Automatically Linking News articles to Blog entries

Daisuke Ikeda
Dept. of Computer Science
Tokyo Institute of Technology
2-12-1 Oh-okayama Meguro
Tokyo 152-8550 Japan
ikeda@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

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

Abstract

People often write in their blogs about news articles or events in news articles. In this case, however, the details of the news articles or events are often poorly described in such blog entries. Therefore, the readers of blogs need to find the original articles, which contain more details of the news articles, when they want to know about them.

In this paper, we propose a method for linking news articles to blog entries that refer to them. Since blog entries and news articles are considered to be rather different, the common model for linking is not applicable. Therefore, we similarly used a vector space model for finding similar documents, but we tried to devise a new weighting method and a distance metric to improve the performance, by taking into account the properties of blog entries and news articles document sets.

INTRODUCTION

Today, weblogs(blogs) have spread out rapidly, and their value as source of information is increasing . With blogs, people can easily describe what they want. Therefore, their opinions and impressions on many different topics tend to be expressed in blogs.

People often write in their blogs about news articles or events in the news articles. In the case, they tend to write their opinions and impressions for them. However, the details of the news articles or events are less described in such blog entries. Therefore, the readers of blogs need to find the original articles, which contain more details of the news articles, when they want to know about them. If blog entries contained links to articles, readers could easily follow the links. However, such links do not always exist in blog entries.

There are also requests to find opinions and impressions of news articles in blogs. If information from many blogs referring to a specific news article can be obtained, an aggregate of many opinions and impressions can be seen. However, this is difficult to do, as the number of blogs increases daily.

In this paper, therefore, we propose a method for linking news articles to blog entries that refer to them. With the information of the link, we can find news articles referred to by blog entries. Similarly, we can also find blogs that mention a news article.

The rest of the paper is organized as follows. In the next section, we introduce related work. Following that, we will explain our linking method. Then, we will describe the data sets used in our research. In addition, we will evaluate our method. Finally, we will describe our conclusions and future work.

RELATED WORK

There has been much work about determining a structure for sequences of news articles (Uramoto & Takeda 1998), including work on topic detection and tracking(TDT) (Allan, Papka, & Lavrenko 1998). Such work enables us to find and access news articles easily. TDT includes tasks to determine whether a news article belong to previous sequences of news articles or is a new topic by taking into account similarities between news articles. However, TDT only deals with news articles, while we will discuss the similarity between blog entries and news articles.

Google News(BETA)[1] is a web service that provides grouped news articles on the web and the interfaces to retrieve them. In this service, they crawl and collect news articles from a number of news sources on the web, especially websites of newspaper companies. Collected articles are grouped and their importance values are calculated. A retrieval interface similar to Google News is also provided by Newsblaster(McKeown *et al.* 2002) and NewsInEssence(Radev *et al.* 2001).

Several blog search services, such as Technorati[2], provide news or book popularity ranking in blogs. Most services consider only explicit links from blogs to news or books. As we discussed in Introduction, however, such links do not always exist, and relying on explicit links alone can be considered to be harmful. In our work, therefore, we tried to identify implicit links between blog entries and news articles. By adding the information from such hidden links, we think more precise ranking can be possible.

The patent retrieval task in the 4th NTCIR workshop (Fujii, Iwayama, & Kando 2004) dealt with techniques to retrieve patent documents given a news article. In the task, probabilistic retrieval models were often used. We tried to

[1] http://news.google.com/
[2] http://technorati.com/

link news articles to blog entries rather than patent documents. Therefore, it is natural to think that the models for the NTCIR patent retrieval task can be applied to our task. We will mention the point in the section of Term Distillation again.

Term Distillation

Itoh et al.(Itoh, Mano, & Ogawa 2002) suggested a technique for cross-database retrieval, called Term Distillation. They applied the technique to cross-database retrieval of news articles as queries with a patent database to retrieve, and succeeded to improve the performance. Since our task can be considered as cross-database retrieval with news articles as queries and a blog database to retrieve, we think Term Distillation can be also applicable. In the section of Evaluation, we will describe the comparison of our method with Term Distillation.

The distribution of occurrence frequency of words is considered to be rather different between a query database and a database to retrieve in cross-database retrieval. Therefore, unimportant words could have large scores when using TFIDF to weight words.

Term Distillation is a technique that tries to prevent such cases by filtering out words that could receive incorrect weights. Specifically, all the words in a query document are candidate query words first. Then, for each candidate word, w, a score, TDV, is calculated. TDV is defined as follows:

$$TDV(w) = tf(w) \cdot \frac{p(1-q)}{q(1-p)}, \quad (1)$$

where p is the probability of occurrence for word w in the query database, q is the probability of occurrence for word w in the database to retrieve, and $tf(w)$ is the frequency of w in a query. A fixed number of candidate words that get high TDV scores are used for the retrieval.

OUR LINKING METHOD

Basic Framework

Commonly, linking a document to another document can be seen as relevant document retrieval or finding similar documents for an inputted document. Therefore, for linking news articles to blog entries, common methods for relevance document retrieval or finding similar documents are considered to be also applicable.

Linking news articles to blog entries can be done with the following four steps using a vector space model:

1. Create word vectors from news articles,
2. Create word vectors from blog entries,
3. Calculate the similarity between the vectors of news articles and blog entries,
4. Link news articles to blog entries in cases where the similarity exceeds a pre-determined threshold.

Here, both news articles and blog entries can be query documents for finding similar documents. However, trying to retrieve relevant news articles from blog entries is unrealistic, because of the large number of blog entries, most of which do not refer to any news articles. Therefore, in this paper, we decided that news articles should be query documents.

For the distance metric and the weighting method for the word vectors, the cosine metric and TFIDF weighting are common models and have been considered to yield successful results. However, as we will show in the evaluation section, such models are not applicable to the cases of cross-database retrieval, where we try to find similar documents in different databases, and the performance will tend to degrade as Itoh et al. said in the patent retrieval task (Itoh, Mano, & Ogawa 2002). This might be caused by the difference in the properties of two databases, the one to find documents and the one for the input document.

Linking news articles to blog entries is a form of cross-database retrieval. In news articles, details of occurred events tend to be mainly described. On the other hand, in blog entries, rather than describing the details of news articles or events, authors tend to write their opinions and impressions instead.

Therefore, we also use the vector space model for a method for finding similar documents, but we tried to devise a new weighting method and a distance metric to improve the performance, taking into account the properties of blog entries and news articles databases.

Creating word vectors from news articles

In creating word vectors, determining which words to select and how to weight them is important. Usually, in news articles, titles and leading sentences are considered to be important as summaries. In summarization task, therefore, the lead method (Edmundson 1969), which extracts a couple of the lead sentences, is often used for news articles summarization.

Therefore, we also thought that the words in titles and first sentence were important in a news article, and we selected them as words for word vectors.

Since TFIDF is well known as the weighting method for words, we also use it in our method. The weight for word w is defined as follows:

$$\begin{aligned} weight_{news}(w) &= TFIDF(w) \\ &= tf(w) \cdot \left\{ \log \frac{N}{df(w)} + 1, \right\} \end{aligned} \quad (2)$$

where $tf(w)$ is the term frequency of word w in the news article, $df(w)$ is the number of news articles that word w appears, and N is the total number of news articles.

Creating word vectors from blog entries

Unlike news articles, from which important parts are relatively easily identified, blog entries tend to contain multiple topics, and extracting important parts is difficult. Therefore, we used all the words in blog entries for word vectors.

In a news article, important words usually appear repeatedly, because news articles tend to talk about events. Therefore, term frequency of important words in articles is higher, and TFIDF weighting will yield better results.

However, TFIDF weighting would not work well in blog entries. As we discussed before, in blog entries, rather than describing the details of news articles or events, authors tend to write their opinions and impressions. This means that important words to indicate the referring to news articles might not appear repeatedly.

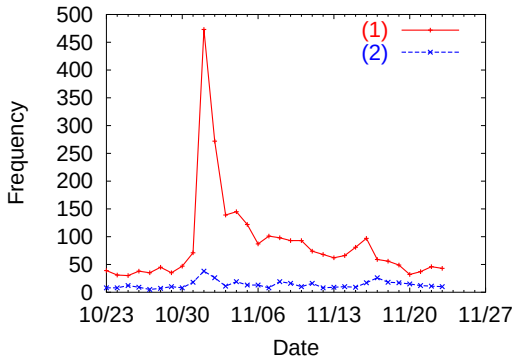

Figure 1: Document frequency of (1)“bill” and (2)“portrait” in blog entries

Thus, we can conclude that the frequency of words in entries is not so important. That is, we use IDF rather than TFIDF as weighting for words in blog entries. The weight for word w can be written as

$$weight_{blog}(w) = IDF(w) = \log \frac{N}{df(w)} + 1, \quad (3)$$

where $df(w)$ is the number of blog entries that word w appears, and N is the total number of blog entries.

Ideally, in linking news articles to blog entries, important words as elements in a word vector for a blog entry are those that are not usually used so much in blog entries but appear more frequently just after the news article is released, because a news article tends to be referred to just after it is released, and the frequency of words in it increases.

The similarity score between a news article and a blog entry that refers to it can be higher if higher weights are given for those important words, and the linking performance will be improved. However, the IDF method does not always give higher weights to such words, since it gives higher weights in cases where words rarely appear in blog entries.

Consider the following example. On November 1, 2004, a new bill was issued in Japan, and news articles about this were frequently referred to by many blog entries. The news articles also mentioned that the portrait had been changed on the new bill. Therefore, the words “bill” and “portrait” must be important in the news article.

The change in the number of blog entries in which the words “bill” and “portrait” appear is shown in Fig. 1. The frequency of “bill” increased rapidly for several days just after November 1. However, frequency of “portrait” did not increase much.

Here, “bill” should be more important than “portrait,” though the IDF weight for “portrait” would be bigger than that for “bill”.

To implement this idea, we compared the usual IDF weights with those for the period just after a news article is released. If IDF for the short period was obviously smaller than the IDF for the whole period, it means that the frequency of the word has increased because of the release of the article. Therefore, our new weighting method for words in blog entries is as follows:

$$new\_weight_{blog}(w) = weight_{blog}(w) + IDF(w) - IDF(w,d), \quad (4)$$

where $IDF(w,d)$ is the IDF for word w over the period d.

As you know, $IDF(w)$ doubles by this. Since, $IDF(w) - IDF(w,d)$ only shows amount of change of IDF in a short period, and it may take zero or negative value, it is not easy to use only this value as the weight. Therefore, we added this value to IDF as the weight.

For the reasons why the period when IDF is remarkably changed is only a short span of time, d is assumed to be three days after the news article was released, because the period when IDF is remarkably changed is very short.

Unlike normal document sets, topics in blog entry datasets change with time. It can be considered to be one of the characteristics of blog entries as texts. The change of document frequency in blog datasets reflects the topic change in blogosphere. Furthermore, topic changes are caused by the release of news articles. Therefore, the weighting method can be said to naturally implement the properties of blog datasets.

Calculating similarity between two vectors

The cosine metric has traditionally been used for calculating the similarity between two documents. It is the inner product, normalized by lengths of the documents. The cosine similarity is defined as follows:

$$sim_{cosine}(news, blog) = \frac{\sum weight_{news}(w) weight_{blog}(w)}{\sqrt{\sum weight_{news}(w)^2 \sum weight_{blog}(w)^2}}. \quad (5)$$

However, since blog entries tend to contain multiple topics, and it is only their fragments that correspond to the descriptions of news articles, normalization by the length would negatively impact longer entries.

Therefore, we used the inner product for the similarity measure, instead of the cosine measure. The similarity is defined as follows:

$$sim(news, blog) = \sum_w weight_{news}(w) weight_{blog}(w). \quad (6)$$

DATA SETS

In this section, we discuss data sets we used in our experiments, blog dataset and news article dataset.

Blog data collected by blogWatcher system

The blogWatcher system is a Japanese blog search engine 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(Nanno *et al.* 2004).

The approach of the system 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, the system also extracts and mines useful information from the collected blog pages.

In the system, therefore, blog entries are those for a day, rather than entries written with a usual blog tool, and tend to contain two or more topics.

We used data collected by this system as the blog data in our experiments.

News article data

We used online news articles and tried to make links between blog entries and news articles.

We collected news articles from the websites of the most famous newspaper companies in Japan, the Asahi Shinbun the Yomiuri Shinbun , and the Mainichi Shinbun Each news article had a title, an entire body, and a date.

Preprocessing of Data

First, we removed all the HTML tags from both the news articles and the blog entries. Then, we applied the Japanese morphological analyzer ChaSen[3] and gave part of speech tags to the words. In our work, we use only the nouns for linking.

EVALUATION

In this section, we describe the experiments we carried out. We made three evaluations. First, we examined whether our model with the inner product and IDF weighting for blog entries outperformed the common model with the cosine metric and TFIDF weighting. Second, we examined whether our new weighting method for blog entries is effective or not. Finally, we compared our linking method with the previous work with term distillation.

First, we explain the evaluation method and evaluation measures. Then, we will show each evaluation and its result one by one.

Evaluation method and measures

In the evaluation, we used news article data and blog data issued from October to December 2004, including 38,912 news articles and 1,372,574 blog entries.

We selected 100 news articles at random from the news article dataset. Then, for each article, blog entries to link to were calculated with the methods to be evaluated. We limited the output to 50 blog entries. The entries were manually evaluated, and their correctness was judged.

We assumed that blog entries linking to news articles are posted in the week after articles are released. Therefore, in finding blog entries to link to news articles, we tried only entries written within seven days of the release of an article.

We used recall, precision, and F-measure as evaluation measures.

By changing the threshold of similarity for linking, we tried to find the best F-measure for each evaluation. We will show the best F-measure as the evaluation results.

Do cosine and TFIDF work well?

As mentioned above, we think cosine similarity and TFIDF weighting do not always work well when we calculate similarity between news articles and blog entries, unlike usual document retrieval. Then, we suggested that inner product and IDF weighting for blog entries are better for linking between them. First of all, therefore, we show the effectiveness of using the inner product with IDF weighting.

[3]http://www.chasen.org/

Table 1: Four methods in first evaluation

	blog weight	similarity
Method 1	IDF	Inner
Method 2	IDF	Cosine
Method 3	TFIDF	Inner
Method 4	TFIDF	Cosine

Table 2: Results of first evaluation

	recall	precision	F-measure
Method 1	0.667	0.694	**0.680**
Method 2	0.564	0.640	0.567
Method 3	0.340	0.223	0.269
Method 4	0.377	0.339	0.357

We tried four methods in our evaluation, as shown in Table 1. In the table, "blog weight" refers to the weighting method of word vectors for blog entries, either IDF or TFIDF. "Similarity" refers to which metric was used, inner product or cosine. The method we suggested in this paper is method 1, and the most common method is method 4.

The results are shown in Table 2. Method 1, our suggested method, yielded the best performance, as we expected. We can say that inner product and IDF weighting for blog entries worked well in linking news articles to blog entries.

Method 4, the most common method, did not outperform our method. This means that the usual method for document retrieval is not always applicable for cross-database retrieval.

Evaluation of new weighting method

Next, we show the effectiveness of our new weighting method for blog entries. We used method 1 from the last evaluation as a base line model. In this evaluation, we compared the weighting in equation (3) with that in equation (4) for word vectors of blog entries. Table 3 shows the result. New weighting method outperformed the base line model.

We can imagine that in cases where the change of document frequency is remarkable, the effectiveness of the method would be large. Since the remarkable change of frequency tends to occur for news articles that attract more attention and are referred to by more blog entries, our method can be considered to be more effective for those 'human interest' articles.

Comparison with previous work

Finally, we compared our linking method with previous work dealing with term distillation.

For each word in a query news article, we calculated TDV, defined in equation (1). We used the words with the fifteen highest TDV scores as queries for retrieval. We used TFIDF for weighting words for news articles. We compare the result with term distillation to the one with method 1.

The results are shown in Table 4. The performance with term distillation degraded, though it yielded better performance in cases of patent retrieval.

We think the following can be the reason for the poor performance of term distillation. In patent documents, more

Table 3: Results of second evaluation

	recall	precision	F-measure
New weight	0.780	0.675	**0.724**
Base line	0.667	0.694	0.680

Table 4: Results of third evaluation

	recall	precision	F-measure
Term distillation	0.510	0.528	0.519
Method 1	0.667	0.694	**0.680**

Figure 2: The screenshot of the result of blog search

technical terms tend to appear than in news articles. Therefore, by selecting words to be used for queries the words which are not so used in news articles but are more commonly used in patent documents, term distillation can yield better results. However, unfortunately, in blog entries, more ordinary words tend to appear than in news articles. In these cases, term distillation will pick up only those ordinary words that are not so useful for identifying which news article to refer to.

APPLICATION

Our method has been implemented on the blogWatcher system `http://blogwatcher.pi.titech.ac.jp`. As mentioned above, blogWatcher is a blog search engine. If a retrieved blog entry in the search result has a link to a news article, the link is shown and you can easily reach the news article.

Figure 2 shows the screenshot of the search result by our system. The number of news articles linked to the retrieved entry is displayed and the list of them is displayed below the snippet of the entry. In addition, the number of blog entries linked to the article in the list is displayed. You can search blog entries again from the news articles in the list.

CONCLUSION

In this paper, we proposed a method for linking news articles to blog entries that refer to them. We devised a new weighting method and a distance metric to improve the performance, by taking into account the properties of blog entries and news articles.

We assumed that the document frequency of important words for linking in blog dataset increases just after the news article is issued. As we expected, the weighting method to implement the idea obtained a good result. The weighting method can be said to naturally implement the property of the blog dataset.

By the experiment, we found that the cross-database retrieval methods for patents does not always work well for blogs. In cross-database retrieval for blogs, we needed to prepare a special method for blogs.

Acknowledgement

We thank Tomoyuki Nanno and other members of the blogWatcher development team for their implementation of this work into blogWatcher.

References

Allan, J.; Papka, R.; and Lavrenko, V. 1998. On-line new event detection and tracking. In *Proceedings of ACM SIGIR'98*.

Edmundson, H. P. 1969. News methohds in automatic abstracting. *Journal of ACM*.

Fujii, A.; Iwayama, M.; and Kando, N. 2004. Overview of patent retrieval task at ntcir-4. *Proceedings of Working Notes of the 4th NTCIR Workshop Meeting*.

Gulli, A. 2005. The anatomy of a news search engine. In *Proceedings of The 14th International World Wide Web Conference*.

Itoh, H.; Mano, H.; and Ogawa, Y. 2002. Term distillation for cross-db retrieval. In *Proceedings of Working Notes of the 3rd NTCIR Workshop Meeting, Part III : Patent Retrieval Task*.

McKeown, K. R.; Barzilay, R.; Evans, D.; Hatzivassiloglou, V.; Klavans, J. L.; Nenkova, A.; Sable, C.; Schiffman, B.; and Sigelman, S. 2002. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of The Human Language Technology Conference*.

Nanno, T.; Fujiki, T.; Suzuki, Y.; and Okumura, M. 2004. Automatically collectiong monitering, and mining japanese weblogs. In *Proceedings of 13th International World Wide Web Conference*.

Radev, D. R.; Blair-Goldensohn, S.; Zhang, Z.; and Raghavan, R. S. 2001. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of The Human Language Technology Conference*.

Uramoto, N., and Takeda, K. 1998. A method for relating multiple newspaper articles by using graphs, and its application to web casting. In *Proceedings of the Conference on COLING-ACL*.