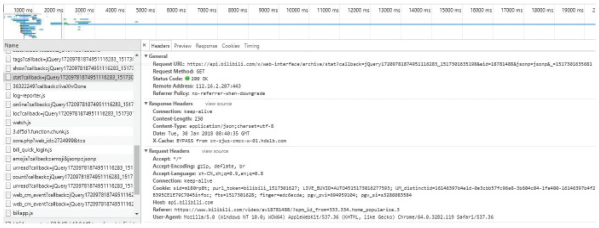

图 3 API 接口分析

Fig. 3 API analysis

得到的数据包以 JSON 类型返回，如图 4 所示。数据包包括 HTTP 状态码(code)、数据属性(data)、信息属性(message)与 TTL 属性。数据属性中不仅包括上述中提到的播放量(view)、弹幕数(danmaku)、捐赠投币数(coin)和收藏数(favorite)，还包括视频编号(aid)、评论数(reply)、分享次数(share)。

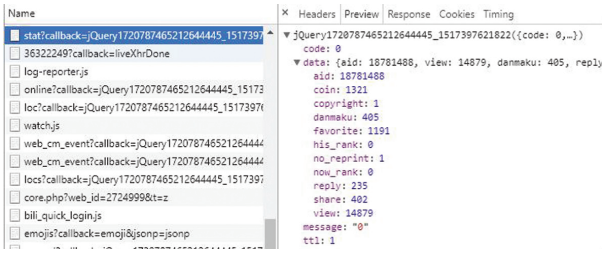

图 4 分析数据包相应信息

Fig. 4 Analyze packet information

3.3 编写爬虫脚本与保存数据

由于视频编号是一个随机数，所以需要从 1 开始循环到视频编号的最大值，且单次设置的最大值不宜过大，否则会导致内存溢出[13]。

```
urls = ["https://api.bilibili.com/x/web-interface/archive/stat?aid={}".format(i) for i in range(100000)]
```

头部请求只需包含用户代理，连接状态、主机地址等，其它信息可不携带[14]。

```
headers = {'User-Agent': 'Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, likeGecko) \Chrome/59.0.3071.115 Safari/537.36',
```

'Host': 'api.bilibili.com'}

因此，请求数据和函数构成如下：

```
data = requests.get(url, headers = headers, timeout = 5).json()
```

最终的爬虫脚本伪代码如下：

```
for url in urls:
    data = get(url, herders = herders).json()
    try:
        download(data)
        open bilibili.csv
        write data
        close bilibili.csv
```

获取的数据以 csv 类型文件保存。在爬取过程结束后，将数据保存至 MySQL 数据库中，截至日前共有 7 600 000 余条记录。以视频播放量为排序条件选取播放量最多的 100 个视频编号。通过视频编号浏览其具体视频页，抓取所在第一级分类和第二级分类具体信息。目标网站所属分类如图 5 所示。

图 5　目标网站所属分类

Fig. 5　Category of target website

3.4　统计数据信息

经过数据分析，根据第一级分类汇总，在视频播放量最多的 100 个视频中，国内外番剧共有 64 个，鬼畜有 16 个，音乐和舞蹈各有 6 个和 3 个，动画有 3 个，其它分类共 8 个，如图 6 所示。

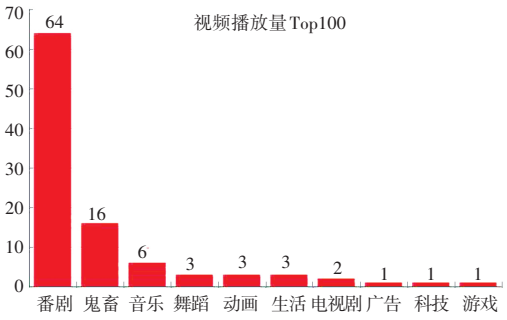

图 6　视频播放量 Top 100

Fig. 6　Video play amount Top 100

统计百分比中，国内外番剧占比最大，共占比 64%，鬼畜分类视频占比 16%，音乐占 6%，舞蹈、生活、动画各占 3%，其它分类共占 5%。

从数据中可以发现，哔哩哔哩弹幕网的用户最喜欢看国内外动漫番剧，在番剧占比 64% 中，其中日本动漫占比 58%，国创动漫占比 6%。由于日本动漫数量远大于国创动漫，因此日本动漫播放量占比最大。鬼畜视频多数由用户自行上传，主要提供用户欢乐和笑声，最受喜爱的视频中占据第二位。哔哩哔哩弹幕网的用户也喜欢音乐和舞蹈，对生活和动画制作这一块也有一定的兴趣。由此统计得到的结论，可以对网站首页的轮播板块设计提供参考。首推动漫视频与鬼畜视频，对音乐和舞蹈制定一定的推送量，对其它分类的视频分类减少推荐。

4　结束语

本文针对初级爬虫获取网页数据存在易于发现和速度慢等问题，利用 Python 的 Requests 库实现反反爬虫算法，并对其进行了技术原理分析，最后通过相关案例描述了反反爬虫技术的简单应用。文中实现的反反爬虫算法是基于 Requests 库开发，具有速度快的优点。但由于获取的数据信息量不够大，因此，下一步将对反反爬虫算法进行改进完善，并结合数据分析和人工智能开展实际案例分析和应用。

参考文献

[1] 刘智慧，张泉灵. 大数据技术研究综述 [J]. 浙江大学学报(工学版)，2014，48(6)：957-972.

[2] 安子建. 基于 Scrapy 框架的网络爬虫实现与数据抓取分析 [D]. 长春：吉林大学，2017.

[3] 邹科文，李达，邓婷敏，等. 网络爬虫针对“反爬”网站的爬取策略研究 [J]. 电脑知识与技术，2016，12(7)：61-63.

[4] 杨定中，赵刚，王泰. 网络爬虫在 Web 信息搜索与数据挖掘中应用 [J]. 计算机工程与设计，2009，30(24)：5658-5662.

[5] 赵本本，殷旭东，王伟. 基于 Scrapy 的 GitHub 数据爬虫 [J]. 电子技术与软件工程，2016(6)：199-202.

[6] 焦文华. 基于 Android 的移动互联网应用的研究和实现 [D]. 北京：北京邮电大学，2013.

[7] 谢克武. 大数据环境下基于 python 的网络爬虫技术 [J]. 电子制作，2017(9)：44-45.

[8] KANG Shulong，ZHANG Chuang，LIN Zhiqing，et al. Complexity research of massively microblogging based on human behaviors [C] //2010 2nd International Workshop on Database Technology and Applications，DBT A2010 —Proceedings. Wuhan，China：IEEE Computer Society，2010：1-4.

[9] BÖTTGER H，MΦLLER A，SCHWARTZBACH M I. Contracts for cooperation between Web service programmers and HTML designers [J]. Journal of Web Engineering，2006，5(1)：65-89.

[10] 吕林涛，万经华，周红芳. 基于 AJAX 的 Web 无刷新页面快速更新数据方法 [J]. 计算机应用研究，2006(11)：199-200，223.

[11] 熊文，熊淑华，孙旭，等. Ajax 技术在 Web2.0 网站设计中的应用研究 [J]. 计算机技术与发展，2012，22(3)：145-148.

[12] 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.

[13] RAMALHO L. Fluent Python [M]. United States：O'Reilly Media Inc，2015.

[14] JONES B，BEAZLEY D. Python Cookbook [M]. 3rd ed. United States：O'Reilly Media Inc，2016.