Browsing System for Weblog Articles based on Automated Folksonomy

Tsutomu Ohkura
Graduate School of Information Science and Technology, Tokyo University
ohkura@r.dl.itc.u-tokyo.ac.jp

Yoji Kiyota
Information Technology Center, Tokyo University
kiyota@r.dl.itc.u-tokyo.ac.jp

Hiroshi Nakagawa
Information Technology Center, Tokyo University
nakagawa@dl.itc.u-tokyo.ac.jp

ABSTRACT

Folksonomy is a new manual classification scheme based on tagging efforts of users with freely chosen keywords. In folksonomy, a user puts an item (i.e. a photo, a book mark) on a server and shares it with other users. The owner and even the other users can attach tags to this item for their own classification, and they reflect many one's viewpoints. Since tags are chosen from users' vocabulary and contain many one's viewpoints, classification results are easy to understand for ordinary users. As a result, folksonomy serves as an efficient browsing method, because users can grasp the essence of items by looking at the tags. Even though the scalability of folksonomy is much higher than the other manual classification schemes, the method cannot deal with tremendous number of items such as whole weblog articles on the Internet.

For the purpose of solving this problem, we try to automate folksonomy to enhance weblog browsing. We create a "tagger" which is a program to determine whether a particular tag should be attached to an item. In addition, we propose a method to create a candidate tag set, which is a list of tags that may be attached to items, from weblog category names. We achieved around 95% precision compared to a candidate tag set created manually.

1. INTRODUCTION

1.1 Folksonomy

Recently, a new manual classification scheme, called "folksonomy"[1], has come under the spot light. Folksonomy is a classification scheme by ordinary users. According to [1], folksonomy is:

> The collaborative but unsophisticated way in which information is being categorized on the web. Instead of using a centralized form of classification, users are encouraged to assign freely chosen keywords (called tags) to pieces of information or data, a process known as tagging.

In 2004, Flickr[2] (an online photo sharing service) and del.icio.us[3] (an online bookmarking service) employed folksonomy as a classification method, and they became successful. The success of these services made folksonomy as one of major players of browsing information on the web.

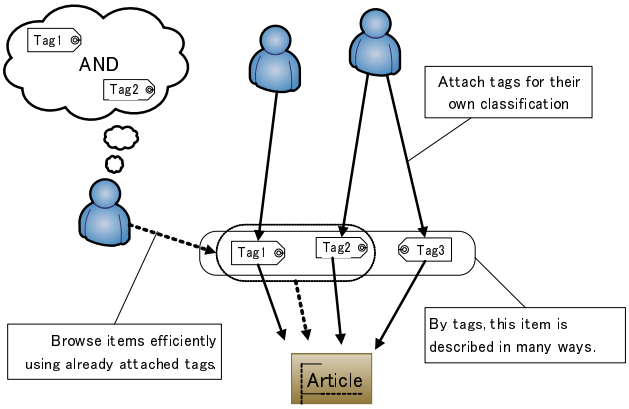

Figure 1: Folksonomy

In folksonomy, tags are user-generated keywords (any vocabulary users can choose) to describe the contents of each item, where an *item* means a photo, a book mark, as well as a weblog article. Tags are usually used to specify the properties of each item when organizing a set of items. They are located in a flat namespace. In other words, they are not categorized in a hierarchy. Since the purpose of the tags is to organize the contents, they are required to:

1. be descriptor words of the items
2. refer to common concepts shared by a group of people.

Collaborative Classification

Folksonomy is a brand-new classification scheme. In contrast to categorization by professionals and authors, folksonomy is classification by ordinary collaborating users (Figure 1).

With the spread of weblogs, the idea of author categorization has become widespread. Since the number of weblog articles is too large to be categorized by a few dedicated professionals, author categorization has become popular for weblogs. The weak point of author categorization is the lack

[1]a word combination of "folk" and "taxonomy"
[2]http://flickr.com/

WWW2006, May 22–26, 2006, Edinburgh, UK.

[3]http://del.icio.us/

of objective viewpoints, which are essential for good categorization.

With folksonomy, another approach is employed: the categorization by ordinary cooperating users. Obviously, each individual user does not have an ability to perform high quality categorization, but if they collaborate with each other, they perform high quality categorization. The collaboration merges multiple users' viewpoints, and brings up the classification quality.

Advantages

Here are some advantages of folksonomy. First, folksonomy can provide serendipity. There are two methods for getting the information you want. These were represented by [2] as:

1. searching to find relevant items in a query, and
2. browsing to find interesting items.

Recently, the first way is popular. Google[4] and other full text search engines are good examples of the first approach. The latter approach was popular in the Internet of 1990s. The hierarchal Yahoo directory[5] was developed for the purpose of browsing, but categorization by a limited number of professionals cannot practically deal with the huge number of web pages. Looking at the cost and limitation of human power, folksonomy is a new promising method for the browsing approach. In addition, since weblogs tend to be viewed in passive styles, folksonomy is promising for browsing weblog articles.

The second advantage of folksonomy is that the classification results are familiar to ordinary people. What should be noted is that taggings are not done by professionals but by ordinary users with their everyday languages. It results in tags that are keywords familiar to ordinary users. Especially, most weblog articles are written in everyday languages. Thus, there is good chemistry between folksonomy and browsing of weblog articles.

In addition, the vocabulary used by ordinary people is changing everyday. For example, a web page concerning "using xHttpRequest JavaScript" should be classified in "JavaScript" in 2003. In 2005, however, it should be classified in "Ajax"[6]. Looking at the changing vocabulary, folksonomy has a big advantage to other methods. In the case of categorization by professionals or by authors, re-classification requires much efforts. However, with folksonomy, when the vocabulary changes, new users attach tags to old articles based on the new vocabulary.

1.2 Automated Folksonomy

Looking at professionals' classification, the scalability is quite limited. Limited numbers of dedicated professionals are unable to deal with a large number of items, such as weblog articles. Folksonomy seems to be able to deal with the large amount of content. There are, however, in folksonomy, many items that are tagged by only a small number of people, and they do not receive enough description. It is one of the weakest points of folksonomy. To work around the problem, it is inevitable to realize automated tagging. And then, we propose an automated folksonomy system, and describe the technical issues of the system in the following part of this paper. The idea is shown in Figure 2, where the system is seen as the same as the folksonomy system depicted by Figure 1 except that people who are tagging are replaced with a machine. We should say that the tagging of the system should also utilize the advantages of folksonomy.

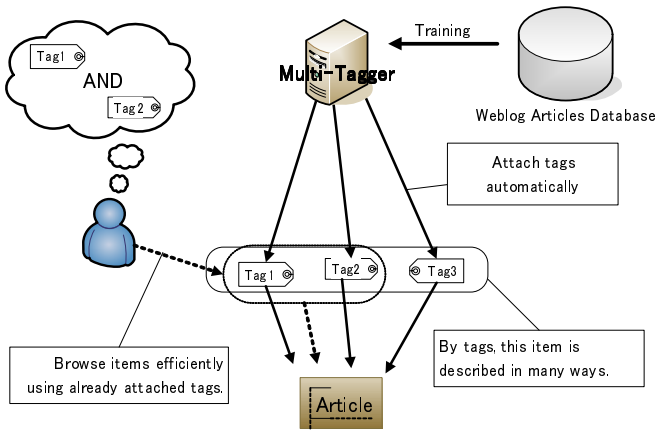

Figure 2: Automated Folksonomy

We think that automated folksonomy should satisfy the following requirements:

1. Tags are selected from a users' vocabulary
2. It should deal with a changing vocabulary
3. The concept of each tag should contain many people's viewpoints.

2. SYSTEM OVERVIEW

The overview of our automated folksonomy system is shown in Figure 3. This system is an automated multi-tagging system for weblog articles. For each weblog article, the system attaches multiple tags. Tag names and their concepts are automatically extracted from collected weblog articles.

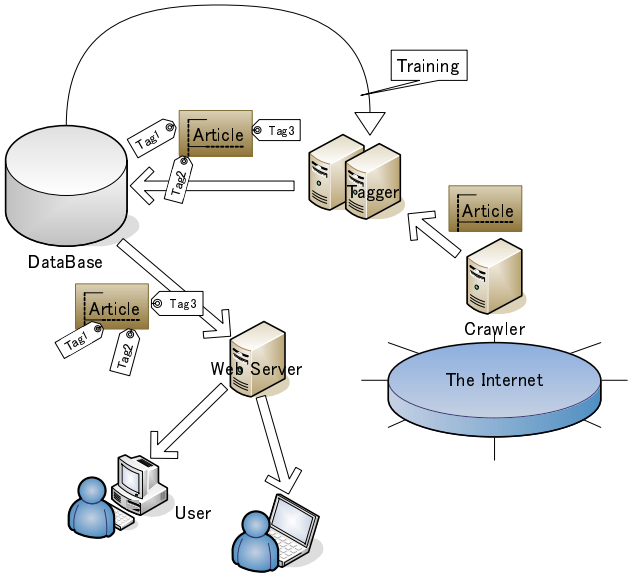

Figure 3: System Overview

[4]http://www.google.com/
[5]http://dir.yahoo.com/
[6]Asynchronous JavaScript + XML

The number of weblog articles is rapidly growing every day. Even when limiting it to Japanese weblogs, more than 10 million articles seem to be newly posted every month. In other words, four new articles are posted every second only in Japan. To deal with this amount of text data, methods applied to weblog articles should be efficient.

The system consists of the following three parts: a weblog articles crawler, a multi-tagger, and a user interface.

Crawler

The crawler is a program for fetching newly posted weblog articles. It uses the information provided by ping services[7]. Most weblog tools notify some ping services about their modification.

After fetching a new article, the crawler extracts words from the article, and stores these words collection into the database. MeCab[8] is used for extracting words from Japanese texts.

Multi-tagger

Tags are attached to each article by a multi-tagger. Our multi-tagger is an array of taggers that determine whether to assign a particular tag or not (see Figure 4).

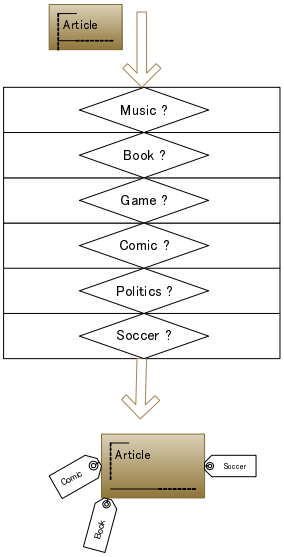

Figure 4: Multi-tagger

Each tagger is a two-class classifier based on a SVM. Details on the tagger are described in Section 3.

The first step in constructing a multi-tagger is to create candidate tag set, which is a set of tags that may be attached to the items. The multi-tagger assigns a subset of candidate tag set to each item. Details on the creation of the candidate tag set are described in Section 4.

For dealing with the changing vocabulary of ordinary people, every tagger is retrained regularly (e.g. once a day).

User Interface

The system has a web-based user interface. Users can efficiently navigate among the huge amount of weblog articles using attached tags. Viewers can get their favorite weblog articles by manually selecting a combination of tags. In addition, they can browse tags using the "related tags" function.

[7]a service that provides recently modified weblog URLs
[8]http://mecab.sourceforge.jp/

3. TAGGER

A multi-tagger is a tagger array. Each tagger is a classifier that determines whether a particular tag is appropriate for an article or not. For each tag of the candidate tag set, the corresponding tagger is created.

A tagger should be periodically trained. We should use the latest weblog articles as the training data, because the tagger should take latest trends into consideration.

For the purpose of multi-tagging, the following conditions are required for a classifier in each tagger. (1) Fast classification speed, (2) low memory consumption (during classification), (3) over fitting avoidance, (4) high classification accuracy. Taking into account of these four conditions, we compared four popular text classification algorithms: k-Nearest Neighbor, Naive Bayes, AdaBoost, and SVM, and we finally selected SVM.

For training each classifier, we used our collected weblog articles described in Section 2. We used articles which were categorized into "A" by those authors, as positive examples for a classifier corresponding to the tag "A". If there are more than 2,000 articles, we used the latest 2,000 articles. As negative examples, we use the same number of randomly selected articles categorized into other than "A". The threshold "2,000" is decided for reducing the time consumption of the experiment, in spite of the fact that we know more articles make the classification accuracy better.

We employed the bag-of-words model. As usual, we converted each weblog article into a vector using the vector space model. In addition to words, since many periodical events are available, we used posting date/time information. The date/time features contributed around 1% improvements in accuracy.

From the browsing point of view, the costs of two types of errors, namely false negative and false positive, are not the same. Since a large number of weblog articles are available, missing tags (false negative) is not critical. However, attaching tags when an item should not be tagged (false positive) is harmful for the efficient browsing experience. Taking into account this condition, we should bias the SVM outputs. To fulfill the requirement, we converted the SVM output value into a probability by the method proposed by [3], and used 90% as the threshold.

4. CANDIDATE TAG SET SELECTION

For multi-tagging, we should prepare a candidate tag set. Multi-tagger assigns a subset of the candidate tag set to each weblog article. In this section, we describe the details of the method to create the candidate tag set.

We create a candidate tag set by selecting category names used on all the weblog sites. This is because most weblog services allow users to construct their own category systems, and many weblog articles are classified by their authors. The important point to note is that these category names depend on the weblog authors' vocabulary. Taking these situations into account, we chose popular and descriptive category names (as stated in Section 1), and used them as the candidate tag set. It should be noted that the selection

should be repeatedly performed for reflecting new category names. Some examples of these category names are shown in Table 1.

First, we measured the popularity of a category name by the number of weblog sites containing the category name in its categorization system. If the category name is a popular word (or a popular short phrase), they should be used by multiple bloggers. We experimentally employed five weblog sites as the threshold for the popularity.

Second, we measured the descriptiveness of each category name. We prepared a classifier for each category name. These are SVM classifier used in the taggers. If a category name is not descriptive, the classification accuracy of the corresponding classifier should be low.

Two conditions are required to the process of deciding whether a category name is descriptive or not. First, unsuitable tags in a candidate tag set should be minimized. Second, the required sample data to make a judgment should be kept minimal.

To fulfill these two requirements, we have to estimate the proper classification accuracy using a small number of articles. And then, we can see a correct classification as a probability event, and by defining p as the proper accuracy of the classifier, the probability of the event is p. Since we do not have a priori knowledge about this distribution, it is natural that the distribution of the classification result is assumed to be a binomial distribution. In addition, according to the central limit theorem, the distribution can be approximated by the normal distribution when n is not too small.

As a result, we can calculate the confidence interval ([4]). For example, setting the confidence rate at 99.5%, and the lower bound of confidence interval of the proper accuracy can be calculated as follows:

$$\frac{c}{n} - 2.58 \times \sqrt{\frac{c}{n}(1.0 - \frac{c}{n})/n}.$$

When the lower bound of the confidence interval exceeds 75% (the threshold will be chosen in Section 5.2), we can judge that the category name is suitable for the candidate tag set.

5. EXPERIMENT

5.1 Experimental Data Set

We used real Japanese weblog articles as our experimental data set. We collected them from the 13th of April, 2005 to the 1st of December, 2005.

We treat the dc:content_encoded section or the description section of an article as its contents, and dc:subject section of an article as its category.

Articles categorized in "Uncategorized", "Diary","weblog", "News", blank [9] were removed from the experimental data set, because weblog articles of these categories are too many, and they prolong our experimental time.

There were 2,460,374 articles and 144,789 categories in total. Investigating these articles, we can see that most of the categories have small numbers of weblog articles, and only a small portion of categories have more than 1,000 articles.

[9]Since our experimental data are Japanese weblog articles, category names are Japanese. In this paper, for convenience, all category names are translated into English.

5.2 Classification Accuracy

We measured the classification accuracy of the classifiers for more than 40 tags. In this experiment, the test set to measure the accuracy is, roughly speaking, increasing according to the number of articles. A part of the results is shown in Figure 5. It is obvious that there is a dichotomy of the classification accuracies for the categories.

The category names in a lower group, say "DIARY" and "Daily Life" in Figure 5, are ambiguous and do not specify their contents. Taking Figure 5 and the same experiments on the other categories into consideration, we chose 75% as the classification accuracy threshold of candidate tag set selection.

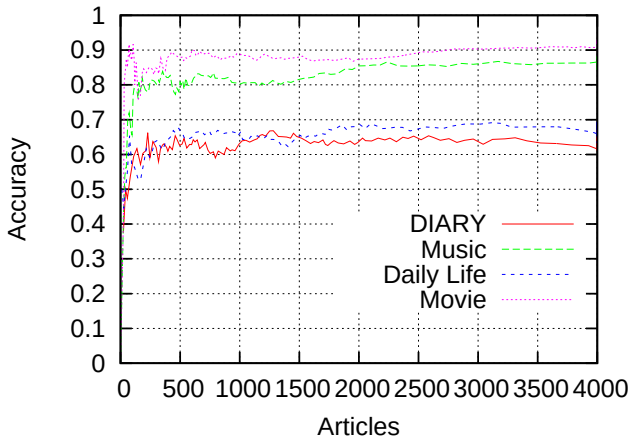

Figure 5: Classification Accuracy of Descriptive and not Descriptive Category Names

5.3 Candidate Tag Set Selection

The candidate tag set selected by our program is compared to a candidate tag set selected by subjects. We applied the algorithm described in the previous section to category names in our experimental data, and compared the output of our program with manually selected tags.

The resulting accuracies and precisions are shown in Table 2. The experimental result implies that our algorithm can select the appropriate candidate tag set. In particular, the precisions of the results are very high. Even when the training has done with about 100 articles, the precision rate is kept high.

The significant false positive error is occured in "Notification". Although this category name seems to imply an overall notification, it actually implies the notification concerning each weblog site. Due to the concept that a notification about each weblog site is specific, the mistake has occurred. In a real situation, a few mistaken category names like "Notification" can be removed by hand.

5.4 Tagger

We performed experiments on tagging by measuring the precision and recall of our taggers, which are limited to categories that have more than 800 articles.

The results of the experiment are shown in Figure 6. Most of the taggers have high precisions. However, recalls of these taggers vary from 0.2 to 0.8. By the way, looking at the situation where users browse weblog articles guided by tags,

Table 1: Category Names and Their Goodness as a Tag

"Tag Name" (Occurence in our experimental data; Accuracy of Classifier when it was trained with 500 articles)

Good Tag	Lack descriptiveness	Lack popularity
"Music" (26316; 80%)	"DIARY" (35630; 62%)	"Blogurmet" (1; -)
"Movie" (17476; 88%)	"Daily Life" (63857; 66%)	"Comparison of Cleaners" (1; -)
"Japanese Sweets" (392; -)	"Others" (33252; 60%)	"The Wing Goes to the Sky" (1; -)
"Final Fantasy XI" (2502; 88%)	"Murmur" (13115; 71%)	"Contents of Subjects" (1; -)

Table 2: Evaluation of Candidate Tag Set Selection

articles	tp	fp	fn	tn	accuracy	precision
>= 100	91	3	113	135	66.0%	96.8%
>= 200	76	3	67	107	72.3%	96.2%
>= 400	41	2	24	62	79.8%	95.3%
>= 800	27	1	5	34	91.0%	96.4%
>= 1600	11	1	0	16	96.4%	91.6%

[articles] Condition of applied category names. ">= 100" means "category names which have more than 100 articles."
[tp] Category names where both the algorithm and human answered it is suitable as a candidate tag.
[fp] Category names where our algorithm indicates that it is appropriate for candidate tag set, but a human do not agree with this.
[fn] Category names where our algorithm indicatess that it is not appropriate for candidate tag set, but a human do not agree with this.
[tn] Category names where both the algorithm and human answered it is not suitable as a candidate tag.
[accuracy] Accuracy of our algorithm compared to human selection. $(tp + tn)/(tp + tn + fp + fn)$.
[precision] Precision of our algorithm compared to human selection. $tp/(tp + fp)$.

precision is very important. From this point of view, our tagger is proven to work well.

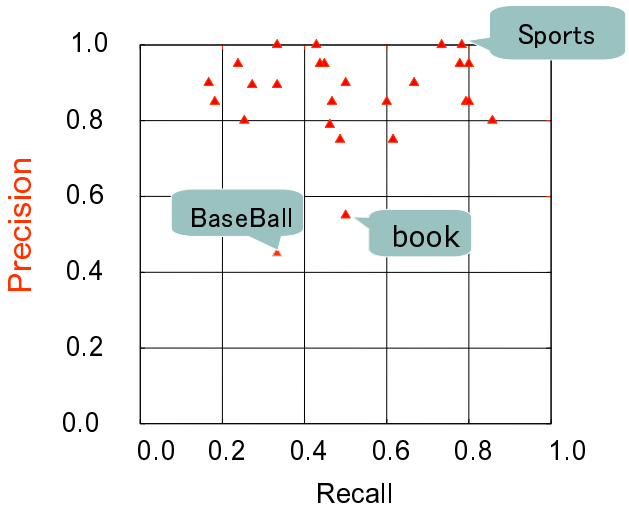

Figure 6: Recall and Precision of Taggers

6. DEMO

Here, we demonstrate our system. Our system is implemented on a web server. On the entrance page of our system, tags are listed (see Figure 7).

If you click the "Music" tag, the system displays listed weblog articles to which the "Music" tags are attached by our system (see Figure 8. [10]). These articles are related to music, not only articles categorized into the "Music" by these authors but also the other categories. In addition, on

[10]We use the theme picked from scuttle project in Figure 8,9 (http://sourceforge.net/projects/scuttle/)

Figure 7: Entrance Page

the right side of the page, tags related to the "Music" tag is listed. These related tags are selected by our system based on tag co-occurrence.

Next, if you click "TV" from listed related tags, the system displays the listed weblog articles that have both the "Music" and the "TV" tags (see Figure 9).

7. DISCUSSION

Advantages and Limitations

First of all, we emphasize the scalability of our method. Since all taggers are independent, the automated folksonomy can be implemented with paralleled machines. In our experiments, we used eight (virtual) processors on two machines that do not share their memories.

Since our proposed method is a replacement of the tagging by users, our method and most of the existing methods

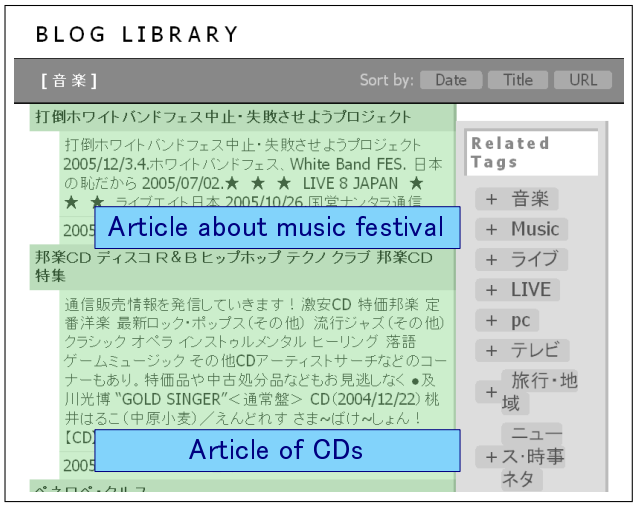

Figure 8: Articles to which the "Music" tags are attached

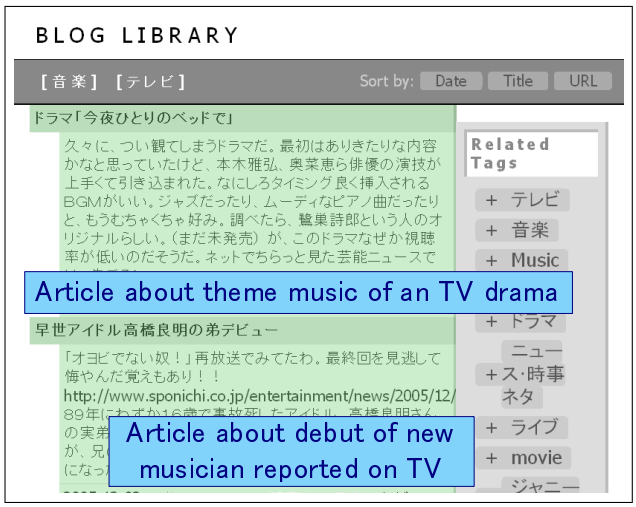

Figure 9: Articles to which both the "Music" and the "TV" tags are attached

related to folksonomy can be used together. Looking at the existing services with folksonomy, many useful techniques are available. These method can be used in coordinated with our automated tagging. In addition, our method can be used even in a user-tagged folksonomy system. Using our method for tag suggestion, better suggestions can be expected.

Since our tag set is focused to browsing, some of the popular tags used in online bookmarking service is not available in our tag set. These are tags which has attached only for personal use, such as "fun" or "personal".

However, compared to folksonomy based on user's tagging, our method attaches too many tags. If an article is about economics and it contains a bit of content related to a game as an example, our system attaches the "Game" tag to this article. When a user who seeks game articles uses our system, the above article is shown. Although the system can determine if a particular concept is contained or not, it cannot determine whether that concept is a peripheral one or one of the main subjects.

Related Work

Looking at multi-tagging (or multi-labeling), many approaches have already been proposed ([5] etc). However, our proposed methods are totally different from most of them. Most studies concerning multi-tagging train their taggers with multi-tagged documents. In contrast, our method learns tags and their concepts from categorized (assigned to one category) documents. The reason why we employ this approach is that the majority of weblog articles are assigned to only one category. Moreover, since multi-tagged weblog articles are not tagged using the same candidate tag set, usual multi-labeling training approaches are not suitable for our aim.

[6] showed the importance of the centralized topic-centric view of the weblog sphere. Our approach aims at the same objective but these two approaches are different. Their approach treats a category as a unit and the relations between the categories are managed. In contrast, our approach treats a weblog article as a unit and each article is classified by its topic.

8. CONCLUSION

We propose a new content browsing method based on "automated folksonomy." To produce an automated folksonomy for weblog articles, we described three requirements in Section 2. For fulfilling these requirements, we developed a multi-tagger based on SVMs with an automatically selected candidate tag set.

Acknowledgments

We thank Mr.Tomohiro Fukuhara for providing weblog data used in our experiments.

9. REFERENCES

[1] Wikipedia. Folksonomy — wikipedia, the free encyclopedia, 2006. http://en.wikipedia.org/wiki/Folksonomy [Online; accessed 29-January-2006].

[2] Mathes Adam. Folksonomies - cooperative classification and communication through shared metadata, 2005. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html [Online; accessed 29-January-2006].

[3] John C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[4] Wikipedia. Confidence interval — wikipedia, the free encyclopedia, 2006. http://en.wikipedia.org/wiki/Confidence_interval [Online; accessed 29-January-2006].

[5] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.

[6] Paolo Avesani, Marco Cova, Conor Hayes, and Paolo Massa. Learning contextualised weblog topics. In *WWW 2005 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2005.