Incremental Mining of Significant URLs in Real-Time and Large-Scale Social Streams

Cheng-Ying Liu[1,2], Chi-Yao Tseng[2], and Ming-Syan Chen[1,2]

[1] Dept. of Electronic Engineering National Taiwan University, Taipei, Taiwan, R.O.C.
[2] Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan, R.O.C.
{bermuda,cytseng,mschen}@citi.sinica.edu.tw

Abstract. Sharing URLs has recently emerged as an important way for information exchange in online social networks (OSN). As can be perceived from our investigation toward several social streams, the percentage of messages with URL embedded ranges from 54% to 92%. Due to the extremely high volume of evolving messages in OSN, finding interesting and significant URLs from social streams possesses numerous challenges, such as the real-time need, noisy contents, various URL shortening services, etc. In this paper, we propose the Significant URLs MINing algorithm, abbreviated as SURLMINE, to produce the up-to-date ranking list of significant URLs without any pre-learning process. The key strategy of SURLMINE is to incrementally update the significance coefficients of all collected URLs by four pivotal features, including Follower-Friend ratio, language distribution, topic duration and period and decay model. Moreover, its capability of incremental update enables SURLMINE to achieve the real-time processing. To evaluate the effectiveness and efficiency of SURLMINE, we apply the proposed framework to Twitter platform and conduct experiments for 30 days (over 75 million tweets). The experimental results show that the precision of SURLMINE can reach up to 92%, and the execution performance can also satisfy the real-time requirements in large-scale social streams.

Keywords: Significant URLs mining, incremental scheme, large-scale social streams, real-time processing.

1 Introduction

Due to the exploding popularity of online social networks (OSN) and microblogging platforms, such as Facebook, Twitter, LinkedIn, and Google+, spreading information with URLs has become a general phenomenon in social interactions. According to our observation by randomly sampling 140 million Twitter messages related to "YouTube", as high as 91.8% of messages contain at least one URL. Moreover, other similar experiments also indicate a high frequency of URL attachment in messages with certain keywords, such as 75.8% in "Google", 60.7% in "News", and 54.2% in "Obama". Since the data generation rate on OSN is very high, it is challenging to efficiently deal with real-time social streams. As far as Facebook is concerned, there are more than 900 million users, and in each

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 473–484, 2013.

day, over 3 billion comments and 300 million photos are added and uploaded in this platform[1]. On the other hand, from the report of mediabistro.com[2], over 500 million accounts have registered on Twitter in March 2012, and the number of tweets per day has reached 400 million[3]. It can be perceived that social streams are now a large-scale data warehouse with a great wealth of real-time information [1], such as news, blog articles, interesting facts, comments, and multimedia content. Although there are several existing services (e.g., Twitter Search, Google, and Bing) offering the social streams searching function based on the similarity between text content and query keywords, the results are often not well-organized and thus cannot provide users concise and meaningful information. For example, if a user intends to query social messages about specific breaking news, it is impossible for him/her to explore all related contents which are unstructured and generated rapidly. Furthermore, users may desire to know what time-sensitive news and hot topics are widely discussed at this moment. These functionalities cannot be provided by existing systems and are not yet fully explored in the literature.

In this paper, we focus on mining significant URLs which attract much attention and are highly discussed on OSN. We aim to monitor social streams containing a specific keyword and return the top-k up-to-date ranking list of significant URLs. The motivations of focusing on URLs are as follows. First, with the text length restriction, people are only allowed to write limited characters in a social message. Thus, to provide more complete information and share news with friends, attaching URLs has become a common approach in OSN and micro-blogging platforms [2, 3]. Second, URL is a universal locator that is language-independent, which means that people using different languages may still share identical URL. Third, since URL is a convenient cross-platform linker, it is helpful for celebrities or companies to find out which mediums are often included and highly correlated to them. However, although URL provides many advantages, discovering significant URLs possesses numerous challenges. The first challenge comes from the wide use of URL shortening services, which greatly increases complexity and difficulty when dealing with various kinds of URLs. Second, some URL shortening services have the time limit, which means URL shorteners could be invalid after a period of time. Third, since social streams are dynamic and ever-increasing, designing an efficient and real-time mining scheme for dealing with such large-scale and noisy data is a challenging task.

To the best of our knowledge, there is no existing mechanism which considers all above issues and fully explores the discovery of significant URLs on large-scale and real-time social streams. In this paper, we propose the Significant URLs MINing algorithm (abbreviated as SURLMINE) that incorporates a variety of social features to determine the significance and popularity of URLs. In addition, each post will just be analyzed once so as to reduce computation time and enable real-time processing to users. To verify the effectiveness of the proposed

[1] http://twopcharts.com/twitter500million.php

[2] http://www.mediabistro.com/alltwitter/twitter-active-total-users_b17655

[3] http://techglimpse.com/index.php/facebook-901-million-user-accounts.php

algorithm, we collect more than 75 million tweets in 30 days from May 6th to June 6th in year 2012 with the specific keywords. The experimental results show that the proposed scheme not only is able to extract significant URLs with high precision, but also satisfies real-time need in large-scale social streams. The rest of this paper is outlined as follows. Section 2 reviews the related studies on searching and recommendation techniques in micro-blogging platforms. We elaborate the details of our data crawling scheme and observation in Section 3, while Section 4 introduces the proposed SURLMINE algorithm. Extensive experiments and performance evaluation are presented in Section 5. Finally, concluding remarks are given in Section 6.

2 Related Work

Mining in OSN has been widely discussed in recent years due to its rapid growth. One major advantage of discovering knowledge from OSN is that social messages generally reflect most recent news and topics, and it is hardly to be accomplished by directly applying traditional algorithms, such as PageRank. The reason is that several related algorithms typically suffer the cold start problem, indicating that time-sensitive and relevant contents cannot be discovered in time to provide latest information [4]. Therefore, some researchers aim to extract interesting contents and rank them based on certain query inputs. In [5], Duan et al. used Rank-SVM technique to obtain critical features for selecting the candidate set, and the tweets are ranked according to the relevance of topics. The three pivotal features in [5] are: whether a tweet contains one or more URLs, the length of a tweet (number of characters), and the authority of user accounts. Phelan et al. [6] proposed a novel news recommendation technique called Buzzer, which harnesses real-time Twitter data as the basis for ranking and recommending articles from the collection of RSS feeds. Dong et al. [7] exploited Gradient Boosted Decision Tree algorithm to improve receny ranking from real-time Twitter streams. Another research direction is the burst detection. In [8], Mathioudakis et al. attempted to catch the vocabularies which suddenly appear with an unusually high rate. They also showed that the proposed approach has good performance in extracting trending topics from real-time information streams.

Note that most prior studies have all encountered the challenge of not being able to process large-scale datasets [9, 10] due to the high cost of computation time. In order to reduce the amount of data returned by each query, some researches proposed to narrow down search range by considering a small group of people among friends and the friends of friends [3, 10]. However, the main weaknesses of those methods are that results may be affected by preconception, and the quality of results also highly depends on the user's social communities. Moreover, even if bursty events can be precisely detected, further description and details are still unable to be given just by a set of words. Considering the period of presidential election, the names of president candidates will appear very frequently in social discussion, but further analysis is required to find out what do users talk about and which articles are representative ones. The most important thing is that the information diffusion patterns and behaviors of participants toward various topics have been confirmed as different [3]. Therefore, it is challenging to design a universal mining scheme which covers various topics in such large-scale data streams.

Our work builds on earlier contributions in three key respects. First, due to the diversity of URL shortening services and the time limitation of URL shorteners, most previous works only tracked one format of URL shorteners. For example, in [7] and [11], the authors focus on tinyURL and bit.ly respectively. In this paper, we consider involving all kinds of URL shorteners, and thus more complete information is covered for mining significant URLs. Second, micro-blogging is a constantly evolving medium where users often leave and join, and relationships between users may also change anytime. Thus, in order to obtain more information, our crawling has included a more wide range of global real-time streams that could be spread from any person and in any language, which is much bigger than crawling from user communities. Third, since social stream is a Big Data with high volume of noise and fast data generation rate, in order to be applicable in such context, each post will be regarded as an impact to certain URLs. Even if a post is a spam message, our algorithm is still able to determine the negative influence and incrementally updates the significant coefficients toward certain URL without any prior learning process.

3 Data Crawling and Pre-processing

In this paper, we evaluate our scheme with Twitter platform, since it is the most popular micro-blogging website with more than 140 million active users[4]. After gathering a large amount of Twitter data, URLs can be directly extracted and expanded from shortened forms to original forms. In Section 3.1, we first introduce two common approaches for crawling social messages from Twitter information streams. Next, we explain the procedures of expanding various URL shorteners and provide detailed analysis in Section 3.2.

3.1 Real-Time Crawling

There are two main approaches for collecting Twitter data, which are REST API[5] and streaming API[6]. Although both methods allow developers to access Twitter data, they are still several different properties between them. REST API provides simple interfaces for most Twitter functionality, and up to a maximum of 100 tweets will be returned per query. Twitter also applies a rate limitation to REST API where at most 350 requests are permitted per hour[7]. On the other hand, streaming API requires keeping a permanent HTTP connection open, and it randomly returns tweets containing a specific search keyword with the total

[4] http://blog.twitter.com/2012/03/twitter-turns-six.html

[5] https://dev.twitter.com/docs/api

[6] https://dev.twitter.com/docs/streaming-apis

[7] https://dev.twitter.com/docs/faq#6861

Table 1. The number of tweets crawled by streaming API and REST API

Keyword	Streaming API		REST API		Multiple
	Total	TPS*	Total	TPS	(S/R)*
YouTube	143,869,821	30.28	6,306,355	1.33	22.81
News	41,482,108	8.73	7,906,215	1.66	5.25
Google	28,720,525	6.04	7,474,687	1.57	3.84
Obama	8,503,834	1.79	5,271,187	1.11	1.61

***TPS**: Tweets Per Second ****S/R**: Streaming/REST

quantity never exceeding 1% of all public data streams. Without the overhead and duplication issues caused by polling REST API at endpoint, streaming API is able to crawl a larger number of tweets. Table 1 shows the information of the data quantity by both APIs from May 6th to June 30th (55 days) in year 2012.

3.2 URL Statistics

By employing the above-mentioned crawling mechanism, we are able to determine the percentage of URLs in Twitter data and the proportional distributions of various URL shortening services. As shown in Table 2, among the specified four keywords, more than 54.22% of tweets attach at least one URL, in particular for tweets that mention ”YouTube”, where the URL attachment rate is as high as 90.80%. This statistic indicates that the popularity of URL shorting services may vary with different topics. Moreover, there are still many other services which are not so well-known and are infeasible to enumerate all of them. Therefore, developing a universal expanding method is desirable for covering more complete information.

Table 2. The distribution of various URL shortening services

Keyword	original	bit.ly	tinyurl	ow.ly	goo.gl	others	URL%
YouTube	96.49%	0.95%	0.14%	0.10%	0.12%	2.20%	90.80%
News	37.92%	17.92%	1.10%	0.00%	2.17%	40.89%	75.77%
Google	54.49%	16.30%	0.98%	2.28%	4.12%	21.83%	60.67%
Obama	30.20%	23.33%	2.27%	2.62%	2.87%	38.71%	54.22%

To resolve this difficulty, we devise a subroutine to expand all kinds of URL shorteners to their original forms by recursively tracking their redirections. To enable the real-time processing, this subroutine will be executed immediately whenever a URL is detected. With original URLs expanded by the devised procedure, we are able to calculate the average number of URLs embedded in a tweet and the frequency of URL occurrence more accurately. As can be seen in Figure 1, most tweets tend to attach only one URL, and tweets that contain more than three URLs are less than 0.1%. In addition, Figure 2 shows the cumulative distribution function (CDF) of the frequency of URL occurrence, which

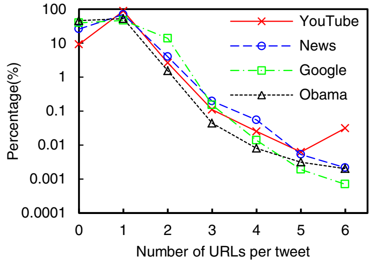

Fig. 1. Distribution of URL quantity in each tweet with different keywords

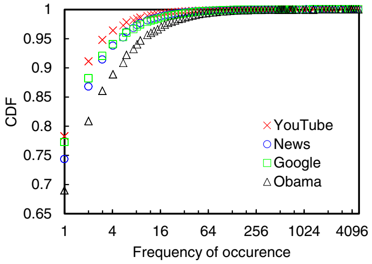

Fig. 2. Frequency of URL occurrence expressed in CDF with different keywords

addresses that only a very small number of URLs are posted frequently, and most URLs are just attached in one message. It is worthy to notice that most frequently attached URLs do not always imply they are significant or popular ones that interest users. It is because in order to gain more focus, spammers usually attempt to continuously post a large number of advertise links or phishing links to defraud audience. Moreover, some ordinary URLs with very high frequency is just owing to their inherent function, such as www.google.com. Table 3 gives instances of top-5 URLs with the most frequency of occurrence in messages containing keywords "Google" and "News", respectively.

Table 3. An example of top-5 most frequent URLs related "Google" and "News"

Keyword	URL
Google	`http://itunes.apple.com/app/rage-of-bahamut/id506944493?mt=8`
	`https://play.google.com/store/apps/details?id=com.ruckygames.gunmaapps`
	`https://play.google.com/store/apps/details?id=com.ruckygames.otherjp`
	`http://www.google.com`
	`http://www.google.com/intl/en/ipv6/`
News	`http://www.billboard.com/bbma/news/justin-bieber-usher-billboard-music-awards-cover-story`
	`http://www.goal.com/`
	`http://mobile.gungho.jp/news/sengoku/root.html`
	`http://www.thedailymash.co.uk/news/international/greeks-apologise-with-huge-horse-20120515`
	`http://news-discussions.com/`

4 Significant URLs Mining

As mentioned previously, the frequency of URL occurrence cannot be directly exploited to guarantee the quality of links due to some malicious operations and inherent characteristics. To better identify the significance of each URL, we devise SURLMINE algorithm to estimate the significance coefficients of URLs by measuring several characteristic features of social messages (i.e. tweets). In Section 4.1, we introduce four pivotal features that are considered in SURLMINE algorithm, and the details of SURLMINE are described in Section 4.2.

4.1 Characteristic Features of Social Messages

There are four characteristic features used to estimate the significance coefficients of URLs. They are (1) Follower-Friend ratio, (2) language distribution ratio, (3) duration and period, and (4) decay model, which are explained as follows.

Follower-Friend Ratio. Nowadays most of OSNs support non-reciprocal relationships to manage the social circles of users. The characteristic of non-reciprocal relationships is that it allows users to add anyone into their circles without their approval. For instance, users on Twitter and Google+ are allowed to directly add celebrities, such as Barack Obama or Lady Gaga, into their social circles without any permission. Thus, we can determine the ratio of followers to friends to quantify the user popularity. Let $\mathcal{S} = \{u_1, u_2, u_3, ...\}$ denote the set of distinct URLs in real-time social streams. For each URL $u_i \in \mathcal{S}$, the Follower-Friend ratio ε_i is defined as follows.

$$\varepsilon_i = \frac{C^i_{follower}}{C^i_{friend}} \tag{1}$$

In above equation, $C^i_{follower}$ and C^i_{friend} respectively represent the number of followers and friends of the author who spreads the URL u_i. As discuss in [12], the ratio ε directly reflect an author's popularity, for a user who has $\varepsilon \geq 2$, it means that he/she is a popular person, and lots of people want to hear what he/she said. Oppositely, if $\varepsilon < 1$, it shows this person is a knowledge seeker but not getting much attention. Therefore, in our scheme, so as to mining those URL that concerned by most of people, if a URL posted by a high Follower-Friend ratio author, the URL will be regarded as more significant.

Language Distribution. In SURLMINE, for a URL u_i, let $l_1, l_2, l_3, ..., l^i_{m_i}$ represent the number of messages used in m_i kinds of languages. We first use m_i for comparing global popularity. Once the m_i is equal to other URLs, the language distribution ratio ξ_i will be determined as follow by using average standard deviation.

$$\xi_i = \frac{\sum_{j=1}^{m_i} |m_i \cdot l^i_j - \sum_{j=1}^{m_i} l^i_j|}{{m_i}^2 \cdot \sum_{j=1}^{m_i} l^i_j} \tag{2}$$

It can be noticed from above equation that if the language distribution of URL u_i is balanced, the value of ξ_i will approach zero. Oppositely, if the distribution is uneven, the value of ξ_i will be much larger. Since each URL definitely links to a web page, if users attach an identical URL with different languages, it can be inferred that this URL probably contains international insights. Moreover, owing to the fact that most spam messages are spread in one particular language, the probability of a spam URL mentioned by several different languages must be very low. In this way, we perceive that language distribution not only reflects global popularity, but also simultaneously contributes to resolve the problem of spam links.

Duration and Period. In our scheme, we use both duration and period to differentiate various URL influences and behaviors. By enforcing the limitation of both period and duration, we can rapidly eliminate outdated URLs and the URLs just discussed in a flash of time. Let $\mathcal{T} = \left\{u_1^i, u_2^i, u_3^i, ..., u_{k_i}^i\right\}$ denote the set of URLs that point to identical web page with u_i in $\mathcal{S}$. For each URL $u_i \in \mathcal{S}$, the duration d_i and period p_i are formulated as follows.

$$p_i = \frac{d_i}{k_i} = \frac{\Sigma_{j=1}^{k_i}(t_j^i - t_{j-1}^i)}{k_i} \tag{3}$$

In Eq. 3, t_j^i represents the timestamp of URL u_j^i in $\mathcal{T}_i$, and k_i is the frequency of occurrence of URL u_i in $\mathcal{S}$ at that time. That is, the duration d_i is the time between t_1^i to t_k^i, and the period p_i is the specific value of duration d_i to the frequency of occurrence k_i.

Decay Model. In addition, a decay function model has been applied to determine the decay ratio λ_k^i for each u in $\mathcal{T}_i$, which is defined as follows.

$$\lambda_k^i = \begin{cases} 1 & , i < \rho \\ 1 - \frac{\rho^2(t_k^i - t_{k-1}^i)^2}{\eta(t_\rho^i - t_1^i)^2} & , \lambda_k^i \geq minDecay \\ minDecay & , otherwise \end{cases} \tag{4}$$

In above equation, ρ is a constant value used to determine the average period of first ρ URLs in $\mathcal{T}_i$, and η is a threshold that determines the multiple of average period for decay cycles. Note that the minimum decay λ_k^i is experimentally set as 0.01 since if a URL is unfrequented for a long time (more than η times as many as the average period), the value of $t_k^i - t_{k-1}^i$ will be large. In such situation, the decay ratio λ_k^i will be very small or negative, and this situation must be prevented. Oppositely, if URL has just been attached, the difference of t_k^i and t_{k-1}^i will be small, and the decay ratio λ_k^i will approach to 1, indicating a weak decay.

4.2 SURLMINE Algorithm

$$\delta_k^i = \prod_{j=1}^{k} \varepsilon_j \cdot \lambda_j^i = \varepsilon_k \cdot \lambda_k^i \cdot \prod_{j=1}^{k-1} \varepsilon_j \cdot \lambda_j^i = \varepsilon_k \cdot \lambda_k^i \cdot \delta_{k-1}^i \tag{5}$$

By considering above-mentioned features, the goal of SURLMINE is to incrementally update the significance coefficient of a URL so as to immediately output the results. Let $\delta_1^i, \delta_2^i, \delta_3^i, ..., \delta_k^i$ denote the significance coefficients for any URL u_i in $\mathcal{T}$. The significance coefficient δ_k^i of URL u_i is formulated in Eq. 5. For any significance coefficient δ_k^i, it only requires the Follower-Friend ratio (i.e. ε) and the decay ratio (i.e. λ_k^i) to determine significance coefficient δ_k^i, where δ_{k-1}^i is the previous state of δ_k^i that is kept in memory.

Algorithm 1. Significant URLs MINing Algorithm (SURLMINE)

```
Input: $u$, A URL extracted from social streams;
Result: $\mathcal{R}$, Update the set of the URL significance coefficients;
Data: $\mathcal{S}$, The set of existing distinct URLs;
        $\mathcal{K}$, The set of the frequency of all URL occurrence;
$\tilde{u}$ = Expand URL $u$;
$\varepsilon$ = Compute Follower-Friend ratio of user $\alpha$ by Eq. 1;
if $\tilde{u} \notin \mathcal{S}$ then /* If $\tilde{u}$ has not been quoted before*/
    Insert URL $\tilde{u}$ to set $\mathcal{S}$;
    Insert the value 1 as the frequency of occurrence of $\tilde{u}$ in $\mathcal{K}$;
    Set significance coefficient $\delta$ of URL $u$ as $\varepsilon$;
else
    $\alpha$ = Get user ID who attaches URL u;
    $\mathcal{A}$ = Get the user list which has attached URL $\tilde{u}$ before;
    if $\alpha \notin \mathcal{A}$ then /* If $\tilde{u}$ is mentioned by a new user */
        Insert user $\alpha$ to set $\mathcal{A}$;
        Add the value 1 to the frequency of occurrence of $\tilde{u}$ in $\mathcal{K}$;
        $\lambda$ = Compute decay ratio of $\tilde{u}$ with its creation time by Eq. 4;
        $\delta$ = Get previous state significance coefficient of URL $\tilde{u}$ from $\mathcal{R}$;
        Set significance coefficient $\delta$ as $\delta \cdot \varepsilon \cdot \lambda$ by Eq. 5;
    end
end
$\xi$ = Compute language distribution ratio of $\tilde{u}$ based on Eq. 2;
Insert both $\delta$ and $\xi$ of URL $\tilde{u}$ to $\mathcal{R}$ in decreasing order;
Compute period $p$ and duration $d$ of $\tilde{u}$ with its creation time by Eq. 3;
if $d < minDuration$ or $p > maxPeriod$ then Set URL $\tilde{u}$ as unavailable;
```

For each incoming URL u, SURLMINE first expands u to the original form $\tilde{u}$, and next identifying whether URL $\tilde{u}$ has been quoted before by examining if URL $\tilde{u}$ is an element of $\mathcal{S}$. If it is not, this implies $\tilde{u}$ is a new URL and the significance coefficient δ of $\mathcal{S}$ will be directly assigned as its Follower-Friend ratio. Otherwise, if it is the first time that author α attaches URL $\tilde{u}$, the significance will be incrementally determined with the previous state of significance coefficient by Eq. 5. Finally, significance coefficient will be inserted into the set $\mathcal{R}$ in decreasing order. When doing insertion, once the significance coefficients are equal, URL that has higher language distribution ratio will be regarded more significant. The two thresholds of *minDuration* and *maxPeriod* are separately set as 1 and 0.5 to filter those immature URL. It indicates even a URL u has a high significance coefficient, the URL can be outputted if and only if duration d_i is larger than *minDuration* and period p_i is less than *maxPeriod*. Overall, with the incremental characteristic, SURLMINE is able to maintain an up-to-date ranking list of significant URLs in the environment of fast-pacing social streams. Moreover, since SURLMINE is unnecessary to scan past data for updating significance coefficients, and thus large storage space and computation time can be saved.

5 Experiments

In this paper, we conduct a series of experiments to verify the effectiveness of SURLMINE on a personal computer with 3.4 GHz CPU and 4 GB main memory. The implementation and experimental design are described in Section 5.1. We then present the performance evaluation of precision and efficiency in Section 5.2.

5.1 Experimental Design

In order to make experiments more extensible and modular, we divide the implementation into two parts. The first part is the data crawling through Twitter streaming API. The major task of this data crawler is maintaining a connection with Twitter servers to continuously access tweets that contain the specified keywords. The second part is responsible for significant URLs mining that outputs up-to-date top-k URLs, where k is defined as 0.01% of URLs collected in that day. In our experiments, we additionally implement Boolean Spreading Activation algorithm and Burst Detection algorithm [8] (abbreviated as BSA and BD respectively) for comparing purpose. The BSA has been widely used in areas such as information retrieval and epidemic models, where its ranking strategy is mainly based on the frequency of occurrence. On the other hand, the BD outputs the URLs that suddenly appear with an unusually high rate by continuously tracking period of URLs. To judge the precision of different algorithms, since the preference for information may differ from person to person, evaluating URL significance through manual study by a specific group of people may be biased. To better solve this problem, we focus on the social streams of Twitter that mention the keyword "YouTube" and attach at least one URL. YouTube is well-known for being the busiest video sharing site, where the total number of views in each day has exceeded 4 billion. Moreover, since each video on YouTube has its own statistics about audience rating, we can validate video significance by considering following conditions (1) more than 500,000 of view counts; (2) more than 1,000 of views per day after 24 hours with present view count growing rate; (3) the ratio of like and dislike is more than 100. If one of the above conditions is satisfied, a video will be identified as a significant video, and the precision is the percentage of significant video URLs.

5.2 Precision and Efficiency Issues

The experiments have been continuously executed for 30 days from May 6th to June 6th in year 2012. During this period, we snapshot our data warehouse at several time points with different durations. As shown in Table 4, after 30 days, there are totally 41 million videos mentioned in 75 million tweets. Moreover, since we only focus on social messages with URL attachment, an additional search term "http" will be automatically appended for any querying topics before crawling. Thus, around 99% of tweets contain at least one valid URL, except some special cases, such as the tweets containing the term "http" but with no URL attached. The results of precision evaluation are shown in Figure 3. It

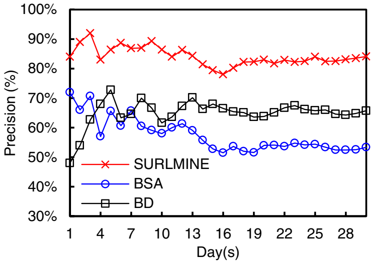

Fig. 3. Day-to-day precision in 30 days from May 6th to June 6th in year 2012

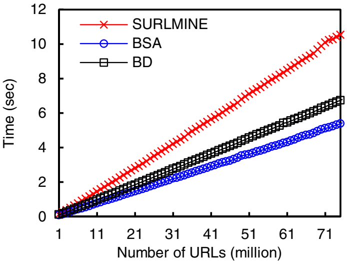

Fig. 4. Cumulative computation time for around 75 million URLs

Table 4. Summary information of dataset with different durations

Duration	Tweet	URL	Video	URL%
6 hours	651,815	645,792	353,119	99.08%
12 hours	1,228,422	1,217,244	681,778	99.09%
1 day	2,782,752	2,754,090	1,508,690	98.97%
7 days	19,830,243	19,637,890	11,020,784	99.03%
15 days	37,506,988	37,195,680	20,505,978	99.17%
30 days	75,500,079	74,865,879	41,408,318	99.16%

can be seen that SURLMINE is more precise than BSA and BD algorithm. On average, the precision can reach up to 92%. By further analyzing the output URLs of these three algorithms, we notice that most advertising and noisy URLs are unable to be excluded by BSA algorithm. In general, these tweets are posted frequently and swiftly accumulate a high frequency of occurrence in a short time. This situation causes that based only on the frequency of occurrence, unwanted URLs always rank high and are hardly to be replaced by new ones. The similar problem has occur in BD algorithm as well. Although BD algorithm has better precision than BSA algorithm, the main weakness of BD algorithm is that it needs more time to become stable. This is because if the frequency of URL occurrence is not large enough for detecting bursty events, the precision of BD algorithm could be lower.

On the other hand, regarding the efficiency issue, SURLMINE only takes about 140 nanoseconds to incrementally include a new URL. Note that the time for the URL expansion step is not covered. Moreover, Figure 4 shows the comparison of cumulative computation time with the number of URLs increasing. It can be seen that the total computation time for analyzing the whole data warehouse (up to 75 million URLs) is only about 11 seconds. Furthermore, although SURLMINE considers more characteristic features and employs more advanced processing, the execution time is not significantly larger than that of BSA, which is solely based on the frequency of occurrence. These evaluations verify that SURLMINE can deal with large-scale and real-time social streams and can be applicable to real applications.

6 Conclusion

In this paper, to enable the real-time processing of significant URLs extraction from OSNs, we proposed an efficient and effective algorithm named SURLMINE. The up-to-date ranking list of significant URLs are produced by incrementally updating the significance coefficients of all collected URLs with four pivotal features, including Follower-Friend ratio, language distribution, topic duration and period and decay model. In our experiments, the collected datasets with over 75 million messages from Twitter cover various kinds of languages, and URLs. With such general settings and such a large quantity of tweets, the precision of SURLMINE can still reach up to 92%, which verifies the effectiveness of the proposed scheme. Moreover, the experimental results also validate that the incremental capability of SURLMINE greatly enhances the efficiency performance. Consequently, these evidences indicate that SURLMINE can be applicable to large-scale and real-time social streams.

References

[1] Kwak, H., Lee, C., Park, H., Moon, S.: What Is Twitter, a Social Network or a News Media? In: 19th ACM International Conference on WWW, pp. 591–600 (2010)

[2] Nagpal, A., Hangal, S., Joyee, R.R., Lam, M.S.: Friends, Romans, Countrymen: Lend Me Your URLs. Using Social Chatter to Personalize Web Search. In: ACM International Conference on CSCW, pp. 461–470 (2012)

[3] Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.: Short and Tweet: Experiments on Recommending Content from Information Streams. In: 28th ACM International Conference on CHI, pp. 1185–1194 (2010)

[4] Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and Metrics for Cold-Start Recommendations. In: 25th ACM International Conference on SIGIR, pp. 253–260 (2002)

[5] Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y.: An Empirical Study on Learning to Rank of Tweets. In: 23rd ACM International Conference on COLING, pp. 295–303 (2010)

[6] Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D.: TwitterStand: News in Tweets. In: 17th ACM International Conference on GIS, pp. 42–51 (2009)

[7] Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., Zha, H.: Time Is of The Essence: Improving Recency Ranking Using Twitter Data. In: 19th ACM International Conference on WWW, pp. 331–340 (2010)

[8] Mathioudakis, M., Koudas, N.: TwitterMonitor: trend detection over the twitter stream. In: ACM International Conference on SIGMOD, pp. 1155–1158 (2010)

[9] Rashid, A.M., Lam, S.K., Karypis, G., Riedl, J.: ClustKNN: A Highly Scalable Hybrid Model- &. Memory-Based CF Algorithm. In: 12th ACM International Conference on WebKDD (2006)

[10] Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering. In: 5th IEEE International Conference on CIT (2002)

[11] Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E.P., Karagiannis, T.: we.b: The Web of Short Urls. In: 20th ACM International Conference on WWW, pp. 715–724 (2011)

[12] Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: The million follower fallacy. In: 4th International AAAI Conference on ICWSM (2010)