Universe Scientific Publishing
Computer System Network and Telecommunications,计算机系统网络和电信(1) 2019,3
ISSN:2661-3719(Print) ISBN:2661-3727(Online)

浅析网络爬虫的关键技术

毛红霞
四川大学锦城学院 计算机与软件学院 四川 成都 611731
DOI:10.18686/jsjxt.v1i3.1277

【摘要】大数据已经渗透到当今每一个行业和业务职能领域,成为重要的生产因素。在海量数据中高效率准确地获取有效数据,越来越得到关注。运用网络爬虫技术能够快速、高效地获取大量数据。本文介绍了基于 Python 的网络爬虫所涉及到的关键技术:爬虫架构、爬取策略、Python 中实现 HTTP 请求、Python 中实现网页解析、Python 爬虫框架。并针对网站的常用的反爬策略制定了相应的应对措施。
【关键词】Python;网络爬虫;爬取策略

0 引言

在大数据时代,网络所产生的数据正以前所未有的速度增长。数据蕴含着巨大的价值,充分挖掘数据潜在的价值,对我们的生活、工作都有很大的帮助。面对互联网这样一个数据的海洋,如何高效准确地获取有效的数据是人们最关心的问题。因此,诞生了网络爬虫这项技术。网络爬虫的应用十分广泛,不仅应用在搜索引擎上,普通用户和企业在获取数据的时候都需要借助于网络爬虫。本文将基于 Python 的网络爬虫所涉及到的一些技术进行讨论。

1 爬虫原理

网络爬虫就是一种程序,用来收集数据。网络爬虫可以自动化地浏览网络中的信息,然后根据指定的规则下载和提取信息。由于爬虫是一种程序,程序的运行速度极快,而且不会因为做重复的事情而感到疲惫,使用爬虫来获取大量数据,会大大提高数据获取的效率。

网络爬虫按照系统结构和实现技术,大致可以分为以下几种类型:通用网络爬虫、聚焦网络爬虫、增量式网络爬虫、深层页面爬虫。实际项目运用的网络爬虫系统通常是几种爬虫技术相结合而实现的。

2 爬虫架构

编写网络爬虫的主要目的就是从目标网站中将网页下载到本地并提取所需的数据。为了完成这两个任务,一个简单的网络爬虫就要包含如图 1 所示的 4 个部分。

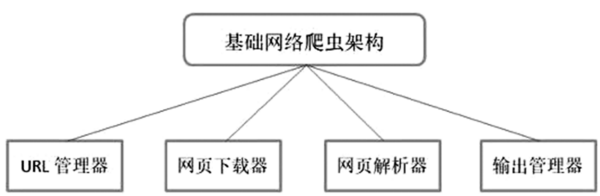

图 1 基础网络爬虫的架构

URL 管理器:管理将要爬取的 URL,防止重复抓取和循环抓取。

网页下载器:这是下载网页的组件,用来将互联网上 URL 对应的网页下载到本地,是爬虫的核心部分之一。

网页解析器:这是解析网页的组件,用来从网页中提取有价值的数据,是爬虫的另一个核心部分。

输出管理器:这是保存信息的组件,用来把解析出来的内容输出到文件或、数据库中。

3 基于 Python 的网络爬虫技术

3.1 Python 中实现 HTTP 请求

网络爬虫需要将所需数据所在的网页下载到本地,那么网页下载就是网络爬虫必备的部分之一。下载网页就需要实现 HTTP 请求,在 Python 中实现 HTTP 请求比较常用的库是 Requests 库。Requests 继承了 urllib2 的所有特性,比 urllib 更加方便。Requests 支持 HTTP 连接保持和连接池,支持使用 cookie 保持会话,支持文件上传,支持自动确定响应内容的编码,支持国际化的 URL 和 POST 数据自动编码。

3.2 Python 中实现网页解析

网络爬虫需要将下载到本地的网页进行页面解析,从 HTML 网页信息中提取所需要的、有价值的数据和链接。在 Python 中实现网页解析的工具主要有:

正则表达式:描述了一种字符串匹配的模式,可以用来检查一个字符串是否含有某种子串,将匹配的子串替换或者从某个串中取出符合某个条件的子串。

Lxml 库:这个库使用的是 XPath 语法,XPath 是一门在 XML 文档中查找信息的语言,XPath 可用来在 XML 文档中对元素和属性进行遍历,通常配合 Chrome 浏览器一起使用,非常方便简单,是解析网

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

页的主力工具。

Beautiful Soup:这是一个可以从 HTML 或 XML 文档中提取数据的 Python 库,能够通过转换器实现惯用文档导航、查找。Beautiful Soup 编写效率高,能帮助程序员提高工作效率,且简单易学,但相比正则表达式和 Lxml,其对网页解析的速度较慢。

3.3 Python 爬虫框架

Python 中有很多帮助实现爬虫项目的框架,爬虫框架允许根据具体项目的情况,调用框架的接口,编写少量代码就可以实现爬虫。爬虫框架实现了爬虫要实现的常用功能,能够节省开发爬虫项目的时间,提高开发爬虫的效率。Python 爬虫框架中最常用的是 Scrapy 框架。Scrapy 框架是一个相对成熟的框架,可以应用在包括数据挖掘、信息处理或存储历史数据等一系列的程序中,图 2 为 Scrapy 框架的架构图。

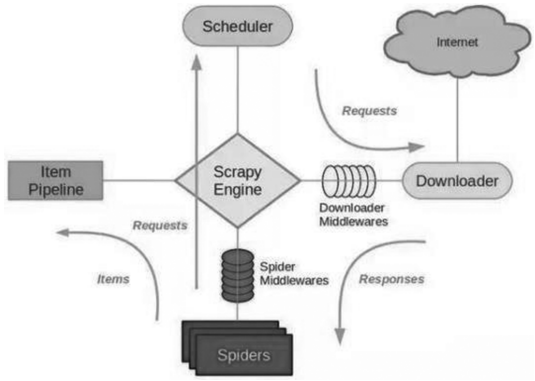

图 2 Scrapy 框架的架构图

(1)调度器(Scheduler):用来接受引擎发过来的请求,压入队列中,并在引擎再次请求的时候返回。实际上就是一个 URL(抓取网页的网址或者说是链接)的优先队列,由它来决定下一个要抓取的网址是什么,同时去除重复的网址。用户可以自己的需求定制调度器。

(2)下载器(Downloader):用于下载网页内容,并将网页内容返回给 Spider。

(3)爬虫(Spider):是用户最关心的部份。用户定制自己的爬虫,用于从特定的网页中提取自己需要的信息,即所谓的实体(Item)。用户也可以从中提取出链接,让 Scrapy 继续抓取下一个页面。

(4)实体管道(Item Pipeline):用于处理爬虫提取的实体。主要的功能是持久化实体、验证实体的有效性、清除不需要的信息。

(5)Scrapy 引擎(Scrapy Engine):是整个框架的核心.它用来控制调试器、下载器、爬虫,控制着整个流程。

4 应对反爬虫策略

因为搜索引擎的流行,网络爬虫已经成了很普及网络技术,对于一些网站来说,受到网络爬虫的光顾是不可避免的。但是由于一些网络爬虫会造成网站访问压力非常大,会导致网站访问速度缓慢,甚至服务器瘫痪无法访问。因此大部分网站为了避免网站被爬取,增加了各种各样的反爬虫措施。如果想要从这些设置了反爬的网站中爬取所需数据就必须要应对网站的反爬虫策略。

4.1 常用的网站反爬虫策略

常用的网站反爬虫策略有以下几种:

(1)通过 Headers 反爬虫

通过识别用户请求的 Headers 反爬虫是最常用的网站服务器反爬虫策略。很多网站都会对 HTTP 请求头部的 User-Agent 进行检测,还有一部分网站会对 Referer 进行检测(一些资源网站的防盗链就是检测 Referer)。

(2)基于用户行为反爬虫

还有一部分网站是通过检测用户行为,例如同一 IP 短时间内多次访问同一页面,或者同一账户短时间内多次进行相同操作,这些操作都会引发网站采取反爬虫措施。

(3)采用动态页面反爬虫

有一部分网站,我们需要爬取的数据是通过 ajax 请求得到,或者通过 JavaScript 生成的,这样就会对网络爬虫产生一定的困难。

4.2 应对网站反爬虫的策略

(1)为爬虫设置 Headers

在遇到了通过检测 Headers 反爬虫的注站,可以直接在爬虫代码中添加 Headers,将浏览器的 User-Agent 复制到爬虫的 Headers 中;或者将 Referer 值修改为目标网站域名。

(2)使用 IP 代理或加大请求间隔时间

针对网站检测 IP 访问的情况,使用 IP 代理就可以解决。可以专门写一个爬虫,爬取网上公开的代理 IP,对 IP 逐一检测,将有效的 IP 起来。在爬虫代码中,可以每请求几次更换一个 IP,这样就能很容易的绕过这种反爬虫策略。

遇到检测统一账户浏览行为的爬虫时,可以在每次请求后随机间隔几秒再进行下一次请求。有些有逻辑漏洞的网站,可以通过请求几次,退出登录,重新登录,继续请求来绕过同一账号短时间内不能多次进行相同请求的限制。

(3)使用 Selenium 框架

针对动态页面,首先用 Firebug 或者 HttpFox 对网络请求进行分析。如果能够找到 ajax 请求,也能分析出具体的参数和响应的具体含义,我们就能采用上面的方法,直接利用 requests 或者 urllib2 模拟 ajax 请求,对响应的 json 进行分析得到需要的数据。

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

但是有些网站把 ajax 请求的所有参数全部加密了,用户根本没办法构造自己所需要的数据的请求,此时就可以使用 Selenium 框架,调用浏览器内核,并利用 phantomJS 执行 JS 来模拟人为操作以及触发页面中的 JS 脚本。用这套框架几乎能绕过大多数的反爬虫策略。

5 网络爬虫的爬取策略

网站的爬取一般有两种爬取策略:深度优先和广度优先。

深度优先:是指网络爬虫会从起始页开始,一个链接一个链接跟踪下去,处理完这条线路之后再转入下一个起始页,继续跟踪链接。这个方法有个优点是网络爬虫在设计的时候比较容易。深度优先策略不一定能适用于所有情况,深度优先如果误入无穷分枝(深度无限),则不可能找到目标节点。

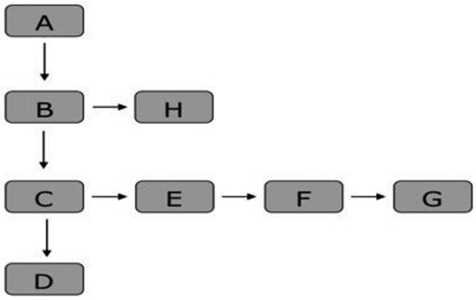

图 3 深度优先策略示意图

广度优先:是指网络爬虫会先抓取起始网页中链接的所有网页,然后再选择其中的一个链接网页,继续抓取在此网页中链接的所有网页。这是最常用的方式,因为这个方法可以让网络爬虫并行处理,提高其抓取速度。广度优先遍历策略属于盲目搜索,它并不考虑结果存在的可能位置,会彻底地搜索整张图,因而效率较低,但是,如果想要尽可能的爬取较多的网页,广度优先搜索方法是较好的选择。

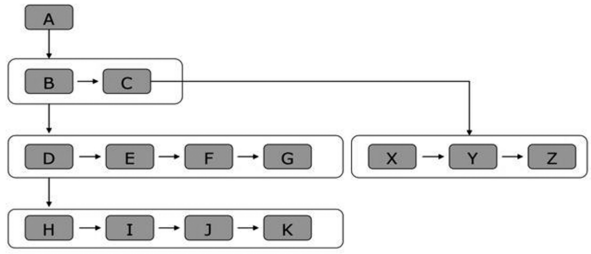

图 4 广度优先策略示意图

6 结束语

在大数据时代,信息的采集是一项重要的工作,如果单纯靠人力进行信息采集,不仅低效繁琐,搜集的成本也会提高,网络爬虫可以代替人们自动地在互联网中进行数据信息的采集与整理。网络爬虫有着非常广泛的用途,因此掌握爬虫技术显得尤为重要。要掌握好网络爬虫这项技术,就需要了解网络爬虫的原理、基础网络爬虫的架构以及爬取策略。本文讲述了基于 Python 的网络爬虫的相关技术,并针对网站的常用的反爬策略制定了相应的应对措施。网络爬虫的开发者需要随时关注所爬取的网站的网页结构,爬虫代码需要根据网页的变化而变化,才能确保爬虫能够正常爬取所需数据。

【参考文献】

[1]蜘蛛抓取策略:广度优先和深度优先分析.
https://blog.csdn.net/zhouhaisunny/article/details/78698448

[2]唐琳,董依萌,何天宇. 基于 Python 的网络爬虫技术的关键性问题探索[J]. 电子世界,2018(14):32-33

[3]刘贵平,刘娜,段红义. 基于聚焦网络爬虫技术的人才招聘数据采集[J]. 电脑编程技巧与维护,2018(05):69-71

[4]刘顺程,岳思颖. 大数据时代下基于 Python 的网络信息爬取技术[J]. 电子技术局与软件工程,2017(21):160-160

[5]李琳. 基于 Python 的网络爬虫系统的设计与实现[J]. 信息通信,2017(9):26-27

[6]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