DOI [illegible] j cnki qhdxxb [illegible]

ISSN 1000-0054 清华大学学报(自然科学版) 2011 年 第 51 卷 第 10 期 13/25
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2011, Vol.51, No.10 1300-1305

新浪微博数据挖掘方案

廉 捷[1], 周 欣[2], 曹 伟[2], 刘 云[1]

(1. 北京交通大学 通信与信息系统北京市重点实验室, 北京 100044; 2. 中国信息安全测评中心, 北京 100085)

摘 要: 随着新浪微博用户群体的增长, 新浪微博的数据获取是微博研究首先需要解决的问题。该文提出了基于新浪微博 API 与基于页面解析的新浪微博数据获取方案。程序逻辑控制 API 调用方法与频率, 获取 JSON 对象并解析实现高效数据获取。同时将传统的网络爬虫结合网页解析技术结合 API 同时使用, 解决了因 API 接口开放不完善, 且因在返回结果数量上限与调用频率方面的限制, 导致不能有效实现新浪微博数据的全面获取的问题。经过实验测试, 通过 2 套方案的结合可以实现新浪微博数据高效全面的获取。

关键词: 新浪微博; 新浪 API; 数据检索; 网页解析

中图分类号: TP 391 文献标志码: A

文章编号: 1000-0054(2011)10-1300-06

SINA microblog data retrieval

LIAN Jie[1], ZHOU Xin[2], CAO Wei[2], LIU Yun[1]

(1. Key Laboratory of Communication & Information Systems of Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China; 2. China Information Technology Security Evaluation Center, Beijing 100085, China)

Abstract: With the large increase of SINA Microblog web users, analyses SINA Microblog data has aroused wide public concern. The data retrieval is a primary need for all these studies. This paper presents two data retrieval techniques based on opened APIs and web crawlers with webpage extraction. A program controls the methods and frequencies of the API queries for efficient data downloading via parsing of JSON objects. A traditional web crawler integrated with webpage extraction overcomes the shortcomings of opened SINA API interfaces that limit retrieval of the entire dataset without quantity and frequency limitations. Test results show that a combination of these two techniques allows complete data retrieval of SINA Microblog data.

Key words: SINA microblog; SINA API; data retrieval; webpage extraction

微博作为 Web 2.0 时代新生网络应用形式, 在最近几年中得到了迅猛的发展。新浪微博中一条用户状态限定 280 字符的内容长度, 不但更适合现代社会快速生活节奏的需要, 而且也更方便用户通过移动通信终端上传和分享自己感兴趣的微博信息[1]。在美国, 微博网站 Twitter 自 2006 年创建以来[2], 用户数量在近几年中突飞猛进, 其中 2009 年 Twitter 的用户增长率达到 2 565%, 是社交网站 Facebook 与 LinkedIn 增长率总和的 10 倍[3]。在中国, 已有 14%的互联网用户开始使用微博, 而新浪微博的市场份额占有率接近 87%, 是中国微博产业的主导力量[4]。

微博的使用人群数量基数大, 状态信息更新频繁、信息传播迅速。并且微博平台媒介用户占有率相对集中, 因此基于微博数据的分析研究成为了十分值得关注的研究方向。HAN Ruixia[5] 介绍了微博平台的特点与基本概念, KANG Shulong[6] 针对新浪微博研究了其群体结构与度分布特征。WANG Rui[7] 通过数据分析解释了用户好友数量与用户状态受关注程度之间的关系, 但其中并没有介绍数据来源, 而新浪微博 API 仍处于测试阶段, 不但内容开放不全面, 而且查询结果返回最大数量与调用频率方面也存在诸多限制, 难以实现全面的数据获取。周立柱等[8-10] 提出了一套依赖网络爬虫抓取网页内容并根据一定的规则提取页面中的有用信息的思路, 但因为没有涉及登录模式, 因此也难以运用在新浪微博的数据获取过程中。因此, 如何实现新浪微博数据的全面高效获取, 成为了研究新浪微博的首要问题。本文提出利用新浪微博 API 与传统网页解析方案, 以实现新浪微博数据的高效抓取。

收稿日期: 2011-08-15

基金项目: 高等学校博士学科点专项科研基金资助项目 (20100009110002); 北京市自然科学基金资助项目 (4112045)

作者简介: 廉捷(1985—), 男(汉), 北京, 博士研究生。

通信作者: 刘云, 教授, E-mail: 08111004@bjtu.edu.cn

1　基于 API 的数据获取

互联网络数据的获取通常是通过网络爬虫实现的。在一段网络爬虫程序中，通过设定入口 URL 地址，程序按照一定的爬行策略将网页内容以文本文件的形式保存在本地存储系统中，并提取网页中有效地址作为下一次爬行的入口 URL 地址，直到爬行完毕或者满足既定爬行条件后程序终止。相比网络爬虫，新浪微博的开放 API 接口可以更加简洁的获取相应的数据，为程序高效获取微博数据提供了保障。

1.1　OAUTH 认证

使用新浪 API 首先要解决的是用户认证问题。所谓认证是指用户在不向第三方透露自己的用户名密码的同时，使第三方软件提供方申请获得该用户资源的授权。OAUTH 认证为用户资源的授权提供了一个安全的、开放而又简易的标准，被用于新浪微博 API 的用户验证协议，过程如下：

1) 用户向新浪微博 OAUTH 服务提供商申请应用，获得应用专属 App Key 和 App Secret，分别以 HMAC-SHA1 算法对用户发出的请求进行数字签名。

2) 通过新浪微博开放平台获取 Request Token。未经服务器授权的 Request Token 中包含了对应密钥、加密算法、发起请求的时间戳、随机字符串与版本号信息。

3) 向新浪微博服务器 Request Token 授权地址发送请求，服务器同意用户的请求，并向其颁发未经用户授权的 Oauth Token 与对应的 Oauth Secret。

4) 将上一步得到的 Token 与 Secret 发给新浪微博用户授权地址申请 Request Token 授权。

5) 授权后，用户再向新浪微博 Access Toke 地址发起请求，将上步授权的 Request Token 换取成 Access Token。

6) 服务器同意用户请求，并向其颁发通过新浪微博授权的 Access Token 与对应的密钥。

7) 用户将获得的授权的 Access Token，以基于 http header 的 OAUTH 形式对请求进行签名，就可以获得软件对于用户身份资源的使用授权。

1.2　新浪微博 API

OAUTH 授权解决了程序访问 API 的用户身份认证问题。新浪 API 可根据请求内容的不同，返回特定的 XML 或 JSON 文件。XML 作为一种跨平台的强结构性扩展标记语言，因为所有的用信息都被对应的标签所标记，例如：〈id〉1861021910〈/id〉〈name〉Hugo〈/name〉，所以用户可以便捷地找出相应信息并理解其中内容。本文的程序由 JAVA 语言开发实现，利用 dom4j. jar 可以轻松实现 XML 文件的读取。但是因为 dom4j 对于 XML 的结构规范要求十分严格，而微博中用户状态与信息可能包含一些用户自身偏好的个性化字符格式，这些字符导致程序将无法正确解析整个 XML 文档，因此采取 JSON 返回方式具有更高的稳定性。

JSON 是一种轻量级的数据交换格式，文件不具有明显的强结构特征，如：{“id”：“1861021910”，“name”：“Hugo”}。在复杂的 JSON 对象中，因为它不像 XML 文件用规范的标签形式标记有效的内容，虽然对于人来说较难整理，但因为 JSON 文件结构简单，因此通过电脑分析 JSON 文件具有强大的处理能力。另外 JSON 文件中因为不再具有用于标记内容属性的说明性标签，所以 JSON 文件相比 XML 文件，查询相同内容的返回文件更小，因此更适合作为微博海量数据获取中的文件传输形式。

在新浪微博 API 接口定义中，如：http：//api. t. sina. com. cn/statuses/followers. json？source＝appkey&user_id＝11051&count＝100&cursor＝1600。这个地址中，http：//api. t. sina. com. cn 为新浪 API 存放服务器地址，statuses/followers 为调用请求的具体方法，json？表示返回的是 JSON 文件形式，问号之后为传递参数。appkey 是上述 OAUTH 认证中用户第一步申请的专属 App Key 字符串；user_id 是所查询用户的数字 ID；count＝100 表示每一次查询最多返回 100 条记录，在新浪 API 中这个数值最大为 200，缺省值为 20；cursor＝1600 表示这一次查询从第 1 600 条记录开始返回之后的 100 条数据，根据新浪微博 API 调用限制，每次查询最大只能返回 5 000 条数据。

为了均衡服务器的负载，通常 API 服务商对用户的 API 接口调用频率与查询范围也会根据用户权限的不同有所限制。除了上述一次请求最多只能返回 5 000 个结果之外，新浪微博 API 对用户调用接口的查询次数也有限制。普通授权用户每小时接口使用上限 1 000 次，并且在程序运行中如果短时间内较多地调用 API 接口，虽然总次数并没有达到每小时的 1 000 次，但因为在这段时间内调用高度频繁，程序调用结果依然会返回 403 错误，即用户已达到 API 使用上限。为了避免用户过高访问新浪 API 接口，程序必须通过线程控制 API 访问频率，程序流程如图 1 所示。

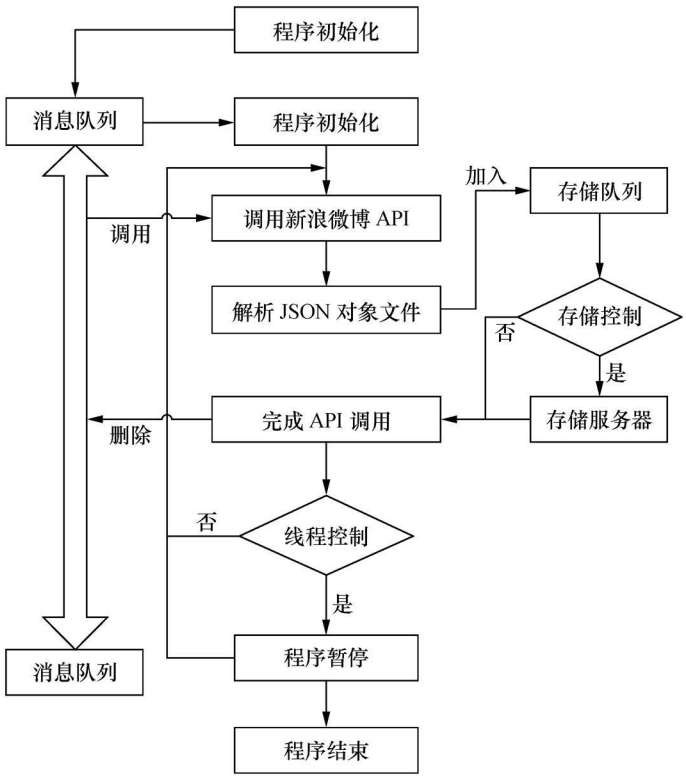


图 1 新浪微博 **API** 数据抓取程序流程图

因为新浪微博的数据量是十分庞大的,程序逻辑不但需要控制从什么地方开始抓取、在什么时候结束,还需要了解程序已经处理过了哪些数据、还要处理哪些数据。数据抓取程序是一段需要长时间运行的网络程序,其中可能会因为各种意外而中止。如果程序意外终止时,没有记录程序到达的位置,那么每一条可能引起重复记录的数据都要进行判断,这样则大大降低了程序的运行效率。因此程序初始化将待处理的信息放入队列,每成功处理一条信息再从队列中删除,以保证所有数据不重复不遗漏得以高效处理。程序的线程用于控制 API 的查询次数。通常每调用 50 至 100 次请求后,程序将暂停数分钟以保证新浪 API 对于用户的请求限制。数据存储与 API 查询是 2 个独立的过程。因为每次 API 查询请求可能返回 0 至 5 000 条消息记录,对于一次查询返回的信息,如果信息的条目太多则会占用较大的系统开销,若太小则会浪费一次 JDBC 建立的存储 session。所以存储控制用来均衡程序每一次向数据库提交信息的数量,保证 JDBC 的高效存贮。

1.3 新浪微博 **API** 与 **SDK** 的比较

新浪 SDK 是为方便微博 API 调用而开发的一套软件开发包,其中封装了从授权认证到数据获取与解析的各项功能。SDK 是建立在 API 基础上的集成与开发,基于 SDK 的工程可以大大降低程序开发的工作量。目前已开发并发布的 SDK 包括支持 JAVA、C++、PHP、PYTHON 等 12 种不同计算机语言的版本。但新浪微博作为一种新生网络应用,SDK 开发并不完善,功能性与稳定性均不如 API 表现出色。另外,因为主要微博服务提供商,如美国的 Twitter、中国的新浪等,授权验证与 API 调用形式是相同的,唯一不同的只是 API 请求发送地址的区别,因此通过 API 自行编写的程序,不但可以灵活实现新浪微博开放 API 接口的所有功能,同时只需要修改 API 调用地址与参数,就可以实现其他微博平台数据的抓取与处理。

2 基于网络爬虫的页面解析

通过调用 API 接口可以实现新浪微博数据的便捷抓取与解析,但所有微博服务商都不会无条件将完整 API 开放给普通用户,因此使用 API 的方式永远只可以解决微博数据获取中的一部分问题。如在新浪微博中,很多重要查询功能的 API 是不开放的,同时对于开放的 API,一条查询的返回结果数目上限为 5 000,而往往那些拥有较大信息量的节点才是微博研究最关心的问题。于是在 API 之外,还需要引入网络爬虫与网页解析技术,来获取更多

的信息。

2.1　新浪微博模拟登录过程

用户通过浏览器访问新浪微博是需要登录的。通常浏览器可以自动获取保存在用户电脑中的cookie并创建session，实现用户登录信息在微博页面中的传递。与API的使用一样，新浪微博服务器无法对一段计算机程序的访问创建对应session，因此，实现爬虫程序对于指定微博页面的正确抓取，必须首先解决程序对于网页的模拟登录问题。

新浪微博的模拟登录过程，是向服务器端以一定的格式发送经过Base64编码加密的原始用户名与密码信息，服务器从http header里包含的授权信息Authorization中提取字符串并解密后得到原有用户名与密码，实现程序对网页的模拟登录过程。其中Base64加密算法是模拟登录过程的关键问题，Base64加密过程如下：

1) 将原始字符串以utf 8编码格式转换为原始二进制字符，其中8b代表一个字节。

2) 将原始二进制字符分组，每3字节一组，如果剩余字符不足3字节，则空余位数补“0”。

3) 将每组3字节24位编码分为4段，每段前+“00”组成新的4字节32位编码。

4) 将每组32位编码，以每8b分段，将每段8bit字符转化为十进制数字。通过Base64编码表将对应数字转化成相应字符，实现Base64加密过程。加密实例如图2所示，Base64编码见[11]。模拟登录的描述过程如下：

```
code= username+“：”+password
auth=“Basic”+Base64.encode(code,”utf 8”)
OpenConnection(url)
connection.setMethord(“GET”)
connection.setProperty(“Authorization”, auth)
InputStream= connection.getInputStrem()
```

将加密后的用户名密码，在每一次请求连接时作为Authorization属性发送出去，用InputStream获取请求返回的内容，从而实现了整个登录流程。

原始字符串	log			
原始二进制	01101100	01101111	01100111	
转换后二进制	00011011	00000110	00111101	00100111
转换后十进制	27	6	61	39
加密结果	b	G	9	n

图2　**Base64**加密算法实例

2.2　页面存储与数据处理

通过上节中InputStream输入流，可以取出指定URL地址中的网页内容信息，并将其以文本文件形式按一定的分类标准保存在本地存储系统中。在新浪微博中相同种类页面的编码规则是一样的，也就是说对于同种性质的页面，只需要制定一套页面解析规则，就可以处理属于这个分类的所有网页文件。

因为网页HTML文件是一种不严谨的网页标记语言，这会对网页信息的提取造成一定困难，因此首先需要将不标准的HTML语言转化为标准的DOM树结构，然后利用XPath可以定位存放关键信息的DOM节点位置，最后抽取XPath特征节点中的内容。在新浪微博用户搜索页面中，生成DOM树后所有有效信息均保存在属性(class)为srch_main_conlf的一对〈div〉节点中，如用户名ID信息uid＝“1726602xxx”的相对地址为class＝“concernBox”的〈ul〉标签中第一个〈li〉标签中，某〈a〉标签内的第3个属性。那么寻找这个uid的相对路径为：//DIV[@class＝‘srch_main_conlf’]/UL[@class＝‘concernBox’]/LI[1]/DIV[1]A/IMG[Attributes(3)]。于是当找到了〈img〉标签中的第三个属性uid后，则取得了uid的值。程序逻辑遍历这个DOM树中的所有LI[@class＝‘MIB_linedot_l’]的节点，便可以得到〈li〉节点中所有用户信息。

3　数据分析

测试机器为DELL R710服务器，Intel Xeon E5506双核CPU，2 GB内存，运行环境为WindowsXP平台下Eclipse3.2，程序由JAVA开发实现，接入网络为100 MB共享教育网专线。

3.1　查询结果文件大小比较

本文提出了基于新浪微博API与基于网页解析技术的2套新浪微博数据获取方案，其中基于API的数据获取根据请求内容的不同可选择性返回XML和JSON文件。测试中首先人工选择100位好友不超过5 000人的新浪微博用户，以保证用户的所有数据均可以通过API和网络爬虫正确获取。

表1为这100个用户的好友查询返回文件大小对比。在测试中，XML与JSON是基于API的查询结果，因为基于API与网络爬虫的两种数据获取方式无法做到时间上的完全同步，因此程序结束时间点的差异导致程序返回的好友总数已在网络中产生了变化。另外，通过API查询用户好

友信息相比网络爬虫返回结果,包括了更多的用户信息以及用户最新更新的一条状态。因此可以看出相比网络爬虫方式获取数据,新浪微博 API 返回文件大小更小,且内容更丰富。而相对于同样基于 API 查询的 XML 返回方式,JSON 具有更高的返回效率。

表 **1** 用户好友查询文件大小对比表

请求方式	好友总数	文件总数	文件大小/MB
XML	151 472	810	329.87
JSON	151 472	810	211.53
网络爬虫	151 618	9478	655.31

表 2 为用户状态查询结果比较。测试分别通过 API 与网络爬虫方式获取用户最新发布的 100 条微博信息。通过返回文件大小对比可以发现,因为网络爬虫所保留的原始 HTML 文本文件中包含大量转义地址信息、JavaScript 标签等内容,因此通过网页解析用户状态效率远小于通过调用 API 方式得到的用户状态信息。而在基于 API 的查询中,JSON 返回方式相比 XML 同样具有更高的返回效率。

表 **2** 用户状态查询文件大小对比表

请求方式	状态总数	文件总数	文件大小/MB
XML	9477	100	37.78
JSON	9477	100	28.26
网络爬虫	9821	200	234.28

3.2 程序处理性能比较

测试中发现,无论通过调用 API 方式还是网络爬虫方式,在原始数据采集方面的时间损耗主要取决于数据采集总量大小以及当时的网络状况与新浪微博服务器负载。而对于数据下载量相同的一段任务,在相似的时间,API 调用方式与网络爬虫方式可以近似看成是无差别的,因此这里只考察程序对下载后数据的处理能力。程序逻辑分别将通过调用 API 方式与网页爬虫方式得到的包含 534 618 个用户信息的原始 XML、JSON 与 HTML 文本文件进行处理,将处理结果保存在 MySQL 数据库中,每存储 10 000 条信息记录一次处理时间,其中 XML 方式处理时间为 248 299 ms, JSON 方式处理耗时为 209 372 ms, 而网页方式处理耗时为 552 876 ms, 具体结果如图 3 所示。

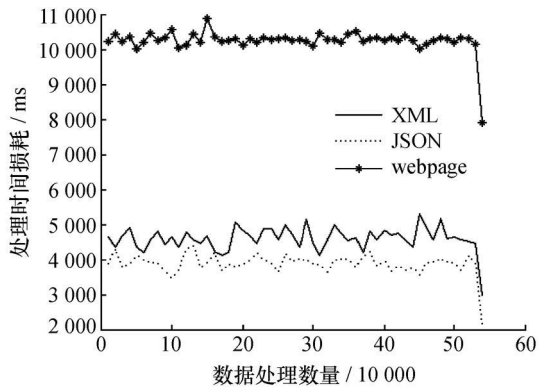


图 **3** 数据处理性能对比

从图 3 可以看出,基于新浪 API 的 XML 与 JSON 处理方式具有更高的程序处理能力,其中 JSON 处理性能最高。而基于网络爬虫的网页解析方式不但处理速度较慢,而且处理数量相同的有效信息需要消耗更大的原始信息文档,因此基于新浪微博 API 的数据抓取相比基于网络爬虫的网页文件解析都更具优势。

3.3 基于 API 与网页解析方案的结合

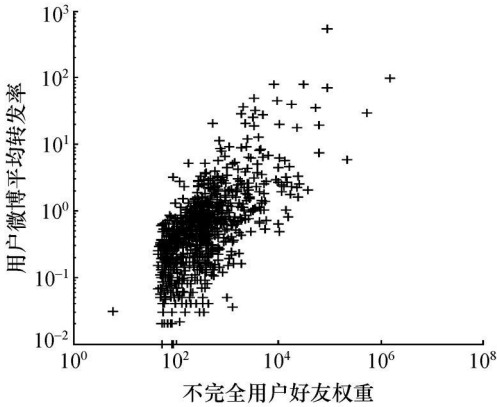
基于新浪 API 的数据抓取策略性能高,但因为服务器限制所以不能获得完整数据集,基于网页解析的数据获取方案可以获得最大的数据文本但效率低下。通常需要将两者结合起来,以实现最佳数据抓取效果。如在研究新浪微博用户受关注程度的问题上,首先通过新浪 API 收集了 857 个用户信息与这些用户的共 53 万条好友信息。因为 API 调用返回上限限制,程序选择通过基于网络爬虫的页面解析方案抓取这 857 个用户的所有微博状态信息,并计算出每一条用户状态的平均转发率和回复率。

$$W_{re} = F + r_{re} \cdot \left\{ \sum_{i \in n} F_i \right\} \cdots\cdots, \quad (1)$$

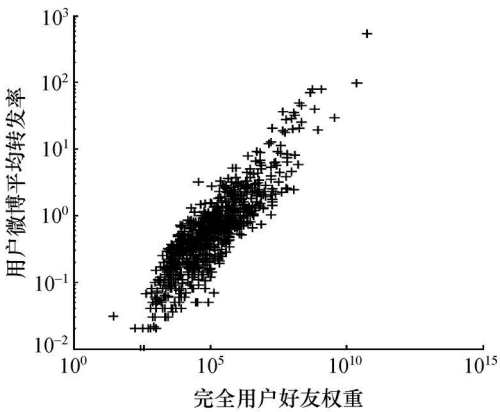
$$W_{rt} = F + r_{rt} \cdot \left\{ \sum_{i \in n} F_i \right\} \cdots\cdots, \quad (2)$$

其中:F 表示当前用户的好友数量;r_{re} 与 r_{rt} 分别代表当前用户状态的平均回复与转发率;F_i 表示当前用户第 i 个好友的好友数量,因此 W_{re} 与 W_{rt} 为定义的用户评论好友权重与用户转发好友权重。

图 4 代表了用户好友权重对于用户微博平均回转发次数的影响,图 4a 横坐标为数据不完全统计时用户好友数量,反应了用户好友数对于用户状态平均转发关系的影响;图 4b 中横坐标为 W_{rt}, 反应了数据完全统计时用户好友权重对于用户状态平均回复与转发数的影响。



(a) 用户好友数量与微博平均转发率的关系



(b) 用户好友权重与微博平均转发率的关系

图 4　用户好友权重对终于状态平均转发数的影响

从图 4 可以看出，结合了 2 套方案获取的数据信息可以更好地描述用户好友权重与用户状态受关注程度之间的线性关系，奠定了新浪微博用户分析研究的基础，因此 2 套方案的结合可以为今后新浪微博的研究提供全面完整的数据保障。

4　结　论

本文提出了基于新浪微博开放 API 与基于网络爬虫的页面解析 2 套新浪微博数据挖掘方案。在新浪微博 API 调用的返回数据形式中，JSON 相比 XML 方式返回效率更高、处理速度更快，对于特殊字符的解析能力也更强，是程序使用 API 调用方式的首选。与基于 API 的数据获取相比，基于网络爬虫的页面解析方案效率与性能相比 API 都有明显差距，还要解决额外的程序模拟登录问题。但页面解析技术可以为今后的数据分析获取更加完整的数据集。通过 2 套方案的结合可以实现新浪微博数据全面高效的抓取和解析，从而为针对新浪微博的网络结构分析、用户行为与群体特征分析、网络话题发现跟踪与预测等研究提供了完善的数据保障。

参考文献　(References)

[1] Westman S, Freund L. Information interaction in 140 characters or less: Genres on twitter [C]//IIiX 2010—Proceedings of the 2010 Information Interaction in Context Symposium. New Brunswick, USA: Association for Computing Machinery, 2010: 323 – 326.

[2] Pieter N, Michiel H. Mining Twitter in the cloud: A case study [C] // Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010. Miami, USA: IEEE Computer Society, 2010: 107 – 114.

[3] Abraham R, Martínez T. Twitter: Network properties analysis [C] // Proceedings of the CONIELECOMP 2010—20th International Conference on Electronics Communications and Computers. Cholula Puebla, Mexico: IEEE Computer Society, 2010: 180 – 184.

[4] Wen E, Sun V. 新浪微博研究报告 [Z/OL]. (2011-05-20), http://www.techweb.com.cn/data/2011-02-25/916941.shtml.

[5] HAN Ruixia. The influence of microblogging on personal public participation [C] // Proceedings of the 2010 IEEE 2nd Symposium on Web Society, SWS 2010. Beijing, China: Association for Computing Machinery, 2010: 615 – 618.

[6] KANG Shulong, ZHANG Chuang. Complexity research of massively microblogging based on human behaviors [C]//2010 2nd International Workshop on Database Technology and Applications, DBTA2010—Proceedings. Wuhan, China: IEEE Computer Society, 2010: 1 – 4.

[7] WANG Rui, JIN Yongsheng. An empirical study on the relationship between the followers' number and influence of microblogging [C] // Proceedings of the International Conference on E-Business and E-Government, ICEE 2010. Guangzhou, China: IEEE Computer Society, 2010: 2014 – 2017.

[8] 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.
ZHOU Lizhu, LIN Ling. Survey on the research of focused crawling technique [J]. *Journal of Computer Applications*, 2005, **25**(9): 1965 – 1969. (in Chinese)

[9] 张彦超，刘云. 基于自动生成模板的 Web 信息抽取技术研究 [J]. 北京交通大学学报，2009, **33**(5): 40 – 45.
ZHANG Yanchao, LIU Yun. Study of web information extraction technology based on automatically generated template [J]. *Journal of Beijing Jiaotong University*, 2009, **33**(5): 40 – 45. (in Chinese)

[10] 欧健文，董守斌. 模板化网页主题信息的提取方法 [J]. 清华大学学报：自然科学版，2005, **45**(09): 1743 – 1747.
OU Jianwen, DONG Shoubin. Topic information extraction from template web pages [J]. *Journal of Tsinghua University: Science and Technology*, 2005, **45**(09): 1743 – 1747. (in Chinese)

[11] 姚峰. Java 平台中 Base64 编码 / 解码算法的改进 [J]. 计算机应用与软件，2008, **25**(18): 164 – 165, 176.
YAO Feng. Improvement of BASE64 encoding/decoding algorithm in JAVA [J]. *Computer Applications and Software*, 2008, **25**(18): 164 – 165, 176. (in Chinese)