

Chapter 9

Applied Methods and Techniques for Modeling and Control on Micro-Blog Data Crawler

Kai Gao, Er-Liang Zhou and Steven Grover

Abstract Models can provide mechanisms to improve system performance. This chapter presents the applied methods and techniques for modeling and controlling on micro-blog crawler. With the rapid development of social studies and social network, millions of people present or comment or share their opinions on the platform everyday, and as a result, produce or spread their opinions and sentiments on different topics. The microblog has been an effective platform to know or mine social opinions. In order to do so, crawling the relevant microblog data is necessary. But it is hard for a traditional web crawler to crawl micro-blog data as usual, as by using Web 2.0 techniques such as AJAX, the micro-blog data is dynamically generated rapidly. As most microblogs' official platforms cannot offer some suitable tools or RPC interface to collect the big data effectively and efficiently, we present an algorithm on modeling and controlling on micro-blog data crawler based on simulating browsers' behaviors. This needs to analyze the simulated browsers' behaviors in order to obtain the requesting URLs to simulate and parse and analyze the sending URL requests according to the order of data sequence. The experimental results and the analysis show the feasibility of the approach. Further works are also presented at the end.

Keywords Models · Social networks · Micro-blog · Crawler

K. Gao (✉) · E.-L. Zhou
School of Information Science and Engineering, Hebei University of Science and Technology,
No. 26 YuXiang Road, Shijiazhuang, 050000, Hebei, China
e-mail: gaokai@hebust.edu.cn

S. Grover
Comrise Company, Concord Center Building 2, Hazlet, NJ, 07730, USA
e-mail: steven@comrise.com

9.1 Background

Nowadays, most of the research in the fields of mechatronic systems or social studies have spent significant effort to find rules from various complicated phenomena by principles, observations, measured data, and logic derivations. The rules are normally summarized as concise and quantitative expressions or “models,” and this can provide mechanisms to improve the system (represented by its model) performance. As for the social studies, the social network data (e.g., Twitter, Facebook, Sina_micro-blog, Tencent_micro-blog, etc.) have attracted millions of users and academic and industry researchers to research on modeling and mining the knowledge behind the magnanimity information, and as a result, there has been tremendous interest in social networks. Due to its fast development and wide usage, the microblog has attracted the attention of users, enterprises, governments, and researchers, and so applied methods and techniques for modeling and controlling in this field is very important.

As the foundation of the micro-blog data mining, data collection is the key phase, crawling or collecting the relevant micro-blog data effectively and efficiently is important. But the microblog has many differences compared with traditional web applications. For example, there are many online users, at the same time, its different interactive and displaying mode and login operation are needed, and AJAX technology is widely used, etc. Traditional web crawlers, for instance, can only get the corresponding web pages, but they cannot get the relevant structure and the corresponding social relationships as well as users’ backgrounds and fans. That is to say, being different from traditional web application, there are some differences on micro-blog data’s login operation, display way, privacy policy, data processing, etc. So the traditional web crawler is not suitable for micro-blog data crawling or collection.

This section presents some details on modeling of micro-blog data crawler based on simulating browsers’ behaviors. On the basis of this method, we have collected several million blog data in a short time period.

9.2 Motivation

Although there has been some research on AJAX-based web pages, the technique is not suitable to the micro-blog application. Encouraging developers to develop applications on micro-blog services, some providers of micro-blog services usually offer some special APIs, which can provide developers with the probability of constructing uniform and universal architecture to utilize the APIs to automatically download and save these special data. But the mere APIs-based method has some limitations on rights, calling times, special policies, and so on, and some extra tasks cannot be done by only using these official APIs.

In this chapter, we present some strategies based on simulating browsers’ behaviors to obtain the data from micro-blog platform. The main idea is to simulate browsers’

behaviors by using the browser's (e.g., FireFox) core to get the corresponding data. This can solve the problem of parsing the JavaScript code, and can do special login operation, etc. In order to crawl the data effectively, we present the following strategies: (1) focused crawling on some special crowds; (2) meta-topic searching and crawling: that is to say, we crawl the special contents by using the microblog's searching function; (3) parallel crawling: based on big data processing by using the *Redia* and *MongoDB*, we use the multiprocessing technology to download and save the data simultaneously. The proposed crawler is composed of four modules, i.e., simulating module, data crawling module, data parsing module, and data persistence module. The experimental results and the analysis show the feasibility of the approach. Further works are also presented at the end.

9.3 Related Work

Online social networking technologies enable individuals to simultaneously share information with any number of peers. With the launch of Twitter in 2007, the microblog has become highly popular, and many researchers want to investigate the micro-blog information propagation patterns [1] or analyze structures of the micro-blog network to identify influential users [2]. Reference [3] discusses some of the ways in which earlier works used text content to analyze online networks, as well as background on language coordination and the exchange-theoretic notions of power from status and dependence. Reference [4] studies several long-standing questions in media communications research, in the context of the micro-blog service Twitter, regarding the production, flow, and consumption of information. A framework which enriches the semantics of Twitter Messages (i.e., tweets) and identifies topics and entities (e.g., persons, events, products) mentioned in tweets is present in reference [5]. Reference [6] conducts a study on recommending URLs posted in Twitter messages and compares strategies for selecting and ranking URLs by exploiting the social network of a user as well as the general popularity of the URLs in Twitter. Authors of reference [7] investigate the attributes and relative influence of 1.6M Twitter users by tracking 74 million diffusion events that took place on the Twitter follower graph over a 2-month interval, and they conclude that the word-of-mouth diffusion can only be harnessed reliably by targeting large numbers of potential influencers, thereby capturing average effects. Reference [8] examines the role of social networks in online information diffusion with a large-scale field experiment, and the authors further examine the relative role of strong and weak ties in information propagation. Although stronger ties are individually more influential, it is the more abundant weak ties that are responsible for the propagation of novel information, and the authors suggest that weak ties may play a more dominant role in the dissemination of information online than currently believed. In reference [9], authors address the problem of discovering topically meaningful communities from a social network, and authors propose a probabilistic scheme that incorporates topics, social relationships, and nature of posts for more effective community discovery, and then

they demonstrate the effectiveness of the model and show that it performs better than existing community discovery models. Reference [10] examines the application of an event-driven sampling approach to the Live Journal social network, and the approach makes use of the “always on” atom feed provided by Live Journal that contains all public blog posts in near real-time to inform the sampling process of user friendship networks, and this has the effect of targeting sampling toward the public active users of the network. In addition to proposing models and algorithms for learning the model parameters and for testing the learned models to make predictions, reference [11] develops techniques for predicting the time by which a user may be expected to perform an action.

As for data crawling, in order to overcome the inherent bottlenecks with the traditional crawling, reference [12] proposes the design of a parallel migrating web crawler. Reference [13] proposes a dynamic data crawling methods, which include the sensitive checking of website changes and dynamic retrieving of pages from target websites, and the authors implement an application and compare the performance between the conventional static approaches and the proposed dynamic ones. In reference [14], authors present a novel URLs ordering system that relies on a cooperative approach between the crawlers and the web servers based on file system and web log information, and the proposed algorithm is based on file timestamps and web log internal and external counts. Reference [15] presents a micro-blog service crawler named as *MBCrawler*, which is designed on the APIs provided by micro-blog services, and the architecture is modular and scalable, so it can fit specific features of micro-blog services. Reference [16] presents a dynamic cooperation model for different crawlers’ message exchanging, and both the experimental results and the application validate the feasibility of the algorithm.

As for the modeling methods, reference [17] presents the commonly used statistical modeling methods, such as stepwise regression, radial basis function partial least squares, partial robust M-regression, ridge regression, and principal component regression that can be applied in the proposed multicollinearity domain. The Viterbi algorithm, a widely used maximum likelihood estimating method, can be used in natural language processing, and reference [18] presents an effective search space reduction for human pose estimation with Viterbi algorithm.

Although the proposed algorithm has some relationship with the above related work, there are many differences. The proposed crawler works with simulating browser behavior to collect Sina_Micro-blog (<http://weibo.com/>) and Tencent_Micro-blog (<http://t.qq.com/>) data. The proposed algorithm is based on simulating browsers’ behaviors. As for browser, reference [19] describes how to calculate various object-oriented metrics of three versions of Mozilla Firefox, and the neural network approach can predict high and medium severity errors more accurately than low severity errors.

The experimental results and the analysis show the feasibility of the proposed approach.

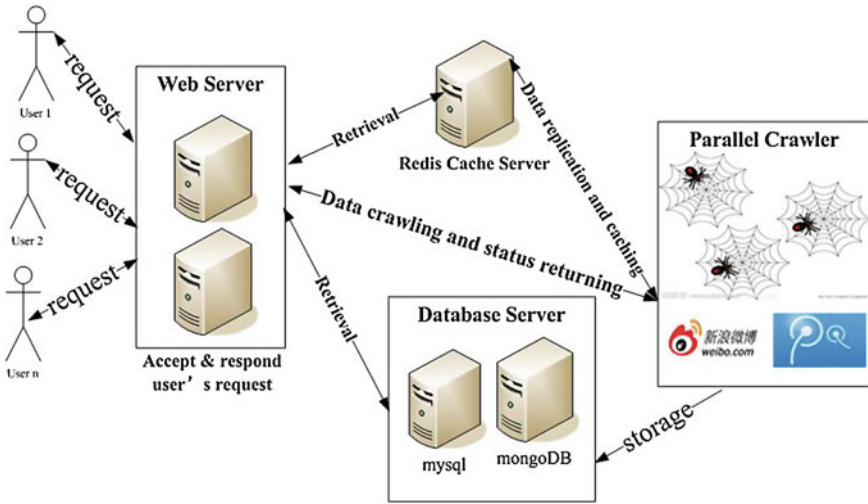


Fig. 9.1 System architecture

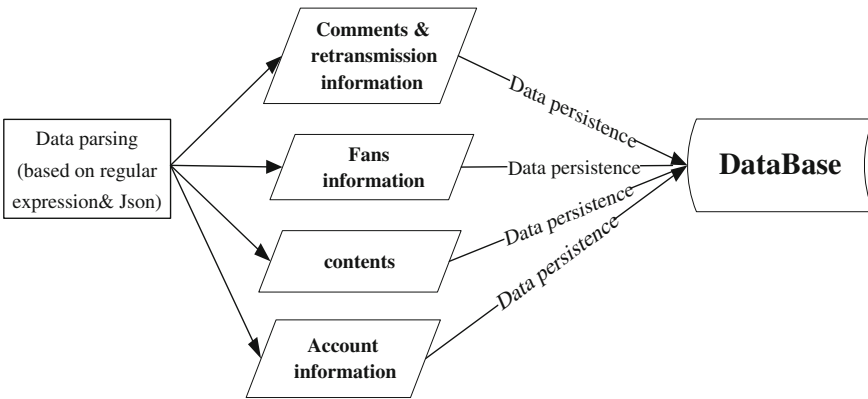


Fig. 9.2 Parsed data and its persistence

9.4 System Architecture

Social networks are often huge, and therefore crawling the micro-blog data could be both challenging and interesting. As microblog’s big data properties, it is impossible to crawl all the micro-blog data. Instead, it is feasible to crawl some kinds of data (e.g., account information, contents or topics, attentions or fans, etc.). In this section, we propose the system architecture on parallel crawling. Figure 9.1 shows the architecture, and Fig. 9.2 shows the parsed data and its persistence.

In practice, we can use the *RDBMS* to store the parsed data, and the *Redis* is used as the cache server, so users’ retrieval request can be done through the web server.

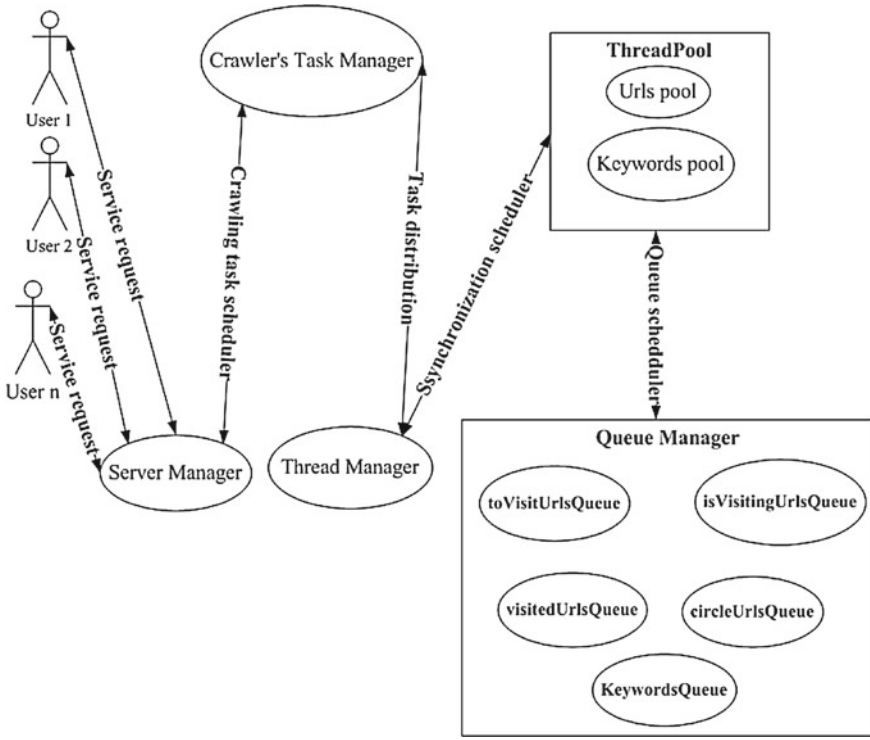


Fig. 9.3 Multi-thread-based parallel crawling

In detail, as for the multi-thread based parallel crawling, we use a thread pool and a queue manager to schedule the tasks. There are several different queues, including the *toVisitUrlsQueue*, *isVisitingUrlsQueue*, *visitedUrlsQueue*, *circleUrlsQueue*, *keywordsQueue*, etc., see Fig. 9.3.

9.5 Case Studies and Implementation of the Simulated Browser-Based Crawling

Instead of merely using the official APIs, we propose a simulated browser-based crawling, as the merely APIs-based method has some limitations on rights, calling times, and so on, and perhaps some extra tasks cannot be done by only using official APIs. We present some strategies based on simulating browsers' behaviors to obtain the micro-blog data, and the proposed crawler is composed of four modules, i.e., simulating module, data crawling module, data parsing module, and data persistence module.



Fig. 9.4 Different situations with the same account

9.5.1 Simulation of the Login Operation and Cookies Data Obtaining

Commercial websites often use technologies (e.g., HTTP compression, SSL encryption and chunked encoding) to provide some reasonable levels of security and system performance. As for the micro-blog data effectively crawling, simulated login operation is necessary. Otherwise (for example, by only using official APIs-based crawling), only few data can be crawling. Here, the simulated login operation means this kind of operation allows the crawler to use some legal accounts and their corresponding passwords to login the corresponding micro-blog platforms, and the key phase is the encrypted data parsing. Here, we use the HttpWatch [20], which integrates with Internet Explorer or Mozilla Firefox to provide some unrivaled levels of HTTP monitoring, without the need for separately configured proxies or network sniffers. Simply interacting with a website, HttpWatch can display a log of requests and responses alongside the web page itself, and it can even show interactions between the browser and its cache. As a result, each HTTP transaction can be examined or parsed so as to see the values of headers, cookies, query strings, and other HTTP-related data. HttpWatch can work well with these technologies to provide a view of HTTP activity. By using HttpWatch, we can obtain 21 or more different parameters during the simulated login phase. But in practice, there usually exist some different situations during the login phase, and Fig. 9.4 shows the two situations when using the same account.

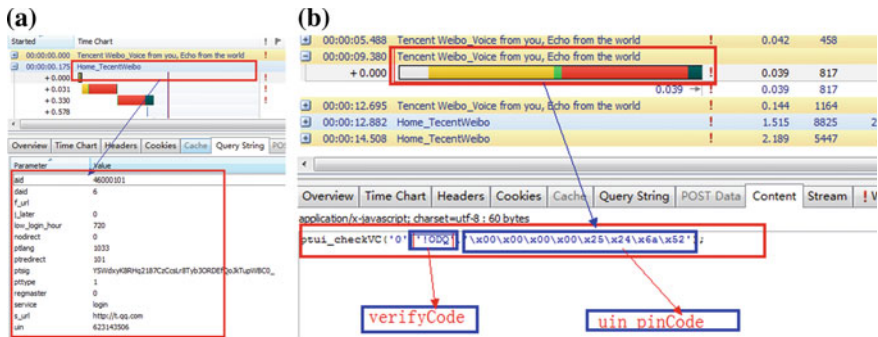


Fig. 9.5 Requested preliminary parameters (a) and the return values after the requested period (b)

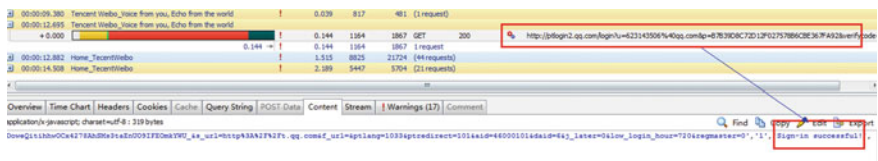


Fig. 9.6 Returned results

From the parsed results, we can conclude that the requested URLs usually contain static and dynamic parameters (e.g., the parameter P and $verifycode$ in Fig. 9.4, and the $verifycode$ parameter is usually used as the password encryption). As the microblogs' login passwords are usually multilevers and multilayer encrypted, it usually contains some other preliminary parameters, and the displayed parameter in Fig. 9.5b (i.e., “ $\backslash\x00\backslash\x00\backslash\x00\backslash\x00\backslash\x25\backslash\x24\backslash\x6a\backslash\x52$ ”) is the encrypted parameter in Fig. 9.4a. Now the encrypted resolving phase is finished, and the returned or parsed content is shown in Fig. 9.6.

As for judging whether the corresponding user is legal or illegal, it needs to analyze the cookies data. On the other hand, whether actually login or not, when requesting the server data, if the user can get the legal cookies, he or she can obtain the same data as if he or she really “login” the web server. In detail, in order to obtain the cookies data, it needs three steps. First, it needs to obtain the $verifycode$ and uin parameters, see algorithm 1 below. Second, by using the JavaScript analysis engine and invoking the encrypted function, we can obtain the parsed parameters, see Fig. 9.7. Last, it needs to merge the relevant parameters to obtain the corresponding cookie data, which is the result of the simulating login phase, see algorithm 2 below, and the parsed cookies data result is shown in Fig. 9.8.


```

//parsing the verifyCode and uin parameters
Algorithm 1 (i.e., void getCheckVC())
Input:
(1) preLoginUrl//requested URLs before the login operation;
(2) username
(3) host//parameter on requesting the preLoginUrl
Output:
(1) retJson: returned parameters on preLoginUrl request
(2) verifyCode
(3) uin
Step1. retJson=getPreLoginJson(preLoginUrl,username,host);
Step2. verifyCode=parseVerifyCode(retJson);
Step3. uin=parseUin(retJson);

```

```

//obtaining the cookie data
Algorithm 2 (i.e., void getCookies())
Input:
(1) loginUrl // Url to login the corresponding micro-blog page
(2) username
(3) password
(4) verifyCode //verification code obtained from the former step
(5) host
(6) referrer// means the source urls to do the login operation
Output: cookie data
Step1. cookie=getCookies(loginUrl,username,password,verifyCode,
host,referrer);
Step2.cookie=checkValidate(cookie) //verify the cookie
Step3.transmitCookie(cookie)//transfer the obtained cookies to
the corresponding threads

```

9.5.2 Data Parsing and Persistence

After collecting the corresponding data, it needs to be parsed and stored. As for the content, there are some differences between data within the blogger's main page and other common pages. As usual, the main page returns data in a traditional way, while other common pages usually use AJAX [21] and JSON [22] technology to return data to client in order to enhance the performance or optimize the user experience. By using JavaScript, these returned data can be parsed and filled into the corresponding sites. Figure 9.9 shows the private crawled data, and algorithm 3 shows the main steps of this processing step.

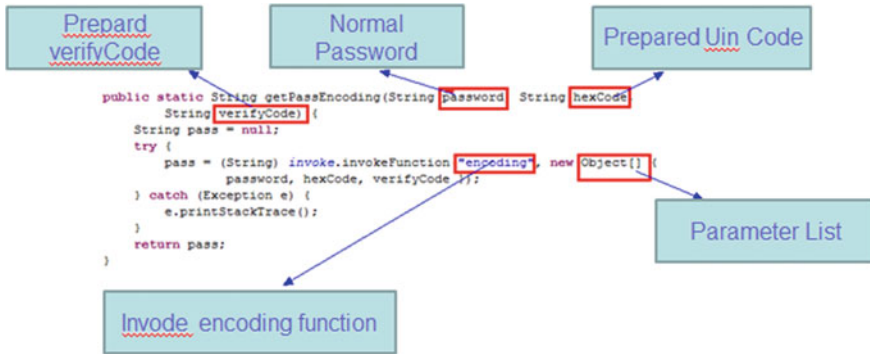


Fig. 9.7 The parsed parameters

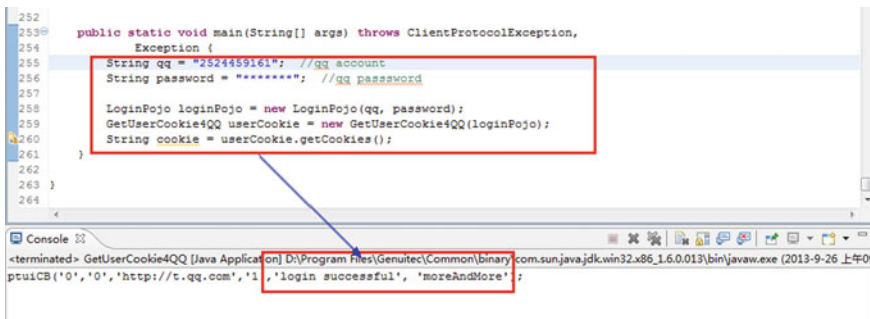


Fig. 9.8 The cookies data of the simulating login phase

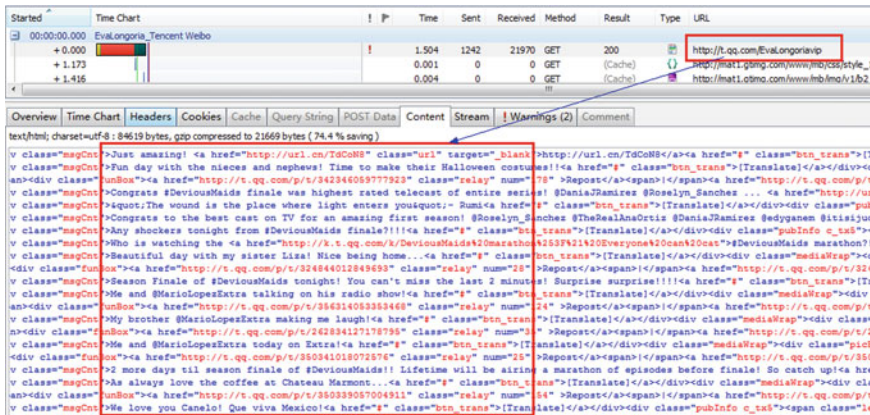


Fig. 9.9 The crawled content

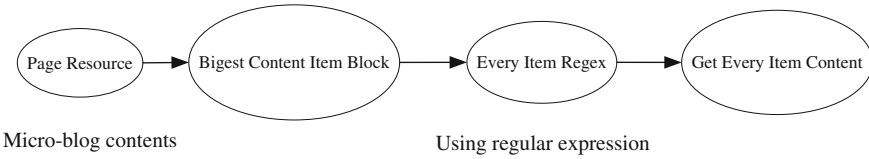


Fig. 9.10 The processing flow

```

// Get the content of corresponding Urls
Algorithm 3 (i.e., grabOnePageArticle(String url, boolean
isFirst))
Input:
(1) Url //pages to be crawled;
(2) Url_Refer //one of the header parameter within the HTTP
request;
(3) Url_Host //one of the header parameter such the port
number, etc.
Output:
(1) content.
Step1. Url=getCompleteUrl(time,currentPage,aid,uid) // Fill in
the Url parameters dynamically, including the current time,
currentPage, aid parameter (i.e., id parameters of the micro-
blog contents, see Fig.\, \ref{fig:9}), uid parameter
(i.e., user name).
Step2. content=grabPageSource4QQ.grabPageSourceOfQQ(Url,
Url_Refer, Url_Host);
  
```

The data parsing module needs to process the crawled data and parse the micro-blog content. The crawled data can be classified into two corpus, i.e., plaintext and cipher text. As the plaintext is regular and uniform, we use the regular expression to extract the real contents (e.g., the micro-blog contents, Url, id, published time, IP address, reviewed or commented number, forwarded number, etc.). The processing flow is shown in Fig. 9.10.

9.6 Experimental Results and Analysis

9.6.1 About the Testing Data Set and the Experimental Environment

As for these micro-blog big data, its persistence is an important issue. We use the *MongoDB*, *Redis* to store and cache them in our real application, and *Mysql* is

(a)			(b)			
url	name	sex	sendUrl	dstUrl	article	person_id
http://t.qq.com/Evalongoriaip	Evalongoria	female	http://t.qq.com/Evalongoriaip	http://t.qq.com/p/163872010546035	Just amazing! http://url.cn/TdCoN8 [Translate]	Evalongoriaip
http://t.qq.com/Evalongoriaip	Evalongoria	female	http://t.qq.com/Evalongoriaip	http://t.qq.com/p/134246059777923	Fun day with the nieces and nephews! Time to make their Halo	Evalongoriaip
http://t.qq.com/PaulScheer	PaulScheer	mail	http://t.qq.com/Evalongoriaip	http://t.qq.com/p/139346042208598	Congrats #DeviciousMaidns finale was highest rated telecast of e	Evalongoriaip
http://t.qq.com/JohnStamos	JohnStamos	mail	http://t.qq.com/PaulScheer	http://t.qq.com/p/1348849124794899	@AnthonyStead is a very bad man who hates San Diego on th	PaulScheer
http://t.qq.com/KaylaCollins	KaylaCollins	female	http://t.qq.com/PaulScheer	http://t.qq.com/p/1317348059180012	My wife with @BakuLunaCline (New Girl: Fight of the Conchords,	PaulScheer
http://t.qq.com/SnoopDogg	SnoopDogg	mail	http://t.qq.com/PaulScheer	http://t.qq.com/p/1313848030426112	Meeting an important Hollywood friend to discuss a project!@	PaulScheer
http://t.qq.com/Nacho_Polo	Nacho_Polo	mail	http://t.qq.com/PaulScheer	http://t.qq.com/p/1310387092696122	NTSF5DSLUV: is going to London (for Real!) for our TV Movie.	PaulScheer
http://t.qq.com/JeffKilburg	JeffKilburg	mail	http://t.qq.com/OlandoJones	http://t.qq.com/p/134997129217626	Hah... well... The more you know...!	OlandoJones
http://t.qq.com/CarolyHemery	CarolyHemery	mail	http://t.qq.com/OlandoJones	http://t.qq.com/p/1258643107676980	Seriously Congress... can't you get your freaking priorities strai	OlandoJones
http://t.qq.com/NickiReed	NickiReed	female	http://t.qq.com/AliciaWitt	http://t.qq.com/p/1227077002802688	10 minutes to showtime! http://url.cn/P9TawJ [Translate]	AliciaWitt
http://t.qq.com/AbigailGencer	AbigailGencer	female	http://t.qq.com/p/1213345120290114	http://t.qq.com/p/1213345120290114	Nov. 13, 1994: Last time the Bengals, Browns and Lions won o	DarrenRovell
http://t.qq.com/BonnieStallin	BonnieStallin	female	http://t.qq.com/DarrenRovell	http://t.qq.com/p/132134042720847	Today's payouts: Northwestern paying Maine \$450K, Nebraska	DarrenRovell
http://t.qq.com/LukeBilyk	LukeBilyk	mail	http://t.qq.com/DarrenRovell	http://t.qq.com/p/132134042720847	New cover for all my Chinese friends!!! Harpers Bazaar China -	cocorochoaonline
http://t.qq.com/JamesPfeifer	JamesPfeifer	female	http://t.qq.com/cocorochoaonline	http://t.qq.com/p/13193480120476115	Wilhelmina Models takes you behind-the-scenes on Coco Rod	cocorochoaonline
http://t.qq.com/windy_official	windy_official	mail	http://t.qq.com/cocorochoaonline	http://t.qq.com/p/13193480120476115	Watch with Allure Magazine as I chop off my long hair into a pi	cocorochoaonline
http://t.qq.com/jamiedcswip	jamiedcswip	mail	http://t.qq.com/cocorochoaonline	http://t.qq.com/p/1276017069296076	So excited to be at the #MetropolitanOpera for opening night	cocorochoaonline
http://t.qq.com/CarrieAnruba	CarrieAnruba	female	http://t.qq.com/cocorochoaonline	http://t.qq.com/p/1243927112474143		cocorochoaonline

Fig. 9.11 The parsed results, a account information b other parsed content

only used as the experimental platform. Figure 9.11 shows the *MySQL*-based parsed Tencent_Micro-blog data.

In order to evaluate the algorithm’s performance, we classify the following tested data set according to their authorities into three classes. The reason we use the following three classes is that the micro-blog official platform usually presents the following three different classes, and the three classes have different data sizes. In detail, the first class has less data while the two others have more. In detail, the first class is the ordinary (i.e., minor authorities) microbloggers, whose propagative scope is only limited within old friends or classmates; the second class is those medium authorities’ microbloggers (e.g., network magazines microblog), whose contents can be followed or spread into all kinds of users; the last class is famous persons’ microblogs. We will test the performance on the above different data environment.

9.6.2 Ordinary Microblogger’s Performance Evaluation

In this section, we use someone’s microblog as an example. Figure 9.12a shows the original microblog, and b shows the crawled and parsed data, respectively. It is clear that there is no difference between them, and all contents have been obtained correctly.

9.6.3 Medium Authorities’ Microblogger Performance Evaluation

Here, we use some medium authorities’ microbloggers as the experimental platform. Figure 9.13a shows the micro-blog interface, while b shows the crawled and parsed data.

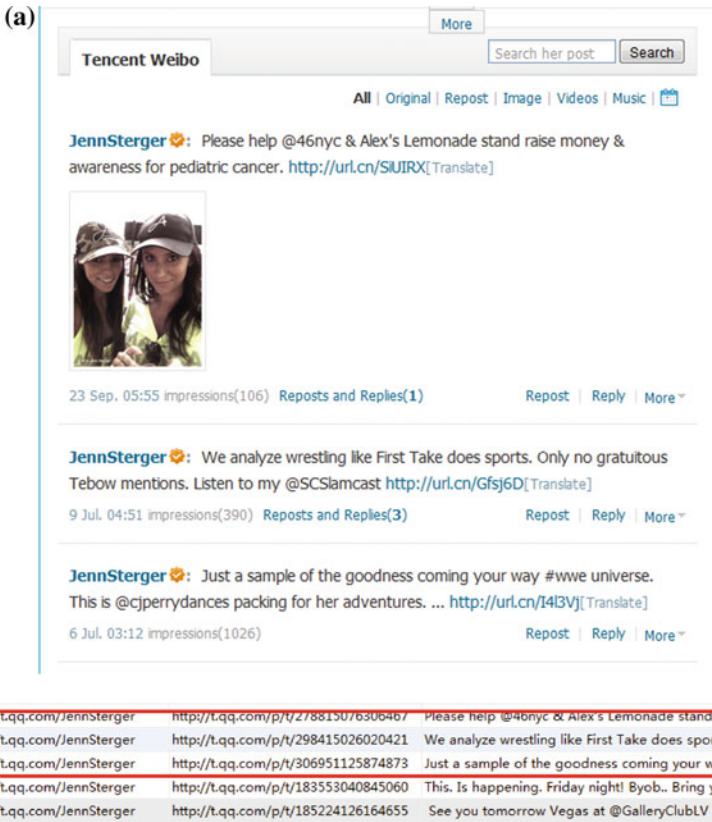


Fig. 9.12 Ordinary and the parsed results of the ordinary microblogger's, a ordinary data, b parsed results

9.6.4 Famous Persons' Microblog

As for these famous persons' microblogs, we use the famous actor Tom Cruise's microblog as an example. Figure 9.14a shows his micro-blog interface, while b shows the crawled and parsed data.


9.6.5 Performance Evaluation

Sometimes micro-blog services provide some APIs. Through these services, the well-structured data can be easily obtained, so it can provide us the probability of constructing uniform and universal software architecture to utilize the provided APIs to automatically download data. However, there are usually some limits and

(a)


All | Original | Repost | Image | Videos | Music | 📁

MarlonWayans 🗨️: NEW YORK!!! WAYANS BROS October 10-13, 2013 Levity Live West Nyack, NY <http://url.cn/L7m5g5> [Translate]



17 minutes ago Impressions(2774) Reposts and Replies(7) [Repost](#) | [Reply](#) | [More](#) ▾

MarlonWayans 🗨️: Somehow @rickmalvarez got a #selfy of himself and me in a fucked sleepy position in the background. 5 minute naps k... <http://url.cn/Qe5MuW> [Translate]



Today 10:18 Impressions(11k) Reposts and Replies(6) [Repost](#) | [Reply](#) | [More](#) ▾

MarlonWayans 🗨️: More #whiteguysshoegame#dallasmaverickscolor #whiteguyteam [Translate]

(b)

http://t.qq.com/Marlon	http://t.qq.com/p/t/336350070906824	NEW YORK!!! WAYANS BROS October 10-13, 2013 Levity Live V
http://t.qq.com/Marlon	http://t.qq.com/p/t/279521038122269	Somehow @rickmalvarez got a #selfy of himself and me in a fi
http://t.qq.com/Marlon	http://t.qq.com/p/t/243931055370641	More #whiteguysshoegame#dallasmaverickscolor #whiteguytea
http://t.qq.com/Marlon	http://t.qq.com/p/t/334849119277361	In my dirty car headed to set... 4 more days!!!! @ahhmovie 2[T
http://t.qq.com/Marlon	http://t.qq.com/p/t/323849108737944	All I've done for the past few months is work, workout and wor

Fig. 9.13 Medium authorities' microblogger and the corresponding parsed result, a ordinary data, b parsed results

obstacles. In order to evaluate the performance, we present the comparison between APIs-based crawling and the simulating browser behavior approach, see Table 9.1.

From the above comparison, as for the proposed method, it is clear that the parsed data's degree of integrity and the accuracy or scope is higher. But, as shown before, if the template or the main framework of the microblog has been changed, the accuracy of the parsed data is lower than usual. Fortunately, these changes occur rarely. If we can track or analyze the parsed data periodically, it is easy to find the changes and then revise some special rules to parse the corresponding data.

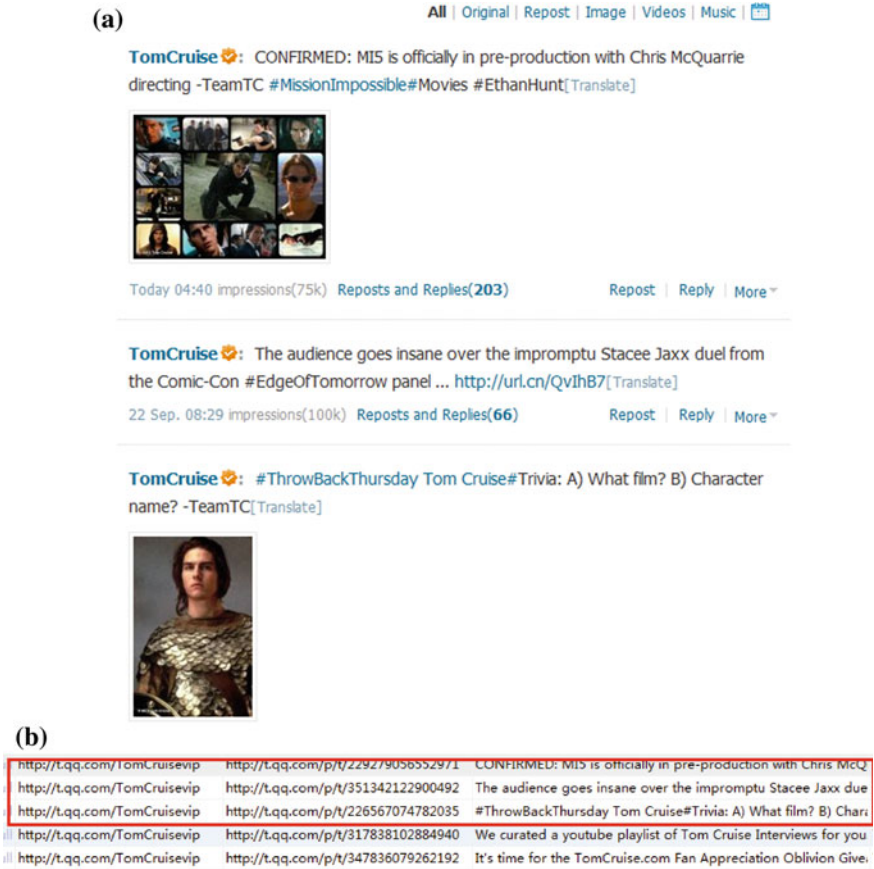


Fig. 9.14 Famous persons’ micro-blog and the parsed results, **a** ordinary data, **b** parsed results

9.7 Conclusion

It is hard for a traditional web page crawler to crawl micro-blog data as usual, and most microblogs’ official platforms cannot offer some suitable tools or RPC interfaces to collect the data effectively and efficiently. This chapter presents some algorithms and strategies on crawling and parsing micro-blog data effectively based on simulating browsers’ behaviors. This needs to analyze the simulated browsing behavior in order to obtain the requesting URLs, to simulate and analyze the sending URLs requests according to the order of data sequence. It needs to focus on crawling on some special crowds and crawl some special contents by using the microblog’s searching function. Parallel crawling and the multiprocessing technology are also used to download the data simultaneously. The experimental results and the analysis show the feasibility of the approach. Existing works are also presented at the end.

Table 9.1 The performance comparison of the two approaches

Performance	Algorithms	
	The proposed simulating browser behavior-based crawling	Official APIs-based crawling
About the micro-blog data entrance	Micro-blog platform account	(1) Micro-blog platform account (2) AppKey (3) AccessToken
About the data usability	Flexible and easy to use	(1) Existing more limitations such as fixed or lower crawling frequency, it is not flexible (2) The parsed contents are usually fixed and researchers cannot choose some special categories from them
About the multi-threads performance	Better	Poor
About the crawling frequency	Flexible, and the only limitation is the bandwidth	Not flexible, as there are more limitations such as API invoking rights or authorization
About the stability	Better. The only limitation is the variation of the Official's platform, but the variation is rare	Better
About the data integrity	Better, and almost all needed data can be parsed	Poor, and some data cannot be parsed such as the <i>docUrl</i> , etc. <i>DocUrl</i> parameter can be seen from the Fig. 9.12b
About visiting or data requested frequency	Larger than 60 times (per minutes)	Less than 1 times (per minutes)
About the degree of data integrity	100 %	95 %, lacking some materials such as <i>docUrl</i> , etc
About the data accuracy	99 %	100 %
About the data scope	Widely, and almost all the data appearing on the micro-blog platform can be crawled and parsed	Narrowly, and the only small new parts, such as microbar, microgroup, etc., can be obtained

Acknowledgments Some earlier works were done in Beijing Institute of Technology with the help of Dr. Hua-ping Zhang and Prof. Yin-ping Zhao. This work is sponsored by the National Science Foundation of Hebei Province (No. F2013208105) and the National Science Foundation of China (No. 61272362). It is also sponsored by Hebei Province Scientific and Technical Key Task (No. 12213516D).

References

1. Kwak H, Lee C, Park H, Moon S (2010) What is Twitter, a social network or a news media? In: 19th international conference on world wide web. ACM Press, USA, pp 591–600
2. Weng J, Lim EP, Jiang J, He Q (2010) TwitterRank: finding topic-sensitive influential twitterers. In: 3rd international conference on web search and web data mining. ACM Press, USA, pp 261–270
3. Cristian DNM, Lee L, Bo P, Kleinberg J (2012) Echoes of power: language effects and power differences in social interaction. In: 21th international conference on world wide web. ACM Press, France, pp 699–708
4. Wu S, Hofman JM, Mason WA, Watts DJ (2011) Who says what to whom on Twitter. In: 20th international conference on the world wide web. ACM Press, India, pp 705–714
5. Abel F, Gao Q, Houben GJ, Tao K (2011) Analyzing user modeling on Twitter for personalized news recommendations. In: International conference on user modeling, adaptation and personalization. LNCS, vol 6787. Springer, Spain, pp 1–12
6. Chen J, Nairn R, Nelson L, Bernstein M, Chi E (2010) Short and tweet: experiments on recommending content from information streams. In: 28th international conference on human factors in computing systems. ACM Press, USA, pp 1185–1194
7. Bakshy E, Hofman JM, Mason WA, Watts DJ (2011) Everyone’s an influencer: quantifying influence on Twitter. In: 3rd international conference on web search and data mining. ACM Press, Hong Kong, pp 65–74
8. Bakshy E, Rosenn I, Marlow C, Marlow C (2012) The role of social networks in information diffusion. In: 21th international conference on world wide web. ACM Press, France, pp 519–528
9. Sachan M, Contractor D, Tanveer AF, Subramaniam LV (2012) Using content and interactions for discovering communities in social networks. In: International conference on world wide web. ACM Press, France, pp 331–340
10. Dan C, Shipman FM (2009) Capturing on-line social network link dynamics using event-driven sampling. In: International conference on computational science and engineering, vol 4. Vancouver, Canada, pp 284–291
11. Goyal A, Bonchi F, Lakshmanan LV (2010) Learning influence probabilities in social networks. In: 3th international conference on web search and data mining. ACM Press, USA, pp 241–250
12. Agarwal A, Durgesh S, Pandey AKA, Goel V (2012) Design of a parallel migrating web crawler. *J Adv Res Comput Sci Softw Eng* 2(4):147–153
13. Kim KS, Kim KY, Lee KH, Kim TK, Cho WS (2012) Design and implementation of web crawler based on dynamic web collection cycle. In: International conference on information networking (ICOIN). Bali, Indonesia, pp 562–566
14. Chandramouli A, Gauch S, Eno J (2012) A cooperative approach to web crawler URL ordering, human–computer systems interaction: backgrounds and applications. *J Adv Intell Soft Comput* 98:343–357
15. Lu G, Liu S, Lü K (2013) MBCrawler: a software architecture for micro-blog crawler. In: International conference on information technology and software engineering. Lecture Notes in Electrical Engineering, vol 212. Springer, Berlin, Heidelberg, pp 119–127
16. Gao K, Li SW (2010) The cooperation model for multi agents and the identification on replicated collections for web crawler. *Int J Model Identif Control* 11(3–4):224–231

17. Garg A, Tai K (2013) Comparison of statistical and machine learning methods in modelling of data with multicollinearity. *Int J Model Identif Control* 18(4):295–312
18. Han G, Zhu H, Ge J (2013) Effective search space reduction for human pose estimation with Viterbi recurrence algorithm. *Int J Model Identif Control* 18(4):341–348
19. Singh S, Mittal P, Kahlon KS (2013) Empirical model for predicting high, medium and low severity faults using object oriented metrics in Mozilla Firefox. *Int J Comput Appl Technol* 47(2/3):110–124
20. HttpWatch: Introduction to HttpWatch 8.x (2013). <http://help.httpwatch.com/#introduction.html>
21. Ajax: Introduction to Ajax (2013). <http://api.jquery.com/category/ajax/>
22. Json: Introduction to Json (2013). <http://www.json.org/index.html>