

Automatic Collection and Monitoring of Japanese Weblogs

Tomoyuki NANNO

Interdisciplinary Graduate School of Science and
Engineering
Tokyo Institute of Technology
4259 Nagatsuta-cho Midori Yokohama 226-8503
Japan
nanno@lr.pi.titech.ac.jp

Toshiaki FUJIKI

Interdisciplinary Graduate School of Science and
Engineering
Tokyo Institute of Technology
4259 Nagatsuta-cho Midori Yokohama 226-8503
Japan
fujiki@lr.pi.titech.ac.jp

Yasuhiro SUZUKI

Interdisciplinary Graduate School of Science and
Engineering
Tokyo Institute of Technology
4259 Nagatsuta-cho Midori Yokohama 226-8503
Japan
yasu@lr.pi.titech.ac.jp

Manabu OKUMURA

Precision and Intelligence Laboratory
Tokyo Institute of Technology
4259 Nagatsuta-cho Midori Yokohama 226-8503
Japan
oku@pi.titech.ac.jp

ABSTRACT

We present a system that tries to automatically collect and monitor Japanese blog collections that include not only ones made with blog software but also ones written as normal web pages. Our approach is based on extraction of date expressions and analysis of HTML documents.

Categories and Subject Descriptors

H.5.4 [Information Systems]: Hypertext/Hypermedia; H.3.5 [Information Systems]: Online Information Services

General Terms

Management

Keywords

weblog, blog, document analysis, monitoring

1. INTRODUCTION

The reputation of companies and products is now disseminated very quickly on the WWW, because everyone can send a message to the world easily and actively. Therefore, it is highly required to effectively utilize such a vast amount of information disseminated from many people around the world, for the purpose of finding out the reputation of a company or a product, and quickly coping with people's opinions as a kind of risk management, etc. An information source that has attracted much attention for these reasons is the BBS (Bulletin Board System), and there has been research on monitoring BBSs and extracting and/or mining useful information from them so that people's opinions could be reflected in companies' product development and other activities[8].

Copyright is held by the author/owner(s).
WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-844-X/04/0005.

Similarly, weblogs (blogs) are now thought of as a potentially useful information source. Although the definition of blogs is not necessarily definite, it is generally understood that they are personal web pages authored by a single individual and made up of a sequence of dated entries of the author's thoughts, that are arranged chronologically. Blogs tend to be frequently updated and include links to others' blogs. The content and purposes of blogs vary greatly, from links and commentary about other web sites, to news about a company/person, to diaries, photos, etc.¹.

It is said that blogs date back to 1996, but they exploded in popularity during 1999 with the emergence of blogger (<http://www.blogger.com/>) and other easy-to-use publishing tools[4]. Most recently in 2002, a Newsweek article appeared estimating the number of weblogs to be half a million[4].

The explosive growth of blogspace has led to the study of the space of weblogs becoming a hot topic[4]. There are some sites that provide blog search services, by collecting and monitoring blogs, such as Technorati², Blogdex³, and Daypop⁴.⁵ These services are all based on the RSS (RDF Site Summary) and/or the ping server for collecting and monitoring blogs. RSS is an extensible metadata description and syndication format, currently used for a number of applications, including news and other headline syndication, weblog syndication, etc.⁶. The ping server is a mechanism to tell that a weblog has changed and is used for tracking updates to weblogs[9]. Therefore, collecting a list of blogs and monitoring them are considered to be relatively easy.

In Japan, however, since long before blog software became

¹<http://new.blogger.com/>

²<http://www.technorati.com/>

³<http://blogdex.media.mit.edu/>

⁴<http://www.daypop.com/>

⁵At <http://www.aripaparo.com/archive/000632.html>, the reader can find a useful list of blog search engines.

⁶<http://www.oasis-open.org/cover/rss.html>

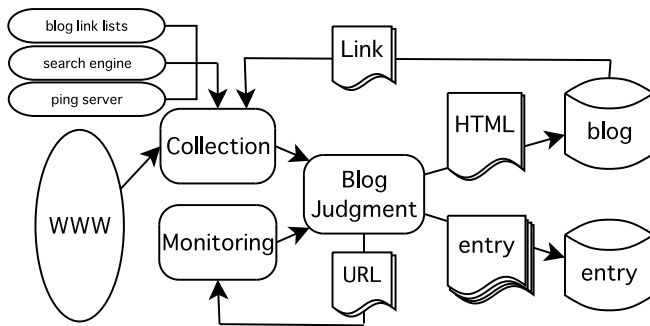


Figure 1: System Architecture

available, people have written ‘diaries’ on the web (called “web diaries”). These web diaries are quite similar to blogs in their content, and people still write them without any blog software. As we will show, hand-edited blogs are quite numerous in Japan, though most people now think of blogs as pages usually published using one of the variants of public-domain blog software, such as ‘Movable Type’⁷. Therefore, it is quite difficult to exhaustively collect Japanese blogs, i.e., collect blogs made with blog software and web diaries written as normal web pages.

With this as the motivation for our work, we present a system that tries to automatically collect and monitor Japanese blog collections that include not only ones made with blog software but also ones written as normal web pages. Our approach is based on extraction of date expressions and analysis of HTML documents, to avoid having to depend on specific blog software, RSS, or the ping server. Furthermore, our system also extracts and mines useful information from the collected blog pages. For more details on the mining module of our system, please refer to [5, 2].

We explain our system architecture in the next section. In Section 3, we explain our approach to judge whether a web page is a blog page or not. We present performance of our blog judgment techniques based on analysis of HTML documents in Section 4 and conclude with some observations in Section 5.

2. ARCHITECTURE OF OUR SYSTEM

The architecture of our system is shown in Figure 1.

The ‘Collection’ module tries to collect candidate blog pages from the WWW. This is done 1) by crawling the WWW, 2) by using the link lists of blog sites and the information from the ping server, and 3) from the links of the pages judged as blogs in the judgment module.

The ‘Monitoring’ module tries to periodically watch the blog pages that the judgment module selected, and extract the updated entries in the pages.

The collected blog entries can be retrieved by using a text search engine called GETA[3]. The entries can be ranked by the number of inbound links, interval of updates, freshness (recency), and their size.

In the next section, we explain the ‘Blog Judgment’ module in detail.

3. BLOG JUDGMENT

The ‘Blog Judgment’ module tries to select only the blog pages in the collection, by using the characteristics of the blog pages. A page is judged to be a blog if and only if a sequence of entries that are articles for a day can be extracted from the page.

The entries should satisfy the following constraints:

1. Entries should contain a date expression, and it should be at the top of the entry.
2. The date expressions of the sequence of entries should be consistently formatted⁸, and be arranged in ascending/descending order.
3. The tag sequence should be uniform for all the entries in the sequence.

The steps of the module are roughly as follows:

1. Extraction of date expressions,
2. Extraction of a sequence of dated entries,
3. Filtering non-blog pages.

3.1 Preprocessing

First, our system applies HTML Tidy[7] to a HTML document in order to translate it into a well-formed XML document. This is done to ensure that the begin tags and end tags are well-balanced in the HTML document. Therefore, in subsequent processes, the constraint that the tag sequence should be uniform for all the entries in the sequence can be used even when the page contains some errors in usages of HTML tags.

3.2 Extraction of Date Expressions

Next, our system tries to find date expressions in the pre-processed web pages automatically. First, it finds all the candidates of the date expressions in the web pages. These candidates include not only the date expressions with complete ‘year,’ ‘month,’ and ‘day’ information, but also the date expressions in which some part is missing.

The date expressions have a variety of formatting. For example, hyphen, slash, space etc. are used as delimiters, and ‘4’ or ‘Apr.’ are often used to express ‘April.’ Though this is a problem peculiar to Japanese date expressions, there are some cases in which ‘year’ is expressed with an era name. (For example, ‘Heisei 16 nen(year)’ is 2004.) Sometimes an era name is expressed with the first letter of its romanized expression. (For example, ‘H16’ means ‘Heisei 16 nen.’) Furthermore, era names are often omitted.

To capture these varieties of formatting, we classified these date expressions into 23 classes, and prepared the manually written regular expressions for each class of the date expression. Some of the date expressions that matched our regular expressions are shown in Table 1, where ‘nen,’ ‘gatsu,’ and ‘nichi’ in Japanese mean ‘year,’ ‘month,’ and ‘day’ respectively.

Because the following date expressions never express the date of entries, our system excludes these expressions from the candidates.

- date expressions which express a time period (ex. 17-22 May 2004.)

⁸‘2003/1/2’ and ‘2-Jan-2003’ are considered inconsistent.

⁷<http://www.movabletype.org/>

Table 1: Example of formatting of date expressions

2004 nen 3 gatsu 5 nichi	2004. 3. 5	2004/3/5
2004-3-5	2004 03 05	3 gatsu 5 nichi
March 5	5 Mar. 2004	5 March 2004.
5-March-2004	March 5 2004	3. 5 2004

- date expressions which are inside a sentence (ex. Tutorial proposals are due November 17th, 2003.)

3.3 Complementing Missing Information

In blog pages, it is not rare that ‘year’ information is located at the top of the page, and does not appear in the date expressions of each entry. Therefore, if the date expressions lack year and/or month information, it is necessary to complement them.

Our system tries to complement the missing information by using heuristics. According to reliability of these rules, they are applied to the candidates of the date expressions in the following order.

1. complementing with the nearest date expression which satisfies the following two restrictions:
 - (a) it must appear before the date expression to be complemented
 - (b) it must be located at the same level as the date expression to be complement, or at a level closer to the root node in the DOM tree
2. complementing with the nearest (in the depth-first search) date expression which appears before the date expression to be complement
3. complementing with the date of ‘last-modified’ as provided by a web server
4. complementing with the date when the page was crawled (Note that this rule is used only for complementing the ‘year’ information of the date expression.)

Moreover, the ‘year’ is often expressed with two digits. In this case, two following ways of interpretation are possible in Japanese blogs: In the first interpretation, the upper 2 digits (like ‘19’ or ‘20’) are omitted. (For example, ‘98’ often means 1998.) In the second interpretation, the Japanese era name is omitted. (For example, ‘16’ often means ‘Heisei 16 nen.’)

To complement such date expressions, our system uses the following two heuristics:

1. When ‘year’ is less than 64, it can be considered that an era name is missing, because, with 64 years, Showa is the longest era as of 2004. Therefore, if the date expression with an era name appears before such date expressions, our system judges that an era name has been omitted, and converts it to Gregorian calendar years. However, because such date expressions can also be interpreted as that the upper 2 digits are omitted, our system does not apply this rule in cases where the converted year is more than 10 years away from the date expression just before it.

2. If there are no date expressions with an era name before the date expression expressed with two digits, our system considers that the upper 2 digits have been omitted. Therefore, our system converts it to Gregorian calendar years by using the following heuristics:

- When “year” is less than 20, our system adds 2000 to it.
- When “year” is greater than or equal to 20, our system adds 1900 to it.

After applying the above complementation process, our system annotates all of the candidates of date expressions which have all the information of ‘year,’ ‘month,’ and ‘day’ with `<date>` and `</date>` tags.

3.4 Extraction of a Sequence of Dated Entries

A page is judged to be a blog if and only if a sequence of entries that are articles for a day can be extracted from the page. To extract a sequence of entries, our system uses the date expressions annotated in the previous step.

Our system first classifies all the date expressions found in the page. The date expressions which satisfy all of the following constraints are classified into the same class.

- all the date expressions should have a uniform sequence to the root node in the DOM tree
- all the date expressions should have a consistent format (ex. ‘2004/1/2’ and ‘1/3’ are considered consistent. ‘2003/1/2’ and ‘2-Jan-2003’ are considered inconsistent.)
- all the date expressions should have the same distance to the previous or next HTML tag (ex. ‘Winter fest - 01.21.2004 -’ and ‘Atom API Support in TypePad - 01.27.2004 -’ have the same distance between the date expression and the next HTML tag; the distance is 2.)

Note that each date expression can belong to multiple classes because of the third constraint.

A page can contain multiple sequences, each of which seems to be a sequence of dated entries. Therefore, our system tries to extract a sequence of entries for each class of the date expressions, because the date expressions which belong to the same class are shown similarly when they are rendered by a browser.

However, a blog page usually contains only a true sequence of entries. Therefore, it is necessary for our system only to select the most reasonable sequence of dated entries. Our system considers that a reasonable sequence for the blog page is one whose total size is the largest of the candidate sequences of entries which are not filtered by the filtering conditions for non-blog pages (described later). If our system cannot find such a sequence of entries, it judges that the page is not a blog.

3.4.1 Determination of the Start Position of Entries

Here we explain the method to determine the start position of each entry for the date expressions which belong to the same class.

Our system first lists XPath expressions for the date expressions which belong to the same class. Comparing these

XPath expressions, our system tries to find the first tag whose position is different between them.

For example, consider the following expressions:

- (1) `/body(0)/div(3)/table(279)/tr(280)/td(281)/date(283)`,
- (2) `/body(0)/div(3)/table(350)/tr(351)/td(352)/date(354)`,

where the number within parentheses shows the position in the HTML document.

In the example, the positions of the ‘table’ tag next to ‘div(3)’ are different between them. This shows that only one date expression is contained in each segment if we segment this HTML document at these table tags. In other words, these table tags show the start positions of entries, because we assume that the date expression for an entry always exists at the top of the entry.

However, when the following date expression also belongs to the same class, as with the above two examples, the ‘tr’ tag next to ‘table’ will be the start positions, because the date expressions (2) and (3) belong to the same segment if this HTML document is segmented by the table tags.

- (3) `/body(0)/div(3)/table(350)/tr(359)/td(360)/date(362)`

Note that the tags which give the start position exist at any time, because every date expressions are annotated by the `<date>` and `</date>` tags and the date tag can be the start position even if all the tags before it are the same.

3.4.2 Determination of the End Position of Entries

Next, we explain the method to determine the end position of each entry.

To extract entries correctly, the pattern in the HTML document should be analyzed, as mentioned in [6]. However, such an analysis of the repetition pattern is costly. Therefore, we assume that the date expression of an entry always appear at the top of the entry, and suppose that the end of the entry is just before the start position of the next entry.

Based on these assumptions, candidates of the end position of an entry are determined as follows:

- (1) the position just before the start position of the next entry,
- (2) the position just before the tag that is closer to the root node than the tag which gives the start position in the DOM tree.

However, for the last entry, the following condition is needed because the above (1) cannot be used for it.

- (3) the position just before the tag on the same level in the DOM tree as the tag giving the start position that never appears in the entries before it.

Our system determines the true end position by selecting the nearest candidate end position from the start position of the entry.

Figure 2 shows an example of extracting entries. Each of the ‘h1’ tag shows the start position of entries.

The first entry dated with ‘date 1’ includes ‘text 1’ in the figure because the end position of this entry is determined by rule (1). (The start position of the second entry is the ‘h1’ tag which annotates ‘date 2’.) The second entry dated by ‘date 2’ includes ‘text 2’ and ‘text 3’. However, this entry does not include ‘text 4’ because the end position of

this entry is determined by rule (2). (The ‘div’ tag which annotates ‘text 4’ is closer to the root node than the ‘h1’ tag which gives the start position of this entry in the DOM tree.) The last entry dated with ‘date 3’ includes ‘text 5’. However this entry does not include ‘text 6’ because the end position of this entry is determined by rule (3). (The previous two entries do not include ‘hr’ tags.)

By these processes, the start position and end position of an entry can be determined. Therefore, given the date expressions which belong to a certain class, our system can extract a sequence of entries for them.

3.5 Filtering Non-blog Pages

Even if a sequence of entries is extracted from a page, the page might not be a blog, because archives of BBSs, chats, and mailing lists, update descriptions on a site page and announcements of events can also include date expressions. Moreover, a sequence of blog entries and an update description on the blog might simultaneously appear on a page. Therefore, our system tries to remove such non-blog sequences by applying filtering rules to multiple extracted sequences. If our system filters out all the extracted sequences of entries, it judges that the page is not a blog.

We prepare two kinds of the filtering rules: First type is the rules for a page, and second type is the rules for the extracted entries.

- Filtering rules for web pages
 - a page whose URL includes typical keywords for non-blog pages, such as ‘bbs,’ ‘chat,’ and ‘session’
 - a page whose title includes typical keywords for non-blog pages, such as ‘bbs,’ and ‘email magazine’
- Filtering rules for the extracted entries
 - a sequence of entries which includes a entry with a future date expression
(Such sequence might indicate announcements of events.)
 - a sequence of entries whose date interval is more than a month
(Blogs will likely be updated more frequently.)
 - a sequence of entries in which the same date expression repeats three times or more
(Such sequence might be an archive of a BBS or chat.)
 - a sequence of entries whose date expressions are not arranged in ascending/descending order
(Entries are arranged chronologically on a blog page.)
 - a sequence of entries where the size of the 2nd largest entry is less than 150 bytes (counted in UTF-8)
 - a sequence of entries whose average size is less than 150 bytes (counted in UTF-8, and excluding alphabets, numbers, and white spaces to count Japanese characters only)
(Such small entries might be update descriptions on a site page.)

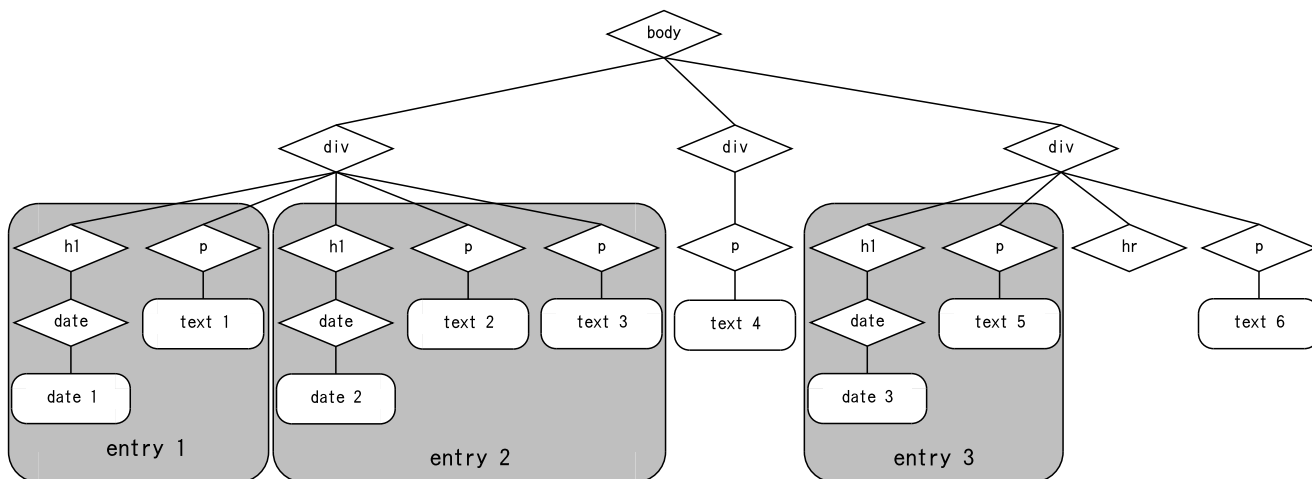


Figure 2: Example of extraction of entries

- a sequence of entries whose date expressions do not appear in the top two lines of an entry (The number of lines is calculated by counting tags with the new-line effect.)
(In a blog page, the date expressions for entries often appear at the top of each entry.)
- a sequence of entries which contain typical keywords included in non-blog pages in all (or more than half) entries, such as ‘admin,’ ‘reply,’ ‘re:,’ and so on
- a sequence of entries which contain no predicates (verb, adjective, etc.)
(A blog page should include descriptions on an event.)

3.6 Filtering Non-blog Pages based on Update Information

Any page judged to be a blog page is monitored by our system. Whenever our system detects update of the page, new entries are extracted and added to our blog database.

Our system also applies the non-blog filters to the page after updating. If we find any violation, the page is judged to be a non-blog page, and the entries which were derived from the page are eliminated from the database, including all the entries which were added in the past. However, the restrictions on the size of an entry are not applied to the page judged to be a blog page any more, because it often occurs that the size of entries becomes temporarily smaller.

Furthermore, there is another case for which a page can be judged to be a non-blog page by using the information acquired from the way of updating. Consider the following example:

- A web page was judged to be a blog page in January 1st, 2004, and entries which had date expressions of December, 2003 were extracted.
- The page had been monitored by our system.
- Update of the page was detected in January 7, 2004, and new entries were extracted.

In this example, it is reasonable to expect that the new entries extracted on January 7, 2004 should have the date

expressions between January 1, 2004 and January 7, 2004. Therefore, our system judges the page to be a non-blog page, if it finds entries with a date expression older than the date of when entries were most recently extracted.

One of the reasons why such a case might be discovered is an error in date expression complementation. When there is no information for ‘year’ within a page, our system complements the information by using the year of when the page was last crawled. Therefore, a entry without ‘year’ information in which events in the future are described will be complemented by the year of one year ago, because our system never complements with a future date. Then, when another future event is added to that page, our system detects a past date because of the complementation error and judges this page to be a non-blog page. Despite the difficulty of knowing whether an entry without ‘year’ describes future or past events, we can filter out such non-blog pages by using this filtering rule.

4. EVALUATION

We operated our system for two weeks from the end of December 2003 and have collected and monitored blog pages. We have obtained 39,272 blogs (pages) and 466,809 entries. These pages have been crawled in the following two ways:

1. Using links in blog pages

First, we applied old version of our blog judgement module to a part of the Web repository which was offered by Kitsuregawa laboratory, Tokyo University. (This repository consisted of about 66 million of JP domain Web pages, and these Web pages were crawled in February 2003.) As a result, we obtained 40,422 blog-like pages from 4,819,143 Web pages. Then, by applying our new version of blog judgement module to these 40,422 blog-like pages, we obtained 11,528 blog pages. (28,894 pages were judged as non-blog pages by added new constraints or could not be crawled in December 2003.) Using these pages as seeds, we crawled Web pages which could be reached by traversing links twice from the pages which we were judged as blog pages. (Note that if a crawled page was also judged as a blog page, our system traversed all links in the page.)

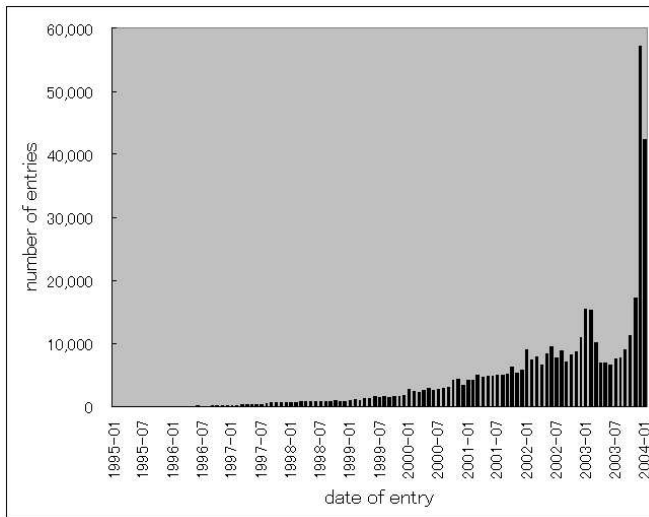


Figure 3: Distribution of the collected blog entries

Table 2: Evaluation for 300 blog pages

blog (correct)	283 pages	(94.3%)
not blog (incorrect)	17 pages	(5.7%)

2. Using ping servers

We also obtained URLs of blog pages by using some popular ping servers in Japan such as “BlogPeople”⁹, “ping.bloggers.jp”¹⁰ and so on.

Note that these crawled Web pages might not be perfect sampling of the Web because we could not process all the crawled Web pages at this time, and moreover, our crawling techniques had some problems.

Figure 3 shows the distribution of the dates for the collected entries. Our system can obtain rather old entries, whereas the collection methods based on RSS and/or the ping server can obtain only recent entries.

300 blogs randomly sampled from the collection were manually evaluated, and we found that 283 were actually blogs (94.3%) and 17 (5.7%) were not, although in 24 of the correct ones, the extraction was not completely correct (Table 2). 17 wrongly identified blogs include 7 BBSs archives, 3 update descriptions on site pages, and 3 announcements of events.

Similarly, we tried to estimate to what degree we can collect blogs without loss. 300 pages were randomly sampled from the collection of pages that were identified to be not blogs and contain at least 3 date expressions, and they were manually evaluated. We found that 201 were actually not blogs (67.0%) and 99 (33.0%) were (Table 3). Since the ratio of the web pages that contain at least 3 date expressions in the collection of pages that were identified to be not blogs was 12.38%, by randomly sampling 30,000 pages from the collection of 175,057 pages, the estimated recall of our

⁹<http://www.blogpeople.net/>

¹⁰<http://ping.bloggers.jp/>

Table 3: Evaluation for 300 non-blog pages

not blog (correct)	201 pages	(67.0%)
blog (incorrect)	99 pages	(33.0%)

Table 4: The percentage of blogs which use blog tools

without blog tools	32,219	(82.0%)
movable type	2,243	(5.7%)
hns ¹¹	1,893	(4.8%)
rakuten ¹²	922	(2.5%)
a-News ¹³	423	(1.1%)
tDiary ¹⁴	307	(0.8%)
cocolog ¹⁵	163	(0.4%)
Akiary ¹⁶	131	(0.3%)
livedoor Blog ¹⁷	127	(0.3%)
MEMORIZE ¹⁸	114	(0.3%)
Tomsoft Diary System ¹⁹	110	(0.3%)
Sarusaru diary ²⁰	106	(0.3%)
using other tools	444	(1.1%)
total page	39,272	

system could be calculated as follows:

$$recall = \frac{39,272 \times 0.943}{39,272 \times 0.943 + 175,057 \times 0.1238 \times 0.33} = 0.838.$$

For more than half (53) of the collection (99) of blog pages that our system judged to be not blogs, the reason is that the time between two adjacent entries was over a month.

By using the perl module that tries to detect what kind of blog creation tool was used for its creation[1] (we patched it for ‘blog tools’ and ‘Web diary tools’ which are popular in Japan), we found that of 39,272 pages, 32,219 (82.0%) were judged as not using any blog tools and that the most popular blog tool was Movable Type (2,243; 5.7%) (Table 4).

5. CONCLUSIONS

We presented a system that tries to automatically collect and monitor Japanese blog collections that include not only ones made with blog software but also ones written as normal web pages.

Although search engines for blogs have become popular, their ability to collect blogs is quite limited because the previous blog collection technique can retrieve only blogs which are made with specific tools. In Japan, however, since long before blog softwares became available, people have written

¹¹hns <http://www.h14m.org/>

¹²rakuten <http://plaza.rakuten.co.jp/>

¹³a-News <http://www.appleple.com/cgi/>

¹⁴tDiary <http://www.tdiary.org/>

¹⁵cocolog <http://www.cocolog-nifty.com/>

¹⁶Akiary <http://www.hi-ho.ne.jp/yakira/akiary/>

¹⁷livedoor Blog <http://blog.livedoor.com/>

¹⁸MEMORIZE <http://www.memorize.ne.jp/>

¹⁹Tomsoft Diary System <http://tds.dive-in.to/>

²⁰Sarusaru diary <http://www.diary.ne.jp/>

'diaries' on the web, that are quite similar to blogs in their content, and people still write them without any blog softwares. To collect both blogs and web diaries, our approach has been based on extracting date expressions and analysis of HTML documents.

The initial results show that our system works well, because blogs written without blog tools and old entries can be retrieved, even though our current system still has some problems. As future work, we plan to refine our system so that it can collect blogs correctly and exhaustively.

We are also developing a system that can extract and mine useful information from the collected blog pages. For details on the mining module of our system, please refer to [5, 2]. At the conference site, we hope we will demonstrate our latest system during the poster session[5].

6. ACKNOWLEDGMENTS

This work was partly supported by the Exploratory Software Project 2003, Information-technology Promotion Agency, Japan (IPA).

This work was also supported by The 21st Century COE Program "Framework for Systematization and Application of Large-scale Knowledge Resources", Japan Society for the Promotion of Science.

This work utilizes the Web repository kindly offered by Kitsuregawa laboratory, Tokyo University.

7. REFERENCES

- [1] M. Ceglowski. Www::blog::identify - identify blogging tools based on url and content.
<http://search.cpan.org/~mceglows/WWW-Blog-Identify-0.06/Identify.pm>, 2003.
- [2] T. Fujiki, T. Nanno, Y. Suzuki, and M. Okumura. Identification of bursts in a document stream. To be submitted to the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, 2004.
- [3] IPA(Information-technology Promotion Agency, Japan). Generic engine for transposable association: Geta. <http://geta.ex.nii.ac.jp/>, 2002.
- [4] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proc. of the 12th International World Wide Web Conference*, pages 568–576, 2003.
- [5] T. Nanno, T. Fujiki, Y. Suzuki, and M. Okumura. Automatically collecting, monitoring, and mining japanese weblogs. Submitted to the Poster Session of the 13th International World Wide Web Conference, 2004.
- [6] T. Nanno, S. Saito, and M. Okumura. Structuring web pages based on repetition of elements. In *Second International Workshop on Web Document Analysis (WDA2003)*, 2003.
- [7] D. Raggett. Clean up your web pages with html tidy.
<http://www.w3.org/People/Raggett/tidy/>.
- [8] J. Wakefield. Catching a buzz. *Scientific American*, 2001. November.
- [9] D. Winer. Weblogs.com xml-rpc interface.
<http://www.xmlrpc.com/weblogsCom>, 2001.