A Real-time Search Framework of Microblog Based on Partial Indexing Mechanism

Bairong Wang
School of Business Administration
Northeastern University
Shenyang, China
E-mail: nihaoalison@163.com

Jingbo Yuan
Institute of Information Management Technology and Application
Northeastern University at Qinhuangdao
Qinhuangdao, China
E-mail: jingboyuan@hotmail.com

Jihao Yang
College of Information Science and Engineering
Northeastern University
Shenyang, China
E-mail: yangjihao1@126.com

***Abstract*—Along with the development of information technology and wide applications of network, information retrieval plays a more and more important role in people's life. But as the Twitter kind of short message service appears, a new problem has emerged in the world of information retrieval, that is, real-time search technology. Real-time search will be a demand of the Internet's development. With Sina microbolg for analysis, a new real-time search framework based on meta-search engine has been put forward in this paper. Moreover, a microblog classification and indexing process on basis of partial indexing mechanism has also been designed. The framework can effectively save the real-time search costs and obtain microblog data with high quality.**

Keywords-real-time search; microblog; partial indexing; meta-search engine

I. INTRODUCTION

There has been a dramatically increasing demand of real-time information year after year with the further development of Internet. Traditional search engine, due to its own flaws, can not meet people's need for real-time information retrieval, so the real-time search arises at the historic moment. As a new media platform, microblog has aroused wide concern in recent years. People become more and more interested in information from microblog. As a social network, a huge amount of information will be generated in a microblog every second as a result. The information amount is too large and characterized by clutter and short-life cycle [1]. How to get the latest and high quality information from the huge and jumbled information becomes the focus of the problem. The real-time network search, such as Twitter [2], has become one of the most popular applications of real-time search technology [3-5].

In brief, Real-time Search is performing fast and instant search for information on the Internet as well as realizing the search asking effect. The appearance of the real-time search will make the Internet more instant-changing, convenient and simple. Users can quickly get fresh first-hand grassroots information with the help of real-time search. The domestic and international events can be more rapidly learnt, too. The biggest characteristic of microblog is too much content and higher updating speed. But microblog's quick spreading will take its users more time and energy to select useful information. The real-time search, however, can fundamentally change this situation which is also the biggest nature of real-time search.

Although some search engine giants have already carried out the real-time search business, the retrieval efficiency and quality are still not satisfying. Therefore, a classification way for queries and microblogs has been put forward based on the partial indexing mechanism. Partial indexing, namely classified index, is also used to improve retrieval efficiency of the system. At the same time, a new framework has been designed on the foundation of meta-search engine, taking the retrieval mechanism of a microblog system itself as information filtering interface to save the cost of information filtering of the real-time search system and improve the microblog's information quality.

II. ANALYSIS OF MICROBLOG REAL-TIME SEARCH SYSTEM

Today's real-time search operators, without an exception, have set up their own social network. The goal is to build own real-time search database. Some service providers even co-operate with outstanding social networking websites in the data sharing, like Baidu and Sina's microblog. Nowadays, most of the microblog systems have lunched their own internal search. Taking Sina microblog [6] as a case, on the users' homepages of the search interface, when a query is submitted, the system will provide user search and microblog search two categories on a drop-down menu. The homepage also contains extended topics and hot search list to recommend users the related and latest information corresponding to the microblog's "Push" data acquisition mode. And their friends' updating shown in the user's page are corresponding to the micro-blog's "Pull" data acquisition mode.

It is not difficult to find that data of Sina microblog search almost come from its own system and not from other search systems, which will often result in certain information lag. For example, the latest news that the Japanese Yen and the RMB can be directly exchanged is displayed at 8:30, the time is almost two and half hours earlier than the query, which is absolutely unacceptable to the real-time search. Therefore, the real-time search should firstly expand their data sources to grab the freshest information. Secondly, since the microblog operators have developed their own retrieval system and are able to filter information and rank popularity, the real-time search engine

Published by Atlantis Press, Paris, France.

can completely imitate meta search engine model, taking the microblog retrieval system as information filtering interface. The real-time search engines are responsible for reordering and outputting the retrieval results.

III. DESIGN OF THE REAL-TIME SEARCH BASED ON SINA MICROBLOG

The number of registered users in Sina microblog has an excess of several hundred million, and about 100 million users use the service every day. To ensure the system's efficiency, the real-time search can be modeled as meta search engine and establish the search service on the basis of various microblog search, in this way it can avoid filtering and indexing all the microblogs. The real-time search system of microblogs will use the retrieve ranking mechanism and only store those microblogs higher ranked by the ranking function.

The framework modeled as meta-search engine includes request submitting agent, cooperation websites, converter, classifier, index apparatus, retrieval device and the results showing agent. The structure of the real-time search is shown in figure 1.

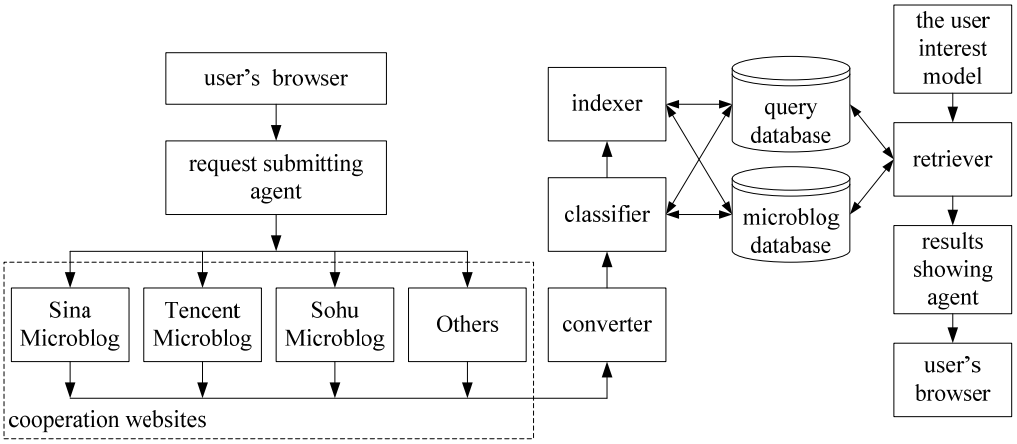

Figure 1. The structure of the real-time search

A. *Request submitting agent*

Request submitting agent is responsible for implementing the users' personalized searching demands, such as calling which microblog search engines, setting the limits of searching time and the number of results, etc. Moreover, it is in charge of distributing the requests to independent microblog systems.

B. *Cooperation websites*

Cooperation websites are mainly the microblog sites, such as Sina microblog, Tencent microblog, etc. Of course, the real-time search can also adopt other websites' information in order to enrich the sources of information. The cooperation websites provide the real-time system with latest data and improve the instantaneity of real-time search.

C. *Converter*

Data formats submitted by each cooperation website are very different, therefore, these data will be cut and unified and the useful information will be extracted afterwards in the converter. All the work is prepared for ranking calculation.

D. *Classifier*

The partial indexing mechanism has to be established on the classification of query and microblog. The classifier is mainly responsible for classifying microblog. Meanwhile, it can also call the data in the database and conduct updating as well as maintenance constantly.

E. *Indexer*

The classifier is responsible for the classification of query and microblog, the indexer mainly adopt different indexing strategy according to the results of the classifier. At the same time, the indexer will update the database regularly, clear out the data more than a certain time range in order to keep the system running at high speed.

F. *Retriever*

The retriever can search the database according to the customers' demands and output the qualified results. In addition, the retriever will learn automatically users' interest models in accordance to the statistics. In this way it can provide users with real-time search suggestions and help users find useful information more quickly.

G. *Results showing agent*

The results showing agent will classify the results and push the popular information to users, and the results will be shown on the homepage and updated continually.

The framework is built on combine with other microblog systems, other microblog systems shoulder the task of information filtering also, which can save the cost of information filtering and indexing and achieve a better effect of real time. However, owing to the differences among data structures, the system needs to unify the data formats before ranking and index. This is inevitable defect of meta-search engine architecture, but comparing with the overhead caused

by filtering all the crude microblogs data, the framework structure still has some advantages.

IV. CLASSIFICATION AND INDEXING PROCESS OF MICROBLOG BASED ON THE PARTIAL INDEXING MECHANISM

A. Classification of microblogs and queries

A big challenge confronting the microblog indexing is how to determine the importance of microblogs. Allowing for the limitation in the number of characters and information content of a microblog, the system can judge the popularity of a topic in reference to users' queries. According to the topic popularity these microblogs belong to, it will divide microblogs into effective microblogs and noisy ones. Finally, the system will conduct real-time indexing on effective microblogs and batch indexing on noisy microblogs respectively in order to satisfy the system's instantaneity.

Definition 1: given a microblog *t* and users' query keywords set *Q*, if for any $q \in Q$, $f(q, t) <= K$, then *t* is effective microblog, or it will be noisy microblog.

Definition 2: given a query *q*, *T* is the microblog set whose microblogs match the keywords of query *q*, the number of microblogs in set *T* is recorded as *T.num*. The popularity value of *q* equals *T.num*, recorded as *Cq*.

Definition 3: given a query *q* and its popularity *Cq*, if $Cq >= M$ (*M* for the system threshold), then *q* is hot query, or *q* for candidate query.

Definition 4: given a microblog set *T*, if all its microblogs are established through reply or forward relationship, then *T* can be defined a microblog tree and stored with tree structure.

A survey on search engine [7] shows that 62 of users click on the first page of results and More than 90 of users don't browse the contents of after the third page. As a result of real time search is updated automatically, real time search need simply to Index the number of records of 3 pages of the search results. *K* is calculated according to the following formula (1).

$$K = C_1 * R \tag{1}$$

Where, $C_1=3$, *R* is the number of records of each page of the search results. Systems can also make adjustments factor C_1 as needed.

For each question, the system stores only the *M* related microblog records, in which, the top *K* microblogs are valid microblogs and microblogs from *K+1* to *M* are noise microblogs. That is, the system will store *M-K* noise microblogs to meet the needs of the remaining 10% users. *M* is calculated according to the following formula (2).

$$M = C_2 * R \tag{2}$$

Where, $C_2=5$, *R* is the number of records of each page of the search results. Systems can also make adjustments factor C_2 as needed.

B. Microblog classification mechanism with partial indexing

1) The overview of partial indexing

The idea of partial indexing emerged as early as the 19th century. It was first introduced in literature [8], in which the advantage in terms of time performance was also analyzed. In short, partial indexing will not index the whole database, instead it just index microblogs that are very likely to be the results of a certain query. In this paper the query is divided into hot query and candidate query, and microblog is divided into effective microblog and noisy microblog respectively. For the effective microblogs of hot queries, the system will do real-time indexing and for the noisy ones use batch indexing with offline mode. All microblogs of candidate queries are with batch indexing so as to improve the system response time and efficiency.

2) Microblog classification and indexing process

Partial indexing mechanism should take query and microblog's classification as foundation. Therefore, classification should be done before partial indexing, the whole process is mainly divided into the following six steps, and its work flow is shown in figure 2.

Step 1: Once a microblog *ti* entered, the system will first judge the type of *ti*. If the microblog belongs to a tree structure, then it will be stored in a tree structure with an increase of the tree's popularity afterwards. If it is a single node structure, then go to Step 2.

Step 2: The system will match the keywords of *ti* with hot queries, if it does not contain the keywords of any hot query, then enter the Step 3. If it contains the keywords of a hot query, popularity of this query will be updated and go to Step 5.

Step 3: The system will match the keywords of *ti* with candidate queries, if it does not contain candidate queries' keywords, then a new query keyword will be created. Meanwhile, the system will record the query's popularity as well as the microblogs' ranking results, and conduct batch indexing on *ti*. If it contains the keywords of this candidate query, then go to Step 4.

Step 4: The system will updated the popularity of this candidate query and make a comparison of popularity *Cq* with system threshold *M*. If $Cq < M$, the query is still a candidate one, the system will conduct ranking calculation and batch indexing afterwards. If $Cq>M$, this candidate query will upgrade for hot query and go to Step 5.

Step 5: The system will calculate ti's ranking. If *score(ti)<score(tk)*, then ti is noisy and batch indexed. If the score is higher than that of the Kth microblog in the query's set, namely *score (ti)* $\geqq$ *score (tk)*, then *ti* belongs to effective ones and go to Step 6.

Step 6: The system will re-rank all effective microblogs belong to this hot query, if *ti* ranked higher than ***K***, then it for real-time indexing, or for batch indexing.

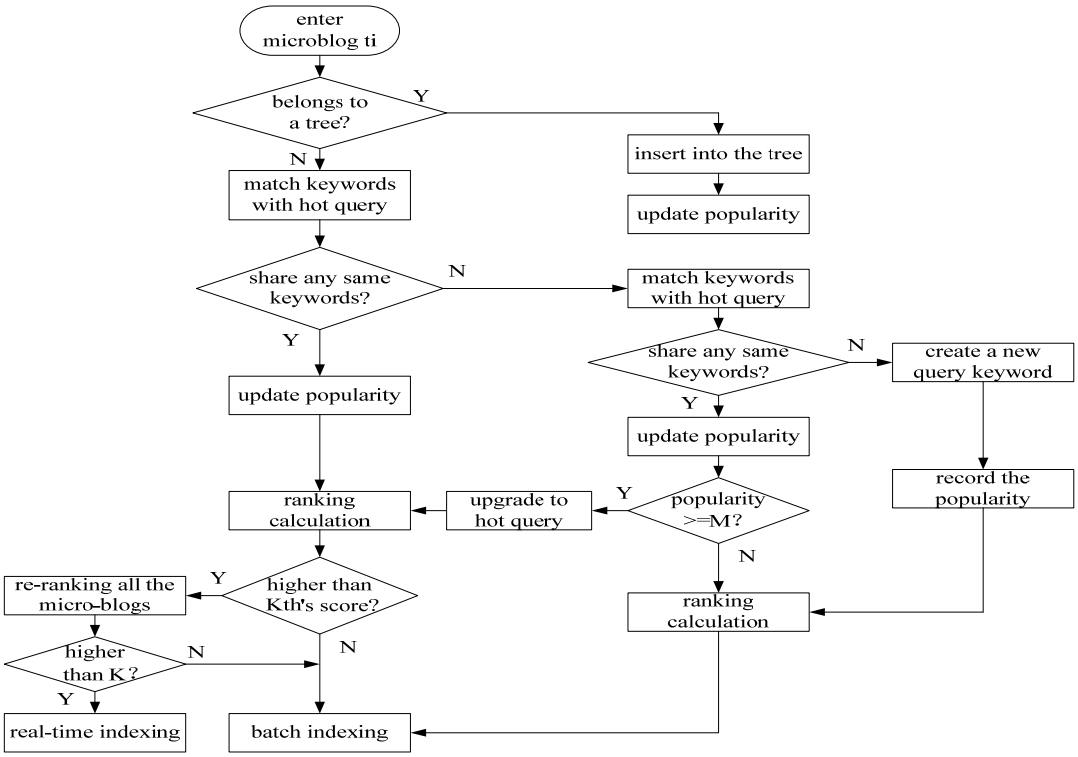

Figure 2. The microblog classification and indexing process

V. CONCLUSION

Classification mechanism and framework of real-time search have been studied in this paper, solving the two key problems faced by real-time search: information filtering and identification, and fast data indexing. Query and microblog classification methods are put forward. Queries are divided into hot query and candidate query according to the number of related microblogs for the query. Microblogs are divided into effective microblog and noise microblog according to its ranking. Effective microblogs of a hot query are for real-time indexing, and all the microblogs of a candidate query and noisy microblogs of a hot query are for batch indexing. Based on the mechanism of meta-search engine and Sina microblog as for example, a framework of microblog real-time search is designed. Taking cooperation microblog systems as the information filtering interface, the framework not only saves the cost of real-time search, but also gets microblog data with high quality.

REFERENCES

[1] Li Yungming, Hsiao Hanwen. Recommender service for social network based application. Proc of the 11th International Conference on Electronic Commerce. New York: ACM Press, 2009; 378—381

[2] http://twitter.com

[3] Chun Chen, Feng Li, Beng Chin Ooi, Sai Wu: TI: an efficient indexing mechanism for real-time search on tweets. SIGMOD Conference 2011: 649-660

[4] Saroop, A. Crawlers for social networks & structural analysis of Twitter..2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA), Page(s): 1 – 8, 2011

[5] Busch, M. Gade, K. ; Larson, B. et al. Earlybird: Real-Time Search at Twitter. 2012 IEEE 28th International Conference on Data Engineering (ICDE), Page(s): 1360 – 1369, 2012

[6] http://blog.sina.com.cn/

[7] iProspect. iprospect search engine user behavior study[EB/OL].www.iprospect.com

[8] M. Stonebraker. The case for partial indexes. SIGMOD Rec., 18(4):4–11, 1989.