

# 基于 Python 的网络爬虫与反爬虫技术研究\*

李 培<sup>1,2</sup>

(1. 西安邮电大学计算机学院 西安 710121)(2. 西安邮电大学陕西省网络数据智能处理重点实验室 西安 710121)

**摘 要** 论文主要为网络爬虫的设计及实现、反爬虫技术的实现及相关技术的研究。通过研究目标网站爬虫门槛的协商及通过的条件,及反爬虫相关技术及最新发展。基于 Python 设计及实现一个完整的网络爬虫,最终完成了对目标网站所有文章数据的提取和存储,并借助对实验室内部网站的测试并实现了绕过反爬虫及反爬虫技术的研究,并对网络爬虫及反爬虫技术进行了理论说明和发展展望。

**关键词** 网络爬虫;Scrapy 框架;反爬虫

**中图分类号** TN711 **DOI**:10. 3969/j. issn. 1672-9722. 2019. 06. 028

## Research on Python-based Web Crawler and Anti-reptile Technology

LI Pei<sup>1,2</sup>

(1. School of Computer Science & Technology, Xi'an University of Posts & Telecommunications, Xi'an 710121)

(2. Shaanxi Provincial Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts & Telecommunications, Xi'an 710121)

**Abstract** This paper is mainly about the design and implementation of Web crawler, the implementation of anti reptile technology and related technology research. Through the study of target website crawler threshold negotiation and pass conditions, and anti reptile related technology and latest development, based on Python, a complete web crawler is designed and implemented. Finally, all the data of the target website are extracted and stored, and the research on the anti reptilian and anti reptilian technology is realized by the test with the web site of the laboratory. The theory and development trend of web crawler and anti crawler technology are also explained.

**Key Words** Web crawler, Scrapy frame, anti reptile

**Class Number** TN711

## 1 引言

网络爬虫是可以自动地大量抓取网页数据的计算机程序和脚本,别称:网络蠕虫、spider(网页蜘蛛)。网络爬虫的相关研究到现在为止,除了 Robots 这一“君子协定”外,并无相关的法律法规对其明显限制,反而是“大数据”的浪潮将网络爬虫的地位日渐上升。将来爬虫还会不断为人们的工作生活带来便利,为社会的发展提供知识的支持。网络爬虫一方需得注意自身行为,网站一方可在 Robots

协议上同各方达成默契,奉献出自己非核心数据,同时也是在为自己的发展提供窗口<sup>[1]</sup>。

## 2 网络爬虫的设计与实现

本研究在进行网络爬虫的相关研究时,提前已经明确了需要爬取的网页内容信息,如文章的题目(title),文章的 URL 以及创建日期、点赞数以及收藏数等。因此,实验中网络爬虫获取的数据并非是要窃取他人的劳动成果,而是对网络数据进行数据分析,形成结构化的数据。同时也是对繁杂的网络

\* 收稿日期:2018 年 12 月 16 日,修回日期:2019 年 1 月 29 日

**基金项目**:国家自然科学基金项目(编号:61105064);陕西省自然科学基金基础研究计划项目(编号:2016JM6085);陕西省教育厅科学研究计划项目“基于文本挖掘的网络社区情感倾向研究”(编号:17JK0687);陕西省普通高等学校重点学科专项资金建设项目资助。

**作者简介**:李培,女,硕士,副教授,研究方向:智能信息处理和网络安全。

数据的萃取、解读。虽说,数据众寡代表不了大数据的价值高低,但是大量的数据可以更有机会提取有价值的信息,带来更有意义的应用<sup>[2-3]</sup>。

## 2.1 框架设计

本研究选择使用数据库为MySQL,对应的连接数据库管理工具为navicat。编程语言使用目前最为流行的Python,网络爬虫实现的具体框架为Scrapy。Scrapy是基于Python开发的一个高层次的快速的网页抓取框架,用于抓取web站点信息并从页面中提取结构化的数据<sup>[4]</sup>。目前网站大多数都偏向是BeautifulSoup+requests,因此相对而言Scrapy是轻量级的,且Scrapy功能齐全。Scrapy相当于建房的框架,同时,Scrapy可以在需要的时候导入requests和beautifulsoup库进行使用。因此本文选择使用Scrapy框架完成网络爬虫的设计和实现<sup>[5]</sup>。

Scrapy框架的优势<sup>[6]</sup>有:

1)Scrapy框架是基于twisted而开发的。twisted是一个异步I/O的框架,异步I/O的特点使得在处理批量数据时效率更高,因此Scrapy性能也是非常高的。

2)Scrapy内置有css和xpath选择器,对获取的数据做selector非常方便,同时,selector也是相当强大,可以完全代替beautifulsoup。

3)Beautifulsoup是一个使用纯Python编写的库,而Scrapy中的lxml是使用C语言写的框库,前面我们已经对各种语言的速度做过描述,运行速度可达到上百倍。

## 2.2 目标分析与逻辑设计

在网络爬虫进行爬取网站的实验之前,我们需要弄清楚目标网站的域名结构和层次,这样才能更好地去编写代码逻辑,更有效地获得网页数据。分析研究的网站,其URL连接结构以主域名为中心,主域名下分有首页、资讯、文章等子域名,也就是导航部分。在子域名下是更详细的文章URL。因此,设计本网络爬虫爬取该网站数据的基本逻辑为

- 1)获取导航下的所有URL。
- 2)进入第一个导航获得所有文章标题URL。
- 3)进入第一个标题获取文章数据。

4)返回上一页获取下一个标题URL并进入获取详细文章的数据,直到获取完这个子域名下的所有文章数据。

- 5)返回首页对下一个子域名执行以上操作。

以上的逻辑存在一个问题,那就是环路。分析该网站结构就会发现获取文章所有URL时会回到首页。这时,就涉及到URL的去重<sup>[7]</sup>。

在爬虫的去重策略中,最简单的去重策略就是将所爬取到的URL存放到数据库中,当取到下一个URL时就数据库中进行了比对。如果取到了相同的URL时跳过,自动往下获取。由于每一次取到URL都得到数据库中比对,无疑大大降低了爬取效率,所以这种方法的使用是比较少的。通常使用的另一个方法。可以将取到的URL存放到内存中,在取出URL比对时就大大降低了耗费的时间。在将数据存到内存的同时,对URL进行md5编码。通常md5的编码长度为16byte,也就是说每个URL可以缩减到16byte,对内存的压力成倍减少。本文在对伯乐在线网站文章的爬取就采用这个去重策略。

### 2.2.1 爬取策略

我们可以以树形图详细地描述网站的域名层级结构,由于树形图比较复杂,为了方便对问题进行分析,因此可以简化成二叉树。通过二叉树遍历网站所有的URL,这样我们就可以获取到整站的内容。在设计爬虫逻辑时,为了更好地完成对网站数据的爬取,就需要通过算法更合理地对爬虫逻辑结构进行设计。网络爬虫的页面爬行策略有深度优先遍历,广度优先遍历和最佳优先遍历<sup>[8]</sup>。本文设计实现对该网站文章爬取的网页搜索策略使用的是深度优先算法。

### 2.2.2 设计思路

本文所设计实现的爬虫主要爬取某指定目标网站的所有文章,获取的内容包括文章标题、作者及URL、正文、日期、点赞数、收藏数、评论数、首页图、tag。通过对网站页面的分析。可以通过“最新文章”页面获得所有文章,而不需要对全站URL的爬取。因此,把这个页面作为目标页。

通过“最新文章”页面获取全部文章的策略有两个。第一种:对比页面跳转URL的变化,是在“page”后数字变化,因此可以通过对数字的递增而获得所有文章。第二种:获取“下一页”链接不断跳转,从而获取所有文章。本文设计的爬虫采用第二种方法。进入详情页可通过标题URL进入,本文通过首页图URL进入。

## 2.3 爬虫实现

前面已经对该目标网站的爬虫策略和逻辑做了设计,本节将对爬虫所涉及的相关操作做出介绍,包括如何对网页进行提取、数据库表的建立以及对程序中重要的方法进行讲解。同时,对获得的结果进行展示和简要分析。下面开始通过Scrapy爬取数据。

第一步:命令行窗口进入需要存放工程的目

录。如输入:

Scrapy startproject ArticleSpider

创建名为ArticleSpider的爬虫工程。

第二步:根据提示,输入:

Scrapy genspider jobbole http://blog.jobbole.com/all-posts/

创建爬虫主程序:jobbole.py

第三步:通过pycharm导入工程项目并配置项目的解释器即可在框架中编写自己的程序。本项目配置的解释器为Python 3.6.1。

可对网页进行数据提取的方式有三个:正则表达式、beautifulsoup以及lxml模块。前面章节中,已经介绍对Scrapy依赖库lxml的安装。且由于Scrapy的lxml模块底层是由C语言编写,在速度上占有较大优势。lxml库是用来对xml进行处理的第三方库。而同时xpath可以在xml中查找数据,类似SQL向数据库中查询数据。因此,xpath在爬虫中使用比较多。为了判断写下的语句是否正确,可新建用于启动爬虫的main文件,不断通过断点debug判断语句是否正确。如果每写一条语句就需要进行一次debug,也是影响了我们的开发效率。Scrapy为我们提供了一个shell的模式。

本爬虫采用xpath和css选择器及正则表达式共同提取网页数据。通过Chrome浏览器进入目标页面,右键页面选择“检查”可获得页面的源码。使用“查询属性”点击日期获取该元素属性。该元素在源码中的位置如图1所示。下面分别以一个实例来了解他们的实现方法。



图1 元素分析

Xpath:右键该元素,选择“copy xpath”得到:

//\*[ @id="post-113793"] /div[2] /p/text()[1]

解释:取任何节点,id=post-113793,往下的第二个div,往下的p,往下的第一个text。通过代码可获取到日期元素,同时,通过代码也可获取到日期create\_date值,取得的create\_date值是一个Selector对象,这样的数据是无法进行结构化的。因此,Selector提供了extract()方法,该方法可提取出数组。访问数组里的第一个值:create\_date.extract()[0],为防止数组为空,可使用使用strip()除去回车

换行符,取到值:2018/03/27,再用replace(“.”,“”)以空的字符串替换“.”,即可取到最终值:2018/03/27。

CSS选择器:通过Scrapy shell以css选择器的方式提取元素。通过搜索,p标签中的class值entry-meta-hide-on-mobile全局唯一,因此可通过这个值准确定位。然后直接取出text即可获得日期的值。

正则表达式:以CSS选择器的方式获取收藏数,得到:‘1 收藏’。而需要的是收藏数“1”,此时就必须使用正则的方式获取,具体如图2所示。

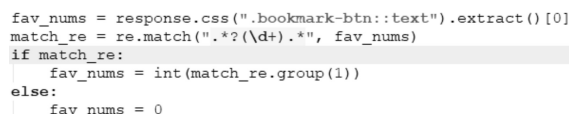


图2 正则的方式获取数据

表达式的含义为数字可出现多个,数字前可以有任何字符且可以出现任何多次,数字后也是可以有任意字符且可以出现任何多次。这样就形成了3个“group”,编号按序0、1、2,取group[1]得到数字“1”,“?”表示使用非贪婪的模式。

接下来,就需要获取文章列表页中首页图的URL,也就是可以进入文章详情页的URL,交给Scrapy解析、下载。

使用Scrapy的另一个类request,由from scrapy.http import Request进行库的导入。request可直接通过yield交给Scrapy下载器进行下载,下载完成,通过:“callback”回调到函数parse。

通过以上步骤已完成了一个基本的网络爬虫,对程序进行debug,其运行过程及记过如图3所示。对比网页数据,爬虫正确运行。



图3 网络爬虫爬取的结果

最终,已经完成了对所有文章URL的遍历及对文章相关字段的提取,将存储包括首页图的以及相关的字段。对图片的存储。

对图片的处理需要安装第三方库Pillow,使用mysql对相关字段进行存储。mysql及管理工具navicat已在前面章节介绍过。对数据库的数据插入可分为同步插入、异步插入。由于异步插入在效率上的优势,本爬虫使用的是异步插入的方式。



根据编写的 items.py 文件,对数据库进行设计。运行爬虫,在 navicat 中得到下图的结果。对目标网站所有文章爬取的任务成功。

id	url	url_object_id	create_date	fav	gravatar	comment_tags	content	front_image_url
100 个开源的工程项目	http://blog.jobbole.com/2018/03/06/100-open-source-projects/	55872281f1b0d34c	2018-03-06	5	2	0	5 网络,程序员,网络	<div class="entry">
20 个 OpenSSH 最佳安全实践	http://blog.jobbole.com/2018/03/03/20-openssh-best-practices/	751b4a769751d2c4c	2018-03-03	5	2	0	0 技术,Linux,OpenSSH	<div class="entry">
6 个开源的数据库自动化工具	http://blog.jobbole.com/2018/03/06/6-open-source-database-automation-tools/	47686898a578838	2018-03-06	1	2	0	0 技术,数据库,工具	<div class="entry">
从 0 开始学习使用 Redis 的 10 个案例	http://blog.jobbole.com/2018/03/03/10-redis-cases/	23891f806a3c52751	2018-03-03	0	1	0	0 技术,Redis,案例	<div class="entry">
4 个 Python 爬虫案例	http://blog.jobbole.com/2018/03/03/4-python-crawlers/	91164d4c5f6c2d40	2018-03-03	2	1	0	0 技术,Python	<div class="entry">
Linux 主目录中的隐藏文件有什么用	http://blog.jobbole.com/2018/03/07/7-linux-hidden-files/	9ec356e6c340e779e	2018-03-07	1	1	0	0 技术,Linux	<div class="entry">
Linux 系统过程分析	http://blog.jobbole.com/2018/03/06/6-linux-system-process-analysis/	2d0d326ed26d87f73	2018-03-06	1	1	0	0 技术,Linux	<div class="entry">
Linux 用户? 系统过程? 8 个重要的 Linux 命令	http://blog.jobbole.com/2018/03/06/8-linux-commands/	1401d806a3d81d34	2018-03-06	0	1	0	0 技术,Linux	<div class="entry">
Linux 系统之启动	http://blog.jobbole.com/2018/03/06/6-linux-system-start/	b48b820d1a88139d	2018-03-06	2	1	0	0 技术,Linux	<div class="entry">
Redis 集群部署与运维	http://blog.jobbole.com/2018/03/03/3-redis-cluster-deploy-and-maintain/	a736103d403a83f35a	2018-03-03	1	1	0	0 技术,Redis,部署	<div class="entry">
为什么 Linux 的目录结构是这样的	http://blog.jobbole.com/2018/03/06/6-why-linux-directory-structure-is-like-this/	a0f064e6b8f996813d	2018-03-06	1	2	0	0 技术,Linux	<div class="entry">
从 24 年独立开发者:大多数程序员	http://blog.jobbole.com/2018/03/06/6-from-24-years-independent-developer-most-programmers/	7a3f6d4e36071292b6	2018-03-06	1	1	0	0 网络,面试	<div class="entry">
使用 tar 和 rsync 来备份 Linux 性能	http://blog.jobbole.com/2018/03/06/6-using-tar-and-rsync-to-backup-linux-performance/	26d1f6ced402bb18	2018-03-06	1	1	0	0 技术,Linux	<div class="entry">
利用 Docker 实现分布式爬虫	http://blog.jobbole.com/2018/03/06/6-using-docker-to-implement-distributed-crawler/	8d4c6e7255230d0d	2018-03-06	3	1	4	0 技术,分布式	<div class="entry">
新到开源的爬虫工具	http://blog.jobbole.com/2018/03/06/6-new-open-source-crawler-tools/	742d4d7917326449	2018-03-06	5	2	0	0 C/C++,爬虫	<div class="entry">
面试: 为什么开发爬虫要用多线程	http://blog.jobbole.com/2018/03/06/6-interview-why-use-multi-thread-to-develop-crawler/	7f64591868f935ab	2018-03-06	0	1	0	0 技术,开发	<div class="entry">
在 Linux 中安装 Redis	http://blog.jobbole.com/2018/03/06/6-installing-redis-on-linux/	1c7d5802b140adefb1	2018-03-06	1	1	1	0 技术,Redis	<div class="entry">
在 Linux 中安装 Redis	http://blog.jobbole.com/2018/03/06/6-installing-redis-on-linux/	4d23500e0c7d51d71	2018-03-06	0	1	0	0 网络,面试	<div class="entry">
开源的爬虫小白 9 种神器	http://blog.jobbole.com/2018/03/06/6-open-source-crawler-tools/	a94d4d0606046e67	2018-03-06	2	1	1	1 业界,开源	<div class="entry">

图4 伯乐在线部分数据表

## 2.4 绕过反爬虫

针对爬虫,很多网站都有了反爬虫手段。建立网络爬虫的第一原则是:所有信息都是伪造的<sup>[9]</sup>。为了能更有效地获取网站数据,编写的爬虫就不得不带有绕过反爬虫的能力。绕过反爬虫的技术由爬虫技术的更新而发展的<sup>[10]</sup>。下面是本文使用的绕过发爬虫实现方法。

1)关闭 Robots 协议。即使 Robots 被称为双方的“君子协定”,然而很多情况下如果开启这个协议会让爬虫一无所获。Scrapy 框架已将 Robots 协议代码的实现写好,默认爬虫程序遵守该协议。修改方式:在 settings.py 将 ROBOTSTXT\_OBEY 的值由 True 修改为 False 即可。

2)设置超时。爬虫访问网页的速度是很快的,很容易被服务器检测到。代码:

```
import socket
socket.setdefaulttimeout(10)
```

可设置 10s 后连接超时,目的是为了爬虫操作更像是人的行为。

3)随机更换 user-agent。爬虫程序默认向对方发送的头文件 user-agent 值为 Scrapy,这就可以让对方轻松拦截访问。user-agent: 用户代理,简称 UA,用于服务器对用户的操作系统以及版本等信息,如: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0,方便用户便提供最好的服务,浏览器其实是一个代理。查看 Chrome 的 user\_agent,对网页打开“检查”,点击“Network”,选择一个响应,查看“header”中“Request Header”,即可获得该浏览器 user\_agent。

代码实现随机更换 user\_agent 可通过随机静态的 user\_agent 池和动态更换 user\_agent。本文实现的网络爬虫使用的是动态更换 user\_agent 的方式。fake-useragent 是基于 GitHub 开源的第三方库,用于随机更换 user\_agent。下载安装,使用代码:from

fake\_useragent import UserAgent 导入,ua.random 实现对 user\_agent 的随机获取。

4)代理 IP。用户访问对方网站,对方的服务器会记录用户的访问日志,可通过 IP 识别并记录用户。如果访问者行为异常,对方很容易将其 IP 拉黑。IP 就像我们的个人身份证,唯一标识自己,没了 IP 就无法进行网络活动,因此,IP 资源是非常珍贵的。伪装 IP,同样有静态与动态的方法。动态获取代理存储在数据库中,再随机向数据库中取出可用 IP。为验证代理 IP 是否生效,需要获得对方服务器的访问记录,但目标网站的访问记录本人无法得到,故选择实验室内部网站服务器进行该功能的测试,区域 1 位使用本机正常访问该网站,区域 2 是加入了 IP 代理的爬虫访问记录<sup>[11-12]</sup>。IP 代理成功生效。

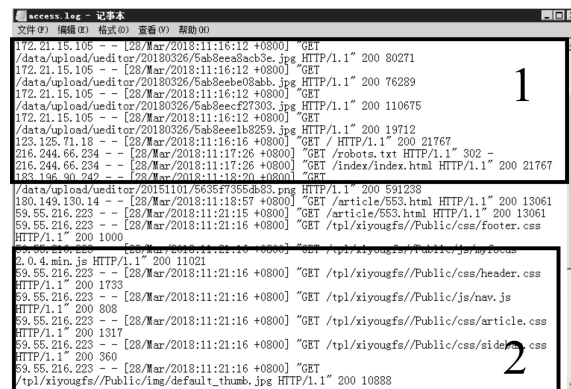


图5 服务器访问记录

## 3 反爬虫技术的研究与实现

### 3.1 反爬虫技术的背景

网络爬虫技术的出现,无疑为人们的生活带来方便,也为社会的发展推波助澜。人们有需求,就会有市场,有了市场,就回吸引到大量“淘金者”的加入。这也使得爬虫大量增加。不同的种类的爬虫,技术含量也参差不齐,爬虫市场开始混乱、泛滥。在信息时代,我们如果综合各种反爬虫方法,可以很大程度上缓解爬虫对网站造成的负面影响,保证网站的正常访问,下面将会对反爬虫技术进行详细介绍。

### 3.2 反爬虫技术设计及实现

反爬虫有很多的方法,也分有好层次。本研究使用的爬虫包含的绕过爬虫的方法,分别是修改 robots 协议、访问频率阈值、更换 user-agent、使用代理 IP。与之相对,反爬虫策略为敦促对方遵守协议、做好访问日志的检查分析,并对不正常操作进行清除、反制<sup>[13]</sup>。

### 3.2.1 启动 Robots 协议

网络爬虫排除协议(是网站用来告诉爬虫哪些页面可以爬取哪些页面涉及隐私不能被爬取的标准)<sup>[14]</sup>。前面章节提到,在编写爬虫时,都需要修改 settings.py 程序,将 ROBOTSTXT\_OBEY 的值 True 修改为 False。这就是在修改 Robots 协议。Robots 协议是网络爬虫在获取网页数据需要面对的第一道关卡。然而现在的网络上搜索引擎众多,大多数为良性获取数据,也有利于网站的发展,倘若书写失误,将会导致网站在搜索引擎消失。网络上有专门的 Robots 协议自动生成网站:站长工具(<http://tool.chinaz.com/robots/>)。其生成结果如图 6 所示。

```
# robots.txt generated at http://tool.chinaz.com/robots/
User-agent: Teoma
Disallow: /
User-agent: twiceler
Disallow: /
User-agent: MSNBot
Disallow: /
User-agent: Gigabot
Disallow: /
User-agent: yahoo-mmcrawler
Disallow: /
User-agent: yahoo-blogs/v3.9
Disallow: /
User-agent: *
Disallow: /
Crawl-delay: 5
```

图 6 生成的结果

将所生成的结果保存到新建的记事本,并将其重命名为 Robots.txt。最后把此文件复制粘贴到网站的根目录即可实现该协议的开启。

### 3.2.2 以设置阈值的方式挑选非正常访问

各家饱受爬虫困扰的网站也在积极应对,通常网站自己的手段可以是基于网络防火墙依靠 IP 识别做阻断,误伤概率较高<sup>[15]</sup>。

设置单位阈值目的在于识别。单位阈值:在任何单位时间内所允许的最高或最低值。普通用户正常访问是有限的,对于普通网站,单个访问次数不会过百。对于淘宝的访问,一秒钟的访问最多为也不会过千,当然,每个网站该由自己的实际情况实时准确地更改自己的单位阈值。倘若某一 IP、user\_agent 标识超出了阈值,对其实施监控,当判断对方行为非正常用户所为,且对己方利益进行了侵害,即可对其进行封禁。但这也难免会误伤到用户。在我们获取到非正常用户时,还可以重定向。将用户页面返回到验证页面。常见的有验证码验证、语音验证、短信验证、邮箱验证。针对代理 IP,反爬虫一方当然也可编写爬虫提前爬取(如西刺代理)所有的代理并在己方服务器将其封禁,这就可以省去很多工作。在对封禁名单操作中,由网站管理者主动操作的有四种:分别是查看封禁 IP,添加封禁 IP,删除封禁,IP 以及修改封禁时间,其系统本身应该还具有分时统计<sup>[16]</sup>。

### 3.2.3 增加登录以限制

在对知乎的数据爬取中,对网站进行 Scrapy shell 返回信息中返回状态码为 500,报错:Internal Server Error。对相关字段进行提取,返回的数组也是为空。伯乐在线的返回信息中返回状态码为 200。原因就在于访问知乎需要注册登录,由此可见,增加了注册登录是可以有效阻止爬虫访问的。

### 3.2.4 AJAX 动态加载数据

由于 AJAX 异步的特点,网站的数据更新可以不和前段网页数据同步。另外,我们编写爬虫提取字段时根据的是 HTML 代码,而在网页中,JS 是根据用户的操作进行 JS 渲染相应,代码是看不到的。因此在爬虫想要获取返回的数据时,还必须得去解析 HTML 源码。这无疑为爬虫设置了技术门槛,要么去学习,要么放弃。同时解释 HTML 还需要较多的时间。为对方造成麻烦就是己方的胜利。

### 3.2.5 以谋略取胜

技术层面的爬虫与反爬虫,都会被相互攻破,而谋略上的反爬虫却可以让自己由被动方变为主动方。其一,长时间的反爬虫经验可以总结出适合自己的,与其他常见不一样的反爬虫策略,这样的方法往往不适用于其他,只属于自己,这样除非是常常交手的双方,而其他人却不会知道对方的套路,无从下手。这些是小套路,虽然与众不同,但从属于以下的框架下<sup>[17]</sup>。

#### 1) 投毒

爬虫在于获取数据,倘若对对方展示了亦真亦假的数据,而对方在无法甄别的情况下使用了己方提供的数据。假设淘宝和京东都在为同一产品的市场进行争夺,价格无疑成为了目标。己方为淘宝,明天 0 点就到了产品正式发售,京东可以晚几分钟上架,在这几分钟内获取了你的价格,这就会让你的价格没了优势,消费者会因为这一点的区别而选择京东。己方的做法是按以前的经验,就在这几分钟向京东程序员提供己方准备好的页面,给他假的数据,然而当其发现时已经来不及了。

#### 2) 心理战

每当我们的每一次成功,对对方程序员进行嘲笑、侮辱等,目的让其崩溃。而失败了可以让他满足虚荣心,此时就会寻找他的空子,如从他的炫耀和嘲讽里就很有可能挖掘出他的策略,最后进行反击。

#### 3) 欲纵故擒

让爬虫感受不到任何威胁,这时爬虫的程序员就放手了对你这个网页的关注,反而是和对手刚上

了,无暇对你网站数据进行查验。而此时,你就可以为所欲为的欺骗爬虫。

## 4 结语

网络爬虫技术与反爬虫技术的研究到这里就完全结束了。或许大家会在以上的描述中感到有些疑惑,它们最终到底是谁存谁亡,谁强谁弱,谁成为了胜利者。然而并没有。爬虫与反爬虫互为矛盾,相生相克。它们之间的会因为技术的不断更新成长,双方不断的打出自己手中的牌,为了资源利益在这个没有硝烟的战场不断战斗。反爬虫,最终会成为双方程序员的尊严战斗,没有成败,没有输赢。

在大多数的资料中,并没有将 Robots 协议列为反爬虫策略中,而本文将 Robots 列为反爬虫策略,需要说明的是,反的爬虫该是危害、变异了的爬虫。变异爬虫,不以爬取公共数据为目的,不成熟、不完善、高伤害的爬虫。爬虫与反爬虫的战争该为变异爬虫与反爬虫的战争。Robots 该作为健康爬虫与反爬虫之间的停战协议。以上的爬虫使用相关技术只为让自己变得完善,而完善的爬虫时不具备给对方带来伤害的。

相互尊重,不涉及核心资源,不为己方而伤害对方。共同在网络世界里完成自己的梦想,取得成绩,为网络的发展增砖添瓦,为人们的生活提供便利。

## 参考文献

- [1] 漆志辉,杨天奇. 网络爬虫性能研究[J]. 北京:微型机与应用,2011,24(05):72-74.  
QI Zhihui, YANG Tianqi. Research on Web Crawler Performance [J]. Beijing: Microcomputers and applications, 2011, 24 (05):72-74.
- [2] 潘昊. 海量网络信息实时抽取引擎设计与实现[D]. 北京:北京邮电大学,2016.  
PAN Hao. Design and Implementation of Massive Network Information Real-time Extraction Engine [D]. Beijing: Beijing University of Posts and Telecommunications, 2016.
- [3] 胡博. 基于网络爬虫的内容资源评价研究[D]. 北京:北京理工大学,2015.  
HU Bo. Content Resource Assessment Based on Web Crawler [D]. Beijing: Beijing Institute of Technology, 2015.
- [4] 米硕,孙瑞彬,李欣,等. Scrapy 分布式爬虫原理分析与概述[J]. 北京:中国新通信,2018,20(04):234.  
MI Shuo, SUN Ruibin, LI Xin, et al. Scrapy Distributed Crawler Principle Analysis and Overview [J]. Beijing: China New Communications, 2018,20 (04): 234.
- [5] 施威,夏斌. 基于 Scrapy 的商品评价获取系统设计[J]. 北京:微型机与应用. 2017,19(04): 12-15.  
SHI Wei, XIA Bin. Design of Commodity Evaluation Acquisition System Based on Scrapy [J]. Beijing: Microcomputers and Applications, 2017,19 (04): 12-15.
- [6] 李代伟,谢丽艳,钱慎一,等. 基于 Scrapy 的分布式爬虫系统的设计与实现[J]. 湖北恩施:湖北民族学院学报(自然科学版),2017,18(03): 317-322.  
LI Daiyi, XIE Liyan, QIAN Shenyi, et al. Design and Implementation of Scrapy based Distributed Crawler System [J]. Enshi, Hubei Province: Journal of Hubei Institute for Nationalities (Natural Science Edition), 2017, 18 (03): 317-322.
- [7] 王敏. 分布式网络爬虫的研究与实现[D]. 南京:东南大学,2017.  
WANG Min. Research and Implementation of Distributed Web Crawler[D]. Nanjing: Southeast University, 2017.
- [8] 赵鹏程. 分布式书籍网络爬虫系统的设计与实现[D]. 成都:西南交通大学,2014.  
ZHAO Pengcheng. Design and Implementation of Distributed Book Crawler System [D]. Chengdu: Southwest Jiao Tong University, 2014.
- [9] 晁望,荣胜彩,费宇貂,等. 基于 Scrapy 的主题爬虫设计与实现[J]. 澳大利亚:先进材料研究,2014, 03(850): 237-238.  
CHAO Wang, RONG Shengcai, FEI Yudiaio, et al. Design and Implementation of the Topic-Focused Crawler Based on Scrapy [J]. Australia: Advanced Materials Research, 2014 03(850):237-238.
- [10] 刘毅. 网站反爬取机制的研究与应用[D]. 北京:北京邮电大学,2017.  
LIU Yi. Research and Application of Website Anti Crawling Mechanism [D]. Beijing: Beijing University of Posts and Telecommunications, 2017.
- [11] Ryan Mitchell. python 网络数据采集[J]. 北京:人民邮电出版社,2016,15(06):174.  
Ryan Mitchell. Python Studio Studio [J]. Beijing: People Post and Telecommunications Press, 2016, 15 (06): 174.
- [12] 陈利婷. 大数据时代的反爬虫技术[J]. 长沙:电脑与信息技术,2016,21(06):60-61.  
CHEN Liting. Anti Reptile Technology in the Age of Big Data [J]. Changsha: Computer and Information Technology, 2016,21 (06): 60-61.
- [13] 鲁萍. 带你进入网络爬虫与反爬虫的世界[J]. 北京: (下转第 1496 页)



- European Conference on Computer Vision. Springer International Publishing, 2016:809–825.
- [16] Zhao R, Ouyang W, Li H, et al. Saliency detection by multi-context deep learning[C]//Computer Vision & Pattern Recognition, 2015:1265–1274.
- [17] Pan H, Jiang H. A Deep Learning Based Fast Image Saliency Detection Algorithm[J]. Computer Vision and Pattern Recognition, 2016.
- [18] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6):1137–1149.
- [19] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder–Decoder Architecture for Scene Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):1–1.
- [20] Li J, Cheng J H, Shi J Y, et al. Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement [M]. Advances in Computer Science and Information Engineering. Springer Berlin Heidelberg, 2012:553–558.
- [21] Shen L, Lin Z, Huang Q. Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks [J]. Computer Science, 2015:467–482.
- [22] Lecun Y, Cortes C. The mnist database of handwritten digits[J]. 2010.

(上接第 1420 页)

- 软件和集成电路, 2016, 17(12): 12–14.
- LU Ping. Take You into the World of Reptiles and Reptiles [J]. Beijing: Software and Integrated Circuits, 2016, 17(12): 12–14.
- [14] 刘毅. 网站反爬取机制的研究与应用[D]. 北京: 北京邮电大学, 2017.
- LIU Yi. Research and Application of Website Anti Crawling Mechanism [D]. Beijing: Beijing University of Posts and Telecommunications, 2017.
- [15] 陆文. 十分钟解决爬虫问题. 超轻量级反爬虫方案 [J]. 石家庄: 计算机与网络, 2017, 21(17): 58–60.
- LU Wen. Ten Minute Resolution of Crawler Problem. Ultra Lightweight Anti reptile Program [J]. Shijiazhuang: Computer and Network, 2017, 21 (17): 58–60.
- [16] 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]. 安徽: 电脑知识与技术, 2016, 23(07): 61–63.
- ZOU Kewen, LI Da, DENG Tingmin, et al. Crawling strategy of web crawler against "anti climbing" website [J]. Anhui: computer knowledge and technology, 2016, 23(07): 61–63.
- [17] 威廉. 哈丁, 安妮塔. 里德, 罗伯特. 盖瑞. Cookies 和网站错误: 它们是什么以及如何一起工作[J]. 美国 费城: 信息系统管理, 2001, 6(3): 321–323.
- William T. Harding, Anita J. Reed, Robert L. Gray. Cookies and Web Bugs: What They are and How They Work Together [J]. USA Philadelphia: Information Systems Management, 2001, 6(3): 321–323.

## 版 权 声 明

本刊已许可万方数据库、中国学术期刊(光盘版)电子杂志社在中国知网及其系列数据库等产品中以数字化方式复制、汇编、发行、信息网络传播本刊全文。著作权使用费与本刊稿酬一并支付。作者向本刊提交文章发表的行为即视为同意我编辑部上述声明。

《计算机与数字工程》编辑部