블로그에서 포스팅 성향 분석과 갱신 가능성 예측

(Analysis of Posting Preferences and Prediction of Update Probability on Blogs)

이 범 석 † 황 병 연 ††

(Bumsuk Lee) (Byung-Yeon Hwang)

요 약 메타 블로그에 등록된 RSS(Really Simple Syndication)의 수는 수십만 개 또는 수백만 개에 이른다. 따라서 이들에 대한 갱신 확인을 수행하는 것은 상당히 긴 시간과 네트워크 자원을 필요로 한다. 메타 블로그나 블로그 검색엔진은 제한된 자원을 가지고 있기 때문에 하루에 방문할 수 있는 블로그의 수가 제한적이다. 하지만 블로그 검색엔진의 성능향상을 위해 새로운 데이터를 최대한 수집하는 것이 필요하기 때문에, 우리는 이 논문에서 수집 효율을 높이기 위한 새로운 방법을 제안한다. 제안하는 방법은 블로그의 포스팅 성향을 분석하여 그것을 토대로 향후 갱신 가능성에 대해 예측하고 갱신 가능성이 높은 시점에만 갱신 확인을 수행한다. 이 연구는 블로그의 입장에서는 분산 서비스 거부 공격(DDoS Attack: Distributed Denial-of-Service Attack)만큼이나 빈번한 갱신확인을 줄이는데 도움이 되고, 인터넷 전체로 보아서는 트래픽을 감소시키는데 기여할 수 있다.

본 논문에서는 블로거들의 포스팅이 이루어지는 요일과 시간에 특정한 패턴이 존재할 것이라는 가정을 하고, 15119개의 실제 블로그에 작성된 포스트에 대해 요일과 시간의 선호도를 분석하였다. 그리고 과거의 포스팅 이력과 요일에 대한 선호도를 바탕으로 갱신 가능성을 예측하기 위한 방법을 제안하고, 12115개의 실제 블로그에 적용하여 그 정확도를 확인하였다. 성능평가를 통해 약 93.06%의 블로그에서 0.5 이상의 정확도를 가짐을 확인하였다.

키워드 : 블로그, 검색엔진, 웹2.0, 피드기술, RSS

Abstract In this paper, we introduce a novel method to predict next update of blogs. The number of RSS feeds registered on meta-blogs is on the order of several million. Checking for updates is very time consuming and imposes a heavy burden on network resources. Since blog search engine has limited resources, there is a fix number of blogs that it can visit on a day. Nevertheless we need to maximize chances of getting new data, and the proposed method which predicts update probability on blogs could bring better chances for it. Also this work is important to avoid distributed denial-of-service attack for the owners of blogs. Furthermore, for the internet as whole this work is important, too, because our approach could minimize traffic.

In this study, we assumed that there is a specific pattern to when a blogger is actively posting, in terms of days of the week and, more specifically, hours of the day. We analyzed 15,119 blogs to determine a blogger's posting preference. This paper proposes a method to predict the update probability based on a blogger's posting history and preferred days of the week. We applied proposed method to 12,115 blogs to check the precision of our predictions. The evaluation shows that the model has a precision of 0.5 for over 93.06% of the blogs examined.

Key words : blog, search engine, Web 2.0, feed technology, RSS

· 이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단(2009-0074376)의 지원과 2010년 가톨릭대학교 교비연구비의 지원으로 이루어졌음

† 종신회원 : University of Alberta Department of Computing Science Postdoc
bumsuk.lee@ualberta.ca

†† 종신회원 : 가톨릭대학교 컴퓨터공학과 교수
byhwang@catholic.ac.kr
(Corresponding author임)

논문접수 : 2010년 8월 18일
심사완료 : 2010년 9월 10일

정보과학회논문지: 데이타베이스 제37권 제5호(2010.10)

1. 서 론

Web 2.0 technologies have emerged on the Internet over the last several years, and important issues have come to the fore, such as growing social networks and personal media [1]. Social networks, represented by such successful sites as *Facebook.com*, not only connect people directly, but also provide new services that use the network as their platform. Many of these sites offer web services such as file sharing and instant messaging, or even network-connected desktop applications.

Personal media has grown rapidly following the appearance of professional blogging tools. Until a few years ago, the maintenance of a personal webpage required basic knowledge of HTML, but today such easily available websites as *WordPress.com* or *Blogger.com* allow anyone to author a personal blog, and no specialized computer knowledge is required. It is quite easy to create articles on a personal blog and publish them for readers via services such as *Google blog search* or *Technorati.com*. Originally, bloggers discussed their personal lives or lifestyle issues, but today many blogs specialize in professional issues such as the economy or politics. Many people believe that blogs are as influential as the traditional media.

According to Technorati's State of the Blogosphere 2008 report, 94.1 million people, or 50% of the Internet users in the United States, read blogs (May 2008) and 22.6 million people (12%) have their own blog. It is estimated that 77% of active Internet users read blogs [2].

RSS plays a key role in blog services [3,4] and is one of the most successful XML services ever [5]. RSS is a technique for notifying subscribers that new content has been posted: the subscriber does not need to visit the website or blog. RSS readers operate on the user's desktop, or are offered as Web services. Users add the URL of the RSS to the application, which periodically checks for updates and notifies readers.

Currently, new services are emerging that are offering RSS reader tools like a portal service, which is referred to as a meta-blog. A meta-blog gathers RSSs from blogs by operating crawlers or by encouraging people to add RSSs to their own blogs. The meta-blog periodically checks the collected blogs for new contents, and indexes them so that it can respond to user queries [6].

The development of an efficient update manager is urgently required, because the contents of an RSS feed are continuously changing [7,8]. Update checking in a short time interval enables meta-blogs to reflect fast changing issues, thus improving user satisfaction. However it can cause network overhead when the meta-blog checks for updates too frequently. Suppose that the meta-blog checks for updates every ten minutes, or once an hour, and there are millions of blogs. Its network traffic and cost might be huge, and we cannot ignore them. Also for the blog servers, requests for checking update might be worth. Since it has similar aspect with distributed DoS attack, frequent requests in a short time can cause shut down of service. So the meta-blog has to check for updates of each blog at a different time interval in order to reduce overhead on both side between blog servers and meta-blogs. Blog postings have a particular pattern unique to each blogger's activities and we expect that it is possible to predict a blog update by analyzing the days of the week and hours of the day that the blogger actively posts new content. In this research, it takes an average of 2.493 seconds to check for a blog update, and over 10 hours to check 15,119 blogs in a queue. We expect that our approach could reduce network cost by using an update manager, which controls different time intervals for each blogs. We analyze blog postings to determine a blogger's posting habits before designing an appropriate update manager.

The aims of this paper are two-fold: (1) to analyze blog postings in order to determine specific posting patterns, in terms of days of the week and, more specifically, hours of the day and (2) to evaluate a heuristic method that predicts the update probability based on the blogger's posting pattern and history. The main contributions of this paper are the analysis results and evaluation. We found that most bloggers prefer certain days of the week and hours of the day. The proposed update prediction method has a high precision, as demonstrated

by the evaluation of real-world blogs.

Necessity of our research could be explained as follows. In this paper, we introduce a novel method to predict next update of blogs. Since blog search engine has limited resources, there is a fix number of blogs that it can visit on a day. Nevertheless we need to maximize chances of getting new data, and the proposed method could bring better chances for it. Also this work is important to avoid distributed denial-of-service attack for the owners of blogs. Furthermore, for the internet as whole this work is important, too, because our approach could minimize traffic.

The rest of this paper is composed as follows. Section 2 of this paper provides a brief overview of existing studies that analyze the characteristics of RSS feeds and discuss the problem of blog update checking. Section 3 describes the method for analyzing the blog postings that we used to find the preferred days of the week and hours of the day for blog updates. Section 4 reports the results of applying this method to predict the update probability by using real-world data. Finally, Section 5 contains the conclusions and a discussion of future studies.

2. Related works

2.1 Characteristics of an RSS feed

Liu, et al [9] analyzed the client behavior and feed characteristics of RSSs. They collected snapshots of RSS content by actively polling every hour 99,714 feeds listed in the feed directory *syndic8.com*, and used them to analyze updates in terms of update rate and amount of change. There were two notable results; one was that the feed update rates exhibited two extremes: either very frequent or very rare. More than 55% of feeds were updated in the first hour, while 25% of feeds were not updated during the entire polling period. The second result shows the average update time. In total, 57% of the feeds had an average update interval of less than two hours, while 25% of the feeds remained the same over three days. These results indicate that the polling periods for RSS readers should depend on the feeds, and that a meta-blog needs an update manager that checks each blog at a different time interval. In other words, checking each blog for updates every ten minutes or once an hour might be unnecessary.

2.2 Problems of checking for an update to an RSS feed

An RSS feed is continuously updated and meta-blogs need to reflect newly updated contents in their search results. Consequently, meta-blogs check for updates to all of the stored blog RSS feeds at a particular time interval. In practice, Blogpulse.com visits blogs at most once a day, and Allblog.net checks update every hour or every 10 minutes based on blog's update frequency. A shorter interval is better for collecting new content of a critical nature. Rapid collection means that a meta-blog can reflect rapidly changing issues in its search results, thus improving user satisfaction. However, update checking results in unnecessary overheads when it is done too frequently. For example, it is not necessary to check for updates more than once an hour when a blog has only one posting a day. If there were millions of blogs in the checking list, the inefficiency of system resource usage is significant [8]. Furthermore checking update of RSS frequently is more threatening to the server for blog, because it has all the characteristics of a distributed DoS attack. Although the requests are legitimate and small, the sheer number of requests in that short time period can cause shut down their RSS feeds completely due to the increased traffic that they could not handle [7,10,11]. Accordingly, a meta-blog requires an adequate time interval to check for updates.

Every blogger has a unique posting pattern, as explained in Section 2.1. Therefore, a meta-blog can reduce checking overheads by applying an adaptive checking method based on the blogger's update patterns. The meta-blog contains an update-checking module, and some meta-blogs group blogs based on the frequency of updates. Most meta-blogs use a multi-thread type of update-checking module in a distributed environment to reduce overheads. They also provide manual ping service to request checking update and these approaches cannot solve the problem basically. High system overhead sometimes results in delayed responses to

user requests. To resolve this problem, this paper proposes a heuristic method that predicts updated content on a blog.

2.3 Blog Mood Level

Gilad, et al [12,13] analyzed blog posts, and captured global mood levels. For several reasons, they supposed that blogs offer a unique look into people's reactions and feelings towards current events. First, blogs are frequently updated and like other forms of diary are typically closely linked to ongoing events in the blogger's life. Second, blog contents tend to be unmoderated and subjective to a greater extent than mainstream media-expressing opinions, thoughts, and feelings. Finally, the large number of blogs enables aggregation of thousands of opinions expressed every minute; this enables us to perform data abstraction, clean out noise and focus on the main issues.

They also seek algorithms for identifying unusual changes in mood levels and explaining the underlying reasons for these changes. By explanation they mean a short snippet of text that describes the event that caused the unusual mood change. Figure 1 shows an example of a case study. In July, 2005, a peak in "excited" was discovered, where the shaded area indicates the "peak area."

Step 1 of their peak explanation method reveals the following overused terms during the peak period: "potter," "book," "excit," "hbp," "read," "princ," "midnight." Step 2 of their peak explanation method exploits these words to retrieve the following headline from the news collection: "July 16. Harry Potter and the Half-Blood Prince released."

Their method shows that simple techniques based on comparing corpus frequencies coupled with large quantities of data, are effective for identifying the events underlying changes in global moods.

3. Analysis of day and time preferences

In this section, we assumed that each blogger has a unique update pattern. We therefore analyzed the days of the week and the hours of the day that the blogs are most likely to be updated.

3.1 Dataset

We obtained a list of 28,881 RSS feeds by implementing an RSS crawler, and selected 15,119 feeds that conformed to RSS 2.0 specifications. The dataset was gathered over the course of four weeks, but only a two-week dataset with a stable collection was analyzed in the experiment.

Figure 2 shows the language usage statistics of the blog dataset. The dataset contained blogs in

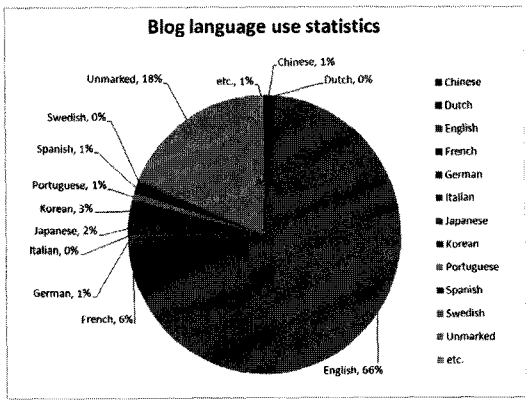

Figure 2. Blog language usage statistics

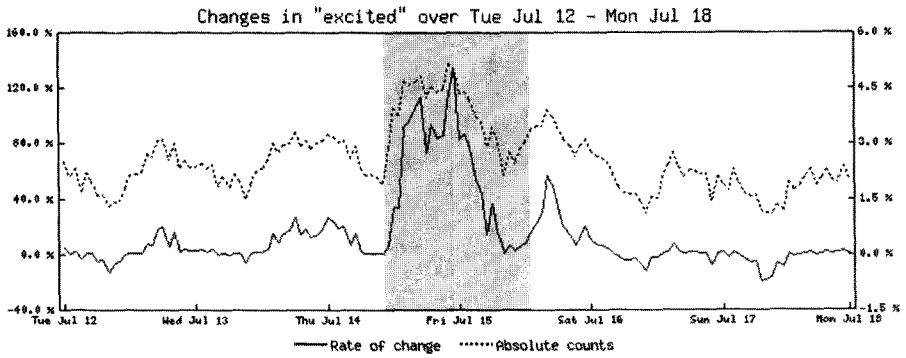

Figure 1. Peak in "excited" around July 16, 2005

English, French, Korean, Japanese and Chinese (66%, 6%, 3%, 2% and 1% of the total, respectively). Language information was not available for 18% of the blogs.

3.2 Preferences: Days of the week and hours of the day

Figures 3 and 4 describe the number of posts in two weeks, and the number of posts in each hour of the day, respectively. Figure 3 shows a very similar graph with a statistics chart of Internet usage according to days of the week [14]. The Monday to Wednesday period had twice the number of posts as the Friday to Sunday period.

Figure 4 shows the number of posts in 24 hours, and the post count corresponds to the daily cycles of modern activities. The number of posts decreases between 1 am and 9 am, and then increases after 9 am, when people are generally using their computers at work. Similarly, it increases by about ten thousand immediately after 2 pm and by about five thousand immediately after 5 pm. This indicates that bloggers tend to write posts either after lunch or after returning home. The increase in the number of posts continues until 1am, when Internet usage peaks. The highest number of daily posts is midnight and 1am.

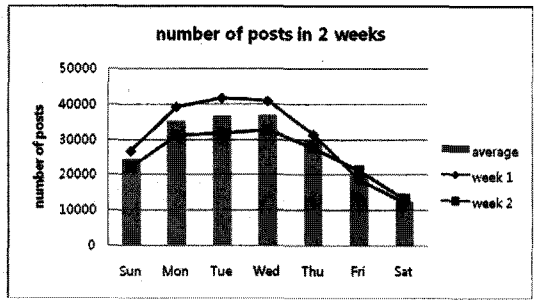

Figure 3. Number of posts in two weeks

An adaptive update manager that searches for updates based on the day of the week and on the time of day might operate effectively in real-world conditions. The application of the proposed method to each blog differs according to its update status, so it is necessary to analyze blogs individually. The rest of this section is dedicated to the analysis of the top-five blogs, based on the number of articles, and four mid-level groups.

Figure 5(a) describes the analysis of the top-five blogs. The average of these five blogs is similar to that indicated by Figure 3, but each blog showed a more explicit preference for certain days of the week. We composed four mid-level groups based on the number of posts, and selected five blogs in each group: blogs with over 100 posts (post-100), and blogs with 50 posts (post-50), 20 posts (post-20) and 10 posts (post-10). The analysis results are shown in Figure 5(b). These groups also show a preference for certain days of the week, especially a blog of the post-100 group that had a strong preference for Sunday. This RSS feed was a pod cast of a German Internet broadcasting service.

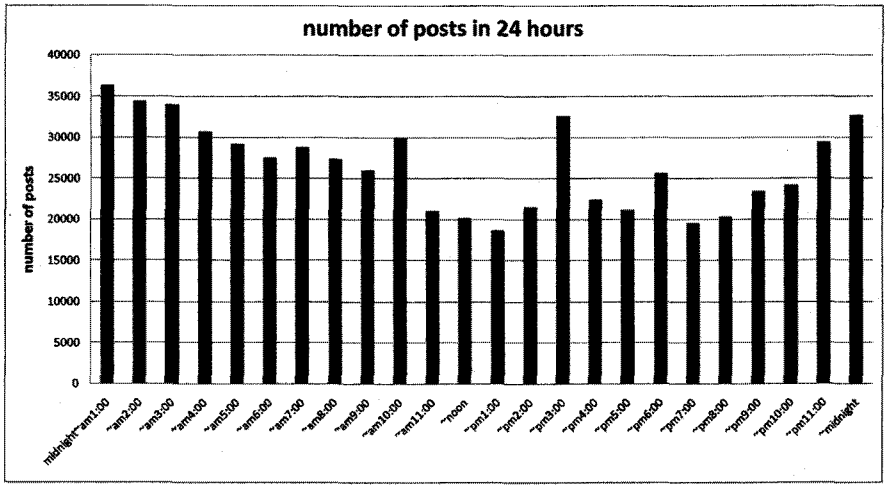

Figure 4. Number of posts in 24 hours

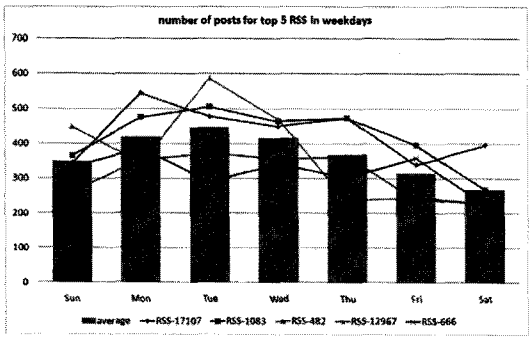

Figure 5(a). Number of posts for top-5 blogs in weekdays

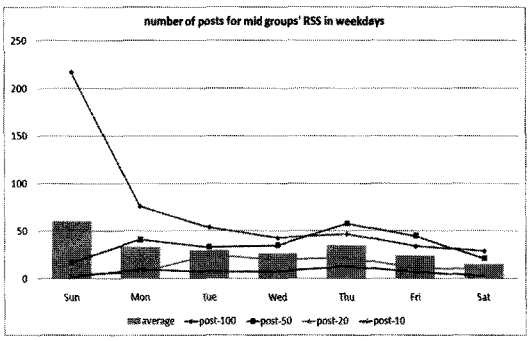

Figure 5(b). Number of posts for mid-level blogs in weekdays

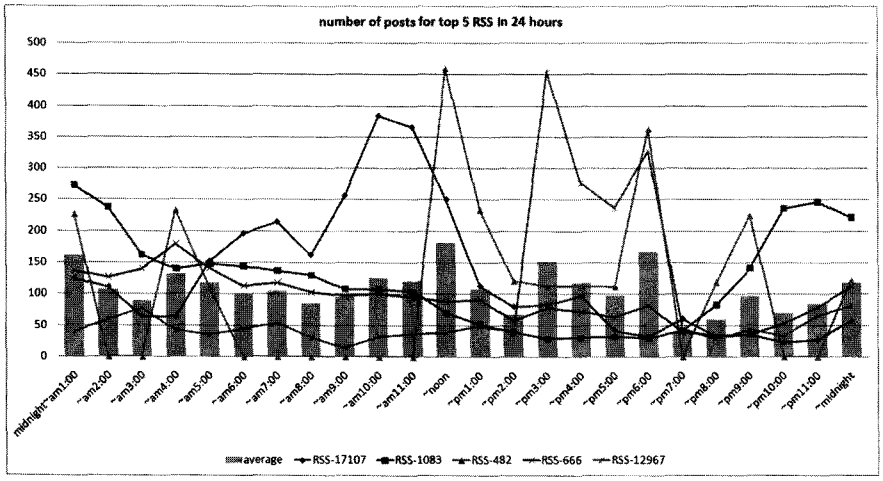

Figure 6(a). Number of posts for top-5 blogs according to time

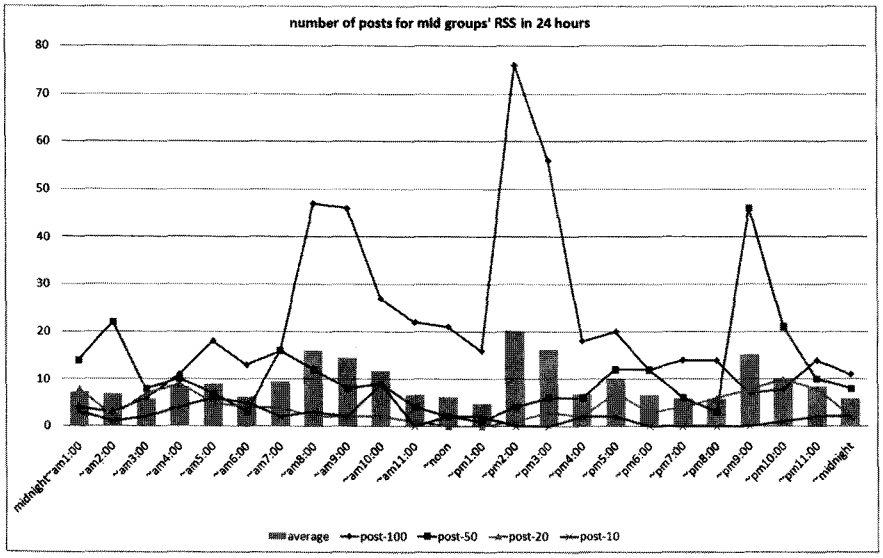

Figure 6(b). Number of posts for mid-level blogs according to time

Figures 6(a) and 6(b) show the number of postings in 24 hours for the aforementioned blogs. Each blog had a strong time preference that was higher than average. RSS-482 in Figure 6(a), which included movie information, showed a highly explicit preference: there were no postings at all between 6 am and 10 am. Figure 6(b) shows that the preferences for the time of day in the mid-level group were also specific.

4. Update prediction and evaluation

4.1 Update prediction

This paper proposes a method to predict the update probability for the following day by analyzing the gathered RSS feeds. We define the update probability *P*.

Definition 1. Update probability P is calculated by using the statistics about the days of the week and the posting history. The weight of two factors is regulated by λ.

Definition 1 can be represented by the following equation:

$$P = \lambda\left(\frac{weekdays_{updated}}{weekdys_{whole}}\right)+(1-\lambda)\left(\frac{days_{updated}}{days_{whole}}\right).$$

The update checking module operates when P exceeds the threshold Θ, which implies that there is a strong probability of an update on that day. Finding suitable λ and Θ values is necessary to improve the precision of the prediction. Suppose the weight factor λ = 0.9 and the threshold Θ = 0.5.

4.2 Data selection

The evaluation was designed to measure the prediction precision. Prediction requires more than one week of data, because the probability can only be calculated based on the posting history. The selected blogs had an interval of more than seven days between the oldest and newest data. We excluded blogs that had more than four articles per day on average. Predicting blogs with such a high update rate might not be needed, because it is likely that they will be updated each day regardless. In total, 12,115 blogs were selected for evaluation based on the aforementioned conditions.

4.3 Data refinement

Figure 7 shows the evaluated data. The left and

raw data		
ridx	pubDate	dayofweek
12	2008-11-22 10:03:19	Sat
12	2008-11-23 11:04:55	Sun
12	2008-11-24 15:16:10	Mon
12	2008-11-25 10:37:30	Tue
12	2008-11-25 14:48:04	Tue
12	2008-11-26 13:03:05	Wed
12	2008-11-27 00:11:37	Thu
12	2008-11-27 13:35:42	Thu
12	2008-11-29 16:44:50	Sat
12	2008-12-02 20:22:40	Tue
12	2008-12-04 18:14:36	Thu
12	2008-12-04 21:29:42	Thu
12	2008-12-06 14:43:28	Sat
12	2008-12-06 16:55:12	Sat
12	2008-12-08 14:59:49	Mon
12	2008-12-10 01:02:39	Wed
12	2008-12-10 11:16:07	Wed
12	2008-12-11 13:52:21	Thu
12	2008-12-11 16:52:26	Thu
12	2008-12-12 01:12:39	Fri
12	2008-12-12 10:21:54	Fri
12	2008-12-12 18:48:57	Fri
12	2008-12-13 15:11:42	Sat
12	2008-12-13 16:30:22	Sat
12	2008-12-13 17:33:08	Sat
12	2008-12-15 10:36:31	Mon
12	2008-12-15 17:35:47	Mon
12	2008-12-16 17:08:25	Tue

after refinement		
calendar	dayofweek	existance
2008-11-22	Sat	1
2008-11-23	Sun	1
2008-11-24	Mon	1
2008-11-25	Tue	1
2008-11-26	Wed	1
2008-11-27	Thu	1
2008-11-28	Fri	0
2008-11-29	Sat	1
2008-11-30	Sun	0
2008-12-01	Mon	0
2008-12-02	Tue	1
2008-12-03	Wed	0
2008-12-04	Thu	1
2008-12-05	Fri	0
2008-12-06	Sat	1
2008-12-07	Sun	0
2008-12-08	Mon	1
2008-12-09	Tue	0
2008-12-10	Wed	1
2008-12-11	Thu	1
2008-12-12	Fri	1
2008-12-13	Sat	1
2008-12-14	Sun	0
2008-12-15	Mon	1
2008-12-16	Tue	1

Figure 7. Example data refinement

right table in the figure shows the raw sample data in the database and the data after refinement, respectively. Data are represented by either a 1 or a 0.

It is possible to achieve a prediction precision similar to the example shown in Figure 8. The table describes the predicted update probabilities versus the real updates. The red cells indicate an incorrect prediction, while the value in parentheses is the real update. The precision measurements started on November 29th in this example.

Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.
1	1	1	1	1	1	0
1	1(0)	1(0)	1	1(0)	1	0
1	1(0)	1	1(0)	1	1	0(1)
1	0	1	1		λ=0.9	Θ=0.5

Figure 8. Result table of precision measurements

For example, the update probability P on December 7th was 0.517, as shown below, which exceeds the threshold. The update-checking module was operating, but there was no update on that day (i.e., this was an incorrect prediction).

$$P = 0.9\left(\frac{1}{2}\right)+(1-0.9)\left(\frac{10}{15}\right)\approx 0.517.$$

The update probability P for the next day was 0.5125, which also exceeded the threshold. There was an updated article on that day (i.e., this was a correct prediction).

$$P = 0.9\left(\frac{1}{2}\right) + (1-0.9)\left(\frac{10}{16}\right) = 0.5125$$

The precision is defined as follows:

Definition 2. Precision is the ratio of days there was a correct prediction to total predicted days. i.e., Precision = the number of correct predictions/the number of predicted days.

If the predictions are correct over the entire period, then the precision is one, whereas the precision is zero if there are no correct predictions. The example in Figure 8 has a precision of 0.66; there were twelve days of correct predictions out of a total of eighteen days.

The precision was calculated for each of the 12,115 blogs and the average precision was 0.76. Figures 9(a) and 9(b) show the blog distribution chart for the precision calculations and the precision distribution chart for the number of posts, respectively. Most of the studied blogs had a precision between 0.8 and 0.9, as shown in Figure 9(a). About 93.06% of the blogs had a precision above 0.5. Figure 9(b) shows that most precision values are distributed between 10 to 20 posts. This is reasonable because the dataset was collected over a three-week period.

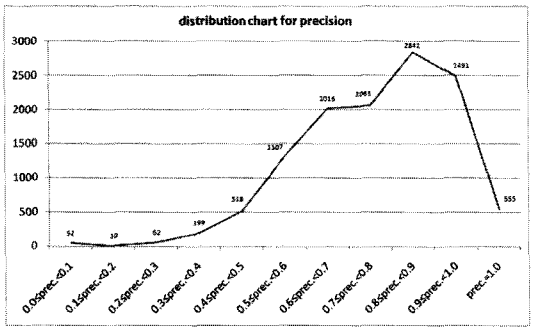

Figure 9(a). Distribution chart for precision

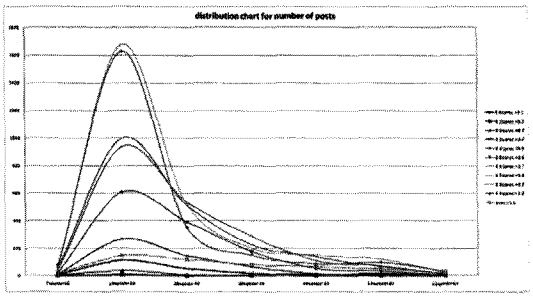

Figure 9(b). Distribution chart for number of posts

4.4 Accumulative update checking time

In this section, we compared the accumulative update checking times in order to verify the efficiency of the proposed update manager. To do this we randomly chose 1,000 blogs. The existing method visits blogs in a queue in consecutive order and takes 6,832 seconds (113 minutes and 52 seconds). On the other hand, the proposed method visits blogs which are predicted to be updated on that day by the update probability equation. Figure 10 depicts the accumulative update checking time for two methods. The proposed method takes less time than the existing method and the gap between the two methods increases gradually. The result implies that the proposed method can help reduce network overhead for checking updates.

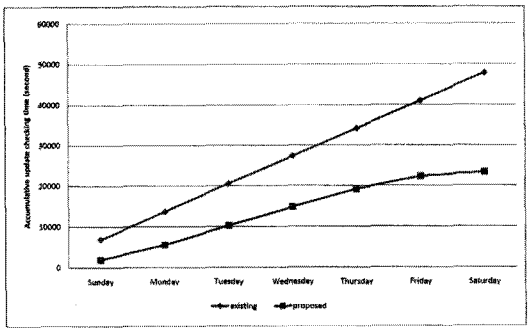

Figure 10. Accumulative update checking time

5. Conclusion

The popularity of personal media, as evidenced by blogs, has resulted in new services such as meta-blogs. A meta-blog collects RSS or ATOM feeds from blogs on the Internet and offers a searching service by indexing content that is included in the feeds. RSS feeds are updated continuously. The number of RSS feeds gathered on meta-blogs is on the order of millions. Accordingly, checking for updates is too time-consuming and uses too many network resources. Most meta-blogs use a multi-thread update-checking module in a distributed environment, and some group blogs based on the update frequency, to provide different checking-time intervals. But for blogs updated only once or twice a week, static update checking intervals cause unnecessary system costs. On the other hand, frequent checking

update could threaten stability of blog server. So our method is proposed to solve these problems.

This paper analyzed postings from 15,119 blogs to determine the preferred days of the week and hours of the day for content updates. Additionally, we proposed a method to predict updates, and we evaluated the precision of this method by using real-world blogs. The average prediction precision for the 12,115 blogs was 0.76, and 93.06% of the blogs had a precision above 0.5. The analysis result shows that there is a specific pattern on blog postings which is linked to bloggers' life styles. Our performance evaluation proved that our approach helps reduce network overhead for update checking. The result implies that the proposed method has the appropriate precision for real world applications, and also can help reduce network overhead for update checking. To predict update probability, our approach takes very short time for access database, and it can be ignored.

The prediction method should be applied over a longer period of time than in this paper. There should be studies on suitable λ and Θ values, to improve the prediction precision. This paper proposed a method to perform update prediction over a period of days and we expect that refinements will result in improved efficiency.

References

[1] T. O'reilly, "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," *Communications & Strategies*, no.65, pp.17-37, 1st Quarter, 2007.

[2] Technorati, "State of the Blogosphere," http://www.technorati.com/blogging/state-of-the-blogosphere/, 2008.

[3] http://en.wikipedia.org/wiki/RSS, 2008.

[4] Berkman Center, "RSS 2.0 at Harvard Law," http://cyber.law.harvard.edu/rss/index.html, 2008.

[5] M. Olson and U. Oqbuji, "The Python Web services developer: RSS for Python," http://www.ibm.com/developerworks/webservices/library/ws-pyth11.html, November 2002.

[6] X. Li, J. Yan, Z. Deng, L. Ji, W. Fan, B. Zhang, and Z. Chen, "A novel clustering-based RSS aggregator," In *Proc. of 16th Int'l Conf. on WWW*, pp.1309-1310, 2007.

[7] K. C. Sia, J. Cho, and H. K. Cho, "Efficient Monitoring Algorithm for Fast News Alerts," Knowledge and Data Engineering, *IEEE Transactions* on vol.19, Issue 7, pp.950-961, July 2007.

[8] B. Lee, J. W. Im, B. Hwang, D. Zhang, "Design of An RSS Crawler with Adaptive Revisit Manager," In *Proc. of the 20th Int'l Conf. on Software Engineering and Knowledge Engineering*, pp.219-222, July 2008.

[9] H. Liu, V. Ramasubramanian, and E. G. Sirer, "Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews," In *Proc. of the ACM Internet Measurement Conference*, 2005.

[10] Chad Dickerson, "RSS Growing Pains," http://www.inforworld.com/d/developer-world/rss-growing-pains.647, 2004.

[11] Paul Festa, "Microsoft flip-flop may signal blog clog," http://news.cnet.com/Microsoft-flip-flop-may-signal-blog-clog/2100-1032_3-5368454.html, 2004.

[12] K. Balog, G. Mishne, and M. d. Rijke, "Why Are They Excited? Identifying and Explaining Spikes in Blog Mood Levels," In *Proc. of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, April, 2006.

[13] G. Mishne and M. d. Rijke, "Capturing Global Mood Levels using Blog Posts," In *Proc. of the AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006)*, pp.145-152, March 2006.

[14] http://www.thecounter.com, 2008.

이 범 석

2004년 가톨릭대학교 분자생물학과(학사). 2006년 가톨릭대학교 컴퓨터공학과(석사). 2010년 가톨릭대학교 컴퓨터공학과(박사). 2010년~현재 University of Alberta 컴퓨터과학과 박사후연구원. 관심분야는 소셜네트워크분석, 정보검색, 데이터베이스, XML, 웹2.0

황 병 연

1986년 서울대학교 컴퓨터공학과(학사) 1989년 한국과학기술원 전산학과(석사) 1994년 한국과학기술원 전산학과(박사) 1994년~현재 가톨릭대학교 컴퓨터정보공학부 교수. 1999년 University of Minnesota 방문교수. 2007년 California State University 방문교수. 관심분야는 XML 데이터베이스, 데이터마이닝, 정보검색, 지리정보시스템