基于 Java 的新浪微博爬虫研究与实现

陈 珂,蓝鼎栋,柯文德,黎树俊,邓文天
(广东石油化工学院 计算机与电子信息学院,广东 茂名 525000)

摘 要:为了高效获取更多的微博数据,针对调用微博 API 和网页版(com 版)等传统微博爬虫在数据采集中所存在的问题,设计开发了一个基于 Java 的采集新浪微博 Weibo.cn 站点的网络爬虫系统。该系统通过广度遍历结合组拼 URL 的方式采集网页源码,使网页源码更加简洁,纯净度更高,降低了网络传输压力并减少了 HTML 源码解析时间。主要实现了微博模拟登陆、微博网页爬取、微博页面数据提取和任务调度控制,并对爬取数据进行了分析,在爬虫中添加了主题微博筛选功能。为验证该系统的有效性和可行性,与其他传统方法进行了分析对比。实验验结果表明,所提出的系统爬取效率更高,实现代码更简便。

关键词:新浪微博;网络爬虫;Java;数据挖掘
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2017)09-0191-06
doi:10.3969/j.issn.1673-629X.2017.09.042

Research and Realization of Weibo Crawler with Java

CHEN Ke,LAN Ding-dong,KE Wen-de,LI Shu-jun,DENG Wen-tian
(College of Computer and Electronic Information,Guangdong University of Petrochemical Technology,Maoming 525000,China)

Abstract:In order to obtain more microblog data efficiently,a Java-based acquisition system of Sina is designed and developed for Weibo API,traditional crawler and Web version (com version),by which Weibo.cn Web site crawler system has been established through the breadth combination of traverse combination to collect web page source code and thus the page source code is more concise and purer,reducing network transmission pressure and the HTML source code analysis time. It mainly realizes the Weibo simulated logging,Weibo web crawling,Weibo page data extraction and task scheduling control,and analyzes the crawling data. The theme Weibo selection is added in the crawler. To verify its effectiveness and feasibility,the analysis and comparison is made with other traditional methods. The experimental results show that it is of higher efficiency with simpler code.

Key words:Sina Weibo;Web crawler;Java;data mining

0 引 言

随着计算机与网络技术的快速发展,社交网络平台是人们喜爱的网络社交方式。目前广为流行的微博系统,对人们的生活方式影响巨大。伴随着微博使用人数的急剧上升,产生了巨大的数据量,由此可以从中挖掘出大量的有用信息,而基于微博的数据挖掘研究已成为当今社会科学和计算机科学研究的重点。微博(Weibo)是一种通过关注机制分享简短实时信息的广播式的社交网络平台,也是一个基于用户关系的信息分享、传播以及获取的平台[1]。用户可以通过 WEB、WAP 等各种客户端组建个人社区,以 140 字(包括标点符号)的文字更新信息,并实现即时分享。微博的关注机制分为可单向、可双向两种。微博作为一种分享和交流平台,其更注重时效性和随意性。

比较知名的微博系统是 2006 年推出的 Twitter,人们对它进行了一系列的研究与数据分析,特别是利用数据挖掘技术从中获取了大量有用信息。2009 年新浪微博系统推出,它提供的微博 API 的使用限制很多,

收稿日期:2016-08-22 **修回日期**:2016-11-24 **网络出版时间**:2017-07-11
基金项目:国家级大学生创新创业训练计划项目(201411656017,201611656002,201611656029,2016pyA033);广东省自然科学基金(2016A030307049);广东省高等学校学科与专业建设专项资金科研类项目(2013KJCX0132);广东省云机器人(石油化工)工程技术研究中心开放基金项目(650007)
作者简介:陈 珂(1964-),男,硕士,教授,研究方向为 Web 数据挖掘、信息检索;蓝鼎栋(1992-),男,研究方向为 Web 数据挖掘、智能信息系统。
万方数据
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20170711.1452.010.html

最大的问题是调用次数少,测试授权单个 IP 每小时只能请求 1 000 次,而且每次调用时都是获取一个用户的某个数据集,虽然利于获取单个用户的大量数据集,但不利于广度抓取全网用户的最新信息。当然传统爬虫有广度爬取页面的能力,但是微博需要用户进行登录验证才有权限访问大量的微博页面,并且传统爬虫获取新 URL 的方式一般是通过 HTML 源码中提取[2-4]。所构建的微博爬虫是通过用户 ID 获取,可以进行 URL 组拼,数据是动态加载的,要判断是否存在下一页。所以,为了更快速获取大量更有用的微博数据,必须编写特定需求的微博网络爬虫系统。目前基于 weibo. cn 数据采集方面的研究较少,针对调用微博 API、传统爬虫、网页版(com 版)微博爬虫存在的问题,设计开发了一个基于 Java 新浪微博 weibo. cn 站点的网络爬虫系统[5-6]。

1 微博爬虫设计

1.1 微博爬虫的主要结构与流程

所用的开源工具包或组件有:org. seleniumhq. selenium-java-2. 45. 0 组件(内含 htmlunit-2. 15. jar,java 页面分析工具包),org. apache. httpcomponents-httpclient-4. 3. 5(内含 httpclient-4. 3. 5. jar,支持 HTTP 协议的客户端编程工具包),org. htmlparser-2. 1(内含 htmlparser-2. 1. jar,html 解析工具包);辅助工具有浏览器 Firefox 和它的网站调试工具 Firebug。

微博爬虫主要功能流程(见图 1)说明如下:

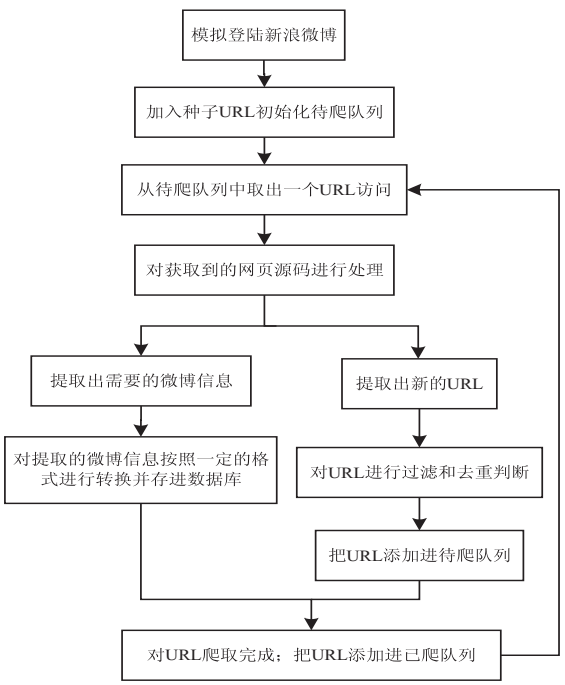

图 1 微博爬虫主要功能流程图

(1)通过 htmlunit 模拟登陆 weibo. cn 站点,登陆成功后会获取到名为 gsid_CTandWM 的 cookie 字段;

(2)把所有返回的 cookie 字段存入 httpclient 的 http 代理(相当于一个浏览器);

(3)从种子 URL 开始爬取,从微博服务器中请求 URL 对应的 html 源码;

(4)交给 html 源码解释程序模块,调用 htmlparser 里面的功能函数和正则表达式进行 URL 和微博数据的提取与入库。

在这个过程中,要维护一个待爬队列和一个近期已爬取过的 URL 的已爬队列,新爬取的 URL 如果都不在这两个队列中,就把它存进待爬队列,已爬取过的 URL 存放进已爬队列,并且根据存放时间决定是否释放掉[7-8]。

1.2 微博模拟登陆

现在微博的无限制访问需要用户进行验证登录,有两种新浪微博服务器可供选择,一种是爬取 weibo. cn 服务器(以下简称 cn 版微博),另一种是爬取 weibo. com 服务器(以下简称 com 版微博)。cn 版微博的页面相对 com 版的页面更加简洁,网页源码更少,登陆账号密码都不加密,而且所需信息全面。电脑版模拟登陆需要对用户名和用户密码进行加密传输,而且 JavaScript 和广告图代码很多,会降低 HTML 源码分析效率和增加网络传输压力[9]。因此,主要采用手机版模拟登陆方式和抓取手机版微博页面。

1.2.1 com 版微博的模拟登陆

通过 Firefox 和 Firebug 分析了微博网页版登陆方式,其主要登陆步骤为:

(1)浏览器向微博服务器发送一个 Get 请求,服务器返回名为 servertime 数值和名为 nonce 的随机生成的字符串,每次登陆前随机生成传回给客户端加密用户密码;

(2)根据步骤 1 返回的字段值,使用 BASE64 算法加密账号名,用 RSA 算法对 servertime 加上 nonce 再加上用户密码这三个值组成的字符串进行加密,得到加密后的用户密码,向服务器发送包含加密过的用户名和用户密码的 Post 请求的登陆 URL[10];

(3)微博服务器收到登陆请求后进行登陆验证,如果验证成功就返回一个重定向的 URL 给客户端,浏览器解析该 URL 后,进入登陆成功页面并把该登陆标志写入本地 Cookies 中。

1.2.2 cn 版微博的模拟登陆

同样通过 Firefox 和 Firebug 来分析手机版的登陆方式,步骤为:

(1)打开手机版登陆 URL:login. weibo. cn,服务器返回一个带有名为“mobile”用于输入用户名输入框和名为“password_4 位随机数字”密码输入框的页面;

(2)向微博服务器登陆 URL 发送一个包含明文形

万方数据

式的用户名和密码的 Post 请求;

(3)微博服务器对收到的登陆请求进行验证,登陆成功后向客户端返回一个重定向 URL,并且 cookie 中包含 gsid_CTandWM 字段,浏览器解析该跳转 URL 进入登陆成功页面并把所有 cookie 字段写入本地 Cookies 中[11]。

以下是通过开发包 httpunit 获取手机版微博登陆 cookie 的 Java 代码实现:

```
import org. openqa. selenium. Cookie;
import org. openqa. selenium. WebElement;
import org. openqa. selenium. htmlunit. HtmlUnitDriver;
(1) HtmlUnitDriver hud = new HtmlUnitDriver( ); //打开 HtmlUnit 浏览器
(2) hud. setJavascriptEnabled(true); //默认执行 JS,如果不执行 JS,则可能登录失败,因为用户名密码框需要 JS 代码来绘制
(3) hud. get("http://login. weibo. cn/login/");//请求进入登陆页面
(4) WebElement mobile = hud. findElementByCssSelector("input[name=mobile]");//获取用户名输入框
(5) mobile. sendKeys(new CharSequence[ ] {username});//输入用户名
(6) WebElement pwd = hud. findElementByCssSelector("input[name^=password]");//获取密码输入框,名字以"password_"开头后面拼接 4 位随机数字
(7) pwd. sendKeys(new CharSequence[ ] {password});//输入密码
(8) submit. click( ); //点击按钮提交输入
(9) Set<Cookie> cookieSet = hud. manage( ). getCookies( ); return cookieSet; //登陆成功后得到的 cookie 集合
```

1.3　爬取手机版微博网页

1.3.1　浏览器中注入 cookie

在请求微博网页方面,在启动 httpclient 浏览器代理时,把 1.2 节通过 html 获取到的 cookieSet 注入进去。htmlunit 底层也是实现 httpclient 的,所以直接通过 httpclient 请求页面,方便而高效。

代码功能流程是,循环迭代出 cookieSet 集合中所有 cookie 键值对,并把它用 BasicClientCookie 对象包装后存进 CookieStore 对象中,最后在创建 httpclient 浏览器时把 CookieStore 对象添加进去。

```
import org. openqa. selenium. Cookie;
import org. apache. http. impl. client. BasicCookieStore;
Cookie cookie;
BasicClientCookie bcCookie;
CookieStore cookieStore = new BasicCookieStore( );
for (Iterator<Cookie> iterator = cookieSet. iterator( ); iterator. hasNext( ); ){
cookie = (Cookie) iterator. next( );
bcCookie = new BasicClientCookie(cookie. getName( ), cookie. getValue( ));
bcCookie. setVersion(0);
bcCookie. setDomain(". weibo. cn");
bcCookie. setPath("/");
cookieStore. addCookie(bcCookie);
}
CloseableHttpClient httpClient = HttpClients. custom( ). setDefaultCookieStore(cookieStore). build( );
```

1.3.2　获取微博页面源代码

要想进行数据提取,必须从微博服务器中获取到具体的网页源代码。通过 httpclient 获取微博 HTML 源代码的具体流程是:把要访问的 URL 传给要执行的 Get 请求(因为单纯访问简短 URL,用 Get 请求更合适);然后执行 Get 请求,服务器返回一个响应对象,通过该对象获取到具体的 HTML 源码 htmlSource。

```
import org. apache. http. client. methods. HttpGet;
import org. apache. http. HttpEntity;
import org. apache. http. client. methods. CloseableHttpResponse;
HttpGet httpGet. setURI(new URI(URL));
CloseableHttpResponse httpResponse = httpClient. execute(httpGet);
HttpEntity httpEntity = httpResponse. getEntity( );
String htmlSource = htmlSource = EntityUtils. toString(httpEntity);
```

1.4　主题关键词匹配

通过某一个主题的关键词来匹配相关微博,如果合适的就加入该主题的数据库表中,并返回该条微博包含的关键词字符串[12]。筛选出与蚊子叮咬传播疾病有关的微博,所设置的关键词有登革热、疟疾、虐原虫、乙脑、乙型脑炎、丝虫。

如果获取到的跟主题相关的微博数量太少,可以通过组拼 http://weibo. cn/search/mblog? keyword = '关键词'&page = n,来获取指定关键词按照发布时间排序的微博。

```
public string getKeywords(String weiboContent){
String[ ] keywords = {'登革热','疟疾','虐原虫','乙脑','乙型脑炎','丝虫病'};
String containsKeyword = new String("");
for(int i = 0;i<keywords. length;i++){
if(weiboContent. contains(keywords[i]))
containsKeyword+ = keywords[i]+";";
}
return containsKeyword;
```

万方数据

}

1.5 任务调度控制

1.5.1 URL 队列

在存放 URL 的队列中,必须要维护待爬队列和已爬队列。待爬队列是存放尚未爬取过的新 URL。已爬队列存放已经爬取过的 URL。已爬队列的作用主要是去重处理,当有新 URL 准备添加进待爬队列前,判断该 URL 是否存在于在其中一个,如果已存在,则把该新 URL 不加进待爬队列。初始化待爬队列,添加微博名人堂 URL 作为种子 URL 开始爬取[13]。

(1)待爬队列。

采用 Berkeley Database(BDB)构建待爬队列,因为从源码中提取的待爬 URL 数量太大,而且数据又简单,使用 URL 的 MD5 的值的十六进制字符串作为 BDB 的 key,而 value 则使用待爬 URL(CrawlURL. java)的对象。

URL 对象字段至少应包括:

private String oriURL; //原始 URL 的值,主机部分是域名

private int layer; //爬取的层次,从种子开始,依次为第 0 层,第 1 层…

待爬队列中提供的使用接口:

public CrawlURL getNext( );//获取下一条链接并把它从队列中移除

public boolean putURL(CrawlURL url);//往队列中添加一条 URL

(2)已爬队列。

在企业级搜索引擎中,常用一个称为 Bloom Filter(BF)的算法实现对已抓取过的 URL 进行去重处理。

BF 会根据大概要存储的数据量,建立一个 java. util. BitSet 对象(一个很长的二进制常量),并将所有位设置为 0。对每个 URL,用 k 个不同的随机数产生器($F_1,F_2,\cdots,F_k$)产生 k 个信息指纹($f_1,f_2,\cdots,f_k$)。再用一个随机数产生器 G(把信息指纹变成一个 0 到 BitSet 大小的随机数)把这 k 个信息指纹映射到 BitSet 对象中对应的 k 个自然数 $g_1,g_2,\cdots,g_k$。现在把这 k 个位置的二进制位全部置为 1。

1.5.2 爬取过程控制

(1)爬取频率控制。

新浪微博除了对微博 API 的调用次数进行了限制,同样对用户正常访问也进行了访问限制。实际操作中,如果微博爬虫访问过于频繁,微博账号或者网络 IP 会被进行封停操作,降低抓取频率,时间设置长一些,访问时间采用随机数,所以要控制好单个 IP 的访问频率。

(2)URL 权重控制。

万方数据

因为微博 URL 是动态更新的,对已经爬取过的 URL,一段时间后,页面的数据会发生变化,所以已爬取过的 URL 不能一直存放于已爬队列中,而且已爬队列是采用内存结构存储的布隆过滤器,如果不定期清理部分 URL,内存会产生溢出。根据计算机内存的大小控制好已爬队列 URL 的数量,一旦数量过多,应该清理等待释放时间最短的那部分 URL。根据 URL 类型,判断重要性,比如:粉丝量大、新闻类微博、明星等这些用户较为重要,对每个 URL 设置一个权值,权值越大,设置在已爬队列中的存放时间越短,并把它添加进一个特殊的 URL 队列中。因为这些 URL 中的微博是更多人阅读和关注的,成为热门微博的可能性更大,所以定期对其进行访问,不错失第一时间抓取这类微博的机会。爬取用户前多少页微博或者判断发布时间,如果发布时间过于久远,就停止爬取该用户的微博。同时爬取被关注的用户,用户的粉丝,也指定爬取前多少页。

2 cn 版微博数据提取

数据提取是从 html 源码中提取需要数据的环节。通过 Firebug 查看微博 html 源码,可以发现 html 源码中的结构和编写规则的特点,利用这些特点进行数据提取与筛选。主要功能有:微博用户资料信息的提取、微博信息的提取、新 URL 的获取、关系用户的获取。

把 1.3.2 节获取到的 html 源码 htmlSource 传入 htmlparser,利用过滤节点和获取节点文本的功能,结合正则表达式的字符串匹配功能,实现数据的提取操作。

```
import org. htmlparser. * ;
Parser parser = new Parser(htmlSource);
NodeFilter titleFilter = new NodeClassFilter(TitleTag. class);
NodeFilter divFilter = new NodeClassFilter(Div. class);
NodeFilter[ ] filters = {titleFilter, divFilter};
NodeFilter orFilter = new OrFilter(filters);
NodeList nodeList = parser. extractAllNodesThatMatch(orFilter);
```

2.1 微博用户资料的信息提取

微博用户的资料信息,比如:已发布微博数量、关注他人数量、粉丝数量,都出现在第一个 节点的文本中,用户的 ID(uid)出现在该 div 节点下的节点中;微博网页的标题是由“用户昵称+的微博”组成,所以就可以从网页标题中截取用户昵称。

2.2 微博信息的提取

每条微博的信息,比如:微博内容、发布时间、赞次数、转发次数、评论次数、发布来源、发表日期、微博配

图,都包装在 节点的文本中,微博 ID(mid)存在于该节点的属性值中;微博图地址存在于该节点里面的<a>中,可用正则表达式:matches(". * \\s+src=\" http://. +\\s+alt=\"图片\"\\s+. * class=\" ib\". * ")进行匹配获取。

观察微博发布日期的形式有:多少分钟前、今天 时:分、几月几日 时:分、年-月-日 时:分:秒,把所有发布日期统一转换成"年-月-日 时:分:秒"这种 java. sql. Timestamp 格式进行存储。

2.3　新用户获取与 URL 组拼

提取并过滤出有用的 URL,网页源码中的 URL 存在于<a>标签中的 src 属性中,利用 htmlparser 进行 URL 提取。但是获取结果中存在大量的无价值 URL,如果不进行过滤筛选,会大大降低爬虫的抓取效率。需要过滤掉的 URL 类型有:非 weibo. cn 内网、点赞、举报、加关注、@他(她)的、相册、查看原图、赞、收藏、举报、回复,微博页面底端定制模块的皮肤、图片、条数、隐私、私信、特别关注、资料,微博分类选项的原创-图片-分组-筛选、查看原图等等,这些 URL 链接都是有特点的格式,很容易过滤掉。因为新浪微博的页面具有一定的格式,而且只爬取新浪微博站点的信息,所以可通过 URL 组拼的方式更有效地获取大量信息[14]。

通过对关注用户和粉丝用户的获取,可以把所有用户组建成若干个网络关系社团。想要获取某个用户关注的人,可以组拼 URL 为:http://weibo. cn/uid/follow? page=n,想要获取粉丝,可组拼 URL 为:http://weibo. cn/uid/fans? page=n。其中 uid 是该用户的 id,n 是结果列表的页号。然后可以从这些结果页面中提取出大量的用户 ID 和昵称。

3　实验及结果分析

3.1　采集性能比较与分析

当网络足够好时,将 cn 版微博爬虫与文献[10]中的新浪微博 API、传统爬虫、网页版微博爬虫的爬取效率进行比较,如图 2 所示。

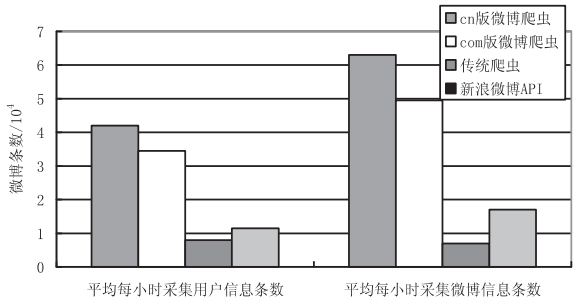

图 2　不同方案采集用户信息和采集微博的速率比较

可以看到,cn 版微博爬虫在效率上高于其他类型的微博采集方式[15]。

3.2　蚊子叮咬传播疾病的数据分析

根据匹配关键词登革热、疟疾、虐原虫、乙脑、乙型脑炎、丝虫提取出的相关微博,然后根据发布时间筛选出属于 2011 年 1 月到 2015 年 11 月的微博总数共 850 794条,具体分布如表 1 所示。

表 1　文字叮咬疾病相关微博分布比例

包含关键词	相关微博数/条数	占总微博数比例/%
登革热	398 010	46.781
疟疾或疟原虫	294 341	34.596
乙脑或乙型脑炎	138 067	16.228
丝虫病	20 376	2.395
总计	850 794	100

经过统计,包含关键词登革热的微博有 395 976 条,包含疟疾或疟原虫的有 292 843 条,占比例 46.781%;包含乙脑或乙型脑炎的有 141 691 条,丝虫病的微博有 20 274 条,可知跟蚊子有关的微博主要是登革热和疟疾。

将所有微博按照时间段从 2011 年 1 月到 2015 年 11 月,然后按微博发布的月份进行分类统计,可以得到每个月相关微博数量和相关参与用户的数量,如图 3 和图 4 所示。

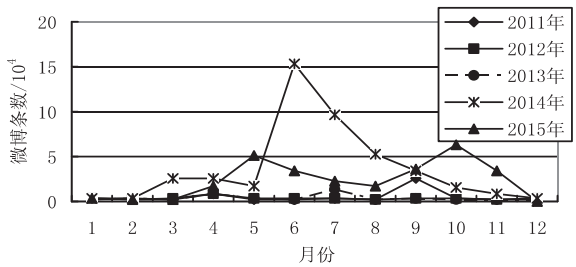

图 3　与蚊子叮咬传播疾病微博数量随时间的变化折线图

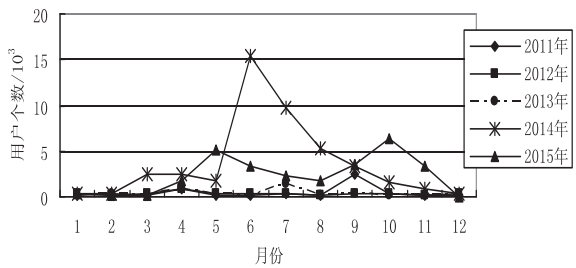

图 4　相关微博的用户参与数量

图 4 统计出了每个月份发布微博对应用户的数量,该用户数量是经过重复性去除后产生的,即找到该月每条微博对应的发布者后,判断该用户是否被统计过,如果已经被统计过,则不再统计。例如,在某个月的多条微博属于同一个用户发布,但是该用户只统计一次。

从图 3 和图 4 可以看出,2014 年 6-8 月与 2015 年

万方数据

5、9、10 月分别出现了一个最大的和较大的波峰,表明这一期间用户对蚊子叮咬产生的相关疾病较为关注。而现实中的情况是,2014 年 6 月,印度东北部和广州同时爆发严重的感染登革热病例的疫情;2015 年 5 月,台湾和其他国家一些地区爆发较为严重的登革热疫情;同年 9 月印度爆发 5 年来最严重的登革热疫情;同年 10 月中国科学家屠呦呦获得诺贝尔医学奖,她发现的青蒿素能有效治疗疟疾,同样引起了很多用户关注。相关微博发布大波峰基本都是爆发登革热疫情,从而引起用户的参与关注,随着时间的推移,用户对其的关注度逐渐降低,发布的相关微博越来越少。

4 结束语

从数据量庞大的微博数据中采集想要的数据,拥有一个效率高、可灵活控制采集方向和主题导向的微博爬虫至关重要。为此,采用 Java 结合流行的 HTML 源码解析工具开发包实现了一个可以灵活控制爬取条件的微博网络爬虫,其采用基于手机版模拟登陆与页面解析的方法,实现起来更加便捷。而 Java 实现爬虫的代码,对相关开发人员来说有更大的实用价值,节省了大量爬虫代码设计时间。利用该爬虫系统,能更高效地对大规模微博数据,特别是最新数据,进行采集,用户也可以结合微博 API 来进行数据采集,扩大微博数据的采集量,从而为微博信息的数据挖掘提供更全面、准确的数据支持。实验结果表明,该系统有效可行,采集效率明显优于其他几种采集方式,且相对于微博 API 更灵活多变,在广度遍历 URL 方面更出色。

参考文献:

[1] Wen E,Sun V. 新浪微博研究报告[R/OL]. 2011-05-20. http://www.techWeb.com.cn/data /2011-02-25/916941.shtml.

[2] 高 凯,王九硕,马红霞,等. 微博信息采集及群体行为分析[J]. 小型微型计算机系统,2013,34(10):2413-2416.

[3] 纪 伟. 微博数据采集系统的设计与实现[D]. 石家庄:河北科技大学,2013.

[4] Han Ruixia. The influence of microblogging on personal public participation[C]//Proceedings of the 2010 IEEE 2nd symposium on web society. Beijing,China:IEEE,2010:615-618.

[5] 周德懋,李舟军. 高性能网络爬虫:研究综述[J]. 计算机科学,2009,36(8):26-29.

[6] 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.

[7] Du Y,Zhang K,Lyu X,et al. The study of gathering and extracting users information based on micro-blog[C]//International conference on computer,mechatronics,control and electronic engineering. [s. l.]:[s. n.],2010:47-50.

[8] 李超锋,卢炎生. 基于 URL 结构和访问时间的 Web 页面访问相似性度量[J]. 计算机科学,2007,34(4):207-209.

[9] 尹 江,尹治本,黄 洪. 网络爬虫效率瓶颈的分析与解决方案[J]. 计算机应用,2008,28(5):1114-1116.

[10] 孙青云,王俊峰,赵宗渠,等. 一种基于模拟登录的微博数据采集方案[J]. 计算机技术与发展,2014,24(3):6-10.

[11] Lu G,Liu S,Lü K. MBCrawler:a software architecture for micro-blog crawler[C]//Proceedings of the 2012 international conference on information technology and software engineering. Berlin:Springer,2013:119-127.

[12] 刘金红,陆余良. 主题网络爬虫研究综述[J]. 计算机应用研究,2007,24(10):26-29.

[13] 周中华,张惠然,谢 江. 基于 Python 的新浪微博数据爬虫[J]. 计算机应用,2014,34(11):3131-3134.

[14] 罗一纾. 微博爬虫的相关技术研究[D]. 哈尔滨:哈尔滨工业大学,2013.

[15] 朱云鹏,冯 枫,陈江宁. 多策略融合的中文微博数据采集方法[J]. 计算机工程与设计,2013,34(11):3835-3839.

(上接第 190 页)

[8] 王丽侠,楼玉萍,吕君可. 基于 Web 服务的电力信息集成系统[J]. 计算机技术与发展,2009,19(5):173-175.

[9] 麦瑞坤,何正友,符 玲,等. 基于电流行波能量和小波变换的输电线路故障选相研究[J]. 电网技术,2007,31(3):38-43.

[10] Ma Lan,Bi Dongjie,Wang Houjun. Analog circuit fault detection using relative amplitude and relative phase analysis[J]. Journal for Control,Measurement,Electronics,Computing and Communications,2015,55(3):343-350.

[11] 陈 旸,胡 炎,邰能灵,等. 基于电压故障分量的超高压线路故障选相新方法[J]. 电力系统保护与控制,2014,42(7):8-14.

[12] Chandel A K,Patel R K. Bearing fault classification based on wavelet transform and artificial neural network[J]. IETE Journal of Research,2013,59(3):219-225.

[13] 段建东,张保会,周 艺,等. 基于暂态量的超高压输电线路故障选相[J]. 中国电机工程学报,2006,26(3):1-6.

[14] 李 勋,龚庆武,贾晶晶. 采用形态小波变换原理的超高速故障选相算法研究[J]. 电力系统保护与控制,2011,39(15):57-63.

[15] 林英建. 数据库逻辑设计性能优化关键技术研究[J]. 计算机技术与发展,2013,23(12):74-77.

[16] 刘 波,范士明,刘 华. 一种面向实时数据库存储引擎的设计与实现[J]. 计算机技术与发展,2011,21(8):34-38.

[17] 马建峰,张广泉. 基于 C/S 架构订单生产管理系统设计与开发[J]. 计算机技术与发展,2008,18(6):174-178.

[18] 周 来,孟祥萍,张本法,等. 智能电网通信与信息管理系统核心问题研究[J]. 计算机技术与发展,2015,25(4):144-147.

方方数据