Building Web Annotation Stickies based on Bidirectional Links

Hiroyuki Sano, Taiki Ito, Tadachika Ozono and Toramatsu Shintani

Dept. of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
{hsano, taiki, ozono, tora}@toralab.ics.nitech.ac.jp

Abstract. We propose a web annotation system which adds the functionality of stickies to web pages and creates bidirectional links between the stickies. The stickies allow for important parts of a web page which contains large amounts of data to be highlighted. We have implemented a positioning method based on the Document Object Model (DOM), as well as a new method for placing stickies which depends on web contents related to the information referenced by the stickies. By using web agents, the system automatically generates bidirectional links between stickies referencing similar information and subsequently categorizes them. Such stickies and links can be used as user preferences, and have the potential to become a much better alternative to bookmarks and tags.

1 Introduction

We propose a web annotation system for web contents. Web pages contain text, images, and other types of information which are often related to more than one topic. We have implemented a web annotation system which enables users to place stickies on web pages. By using the system, users can point out specific contents more accurately than by using bookmarks in web browsers or current tagging systems. Bookmarks and tagging systems enable users to specify only the web page of interest. However, when users generate a link from a web page to a content located on another web page, they need to build a new system for managing the referenced web page.

The stickies provided by our system enable users to point out specific contents on web pages, as well as to generate bidirectional links between the stickies referencing the content. The position of the stickies in the system must be correspond to the relevant content in such a way. Related systems decide the position of the stickies by using absolute coordinates to equal the position of the stickies to the content. However, if the absolute coordinates are used, a problem occurs that a sticky is not displayed at the precise position of the information which a user references with the sticky, which in turn presents a problem when a user shares stickies with other users. We realize a new method for displaying stickies which ensures that each sticky is always displayed at the corresponding place.

An agent adds bidirectional links between the stickies in order to cross-reference similar contents in the system. The agent monitors the stickies which users place and generates bidirectional links between the stickies which were placed on similar contents.

2 Related Work

There are several annotation systems for web pages. Annotea[1][2], which was developed by W3C, is a framework which allows annotations to be placed on web pages. Annotea is available for Firefox through the Annotea Ubimarks extension, as well as for Amaya, which is an open-source web browser developed by W3C. However, although Annotea enables users to create links from one web page to other web pages, this needs to be done manually.

With regard to annotation, there also exist social tagging systems[3], where users add tags to web contents and share those tags with other users. However, to express relations between tags is difficult in current tagging systems, although tags are useful for specifying both web pages and web contents.

3 Web Annotation System

The web annotation system enables users to place stickies on web contents and to provide comments in relation to the content referenced by the stickies. Users can place annotation stickies on all types of contents on web pages, including text data, images, and so on, by using the web annotation system.

3.1 Outline of System

In the system, a web agent, which is referred to as a 'biLink agent', keeps track of the stickies which users have placed on web pages. The biLink agent is constructed from a page agent and a base agent, using the web agent model 'MiSpider'[4]. The page agent sends to the base agent the web content on which a user has placed a sticky. Then the base agent classifies the stickies by using information which it has received from the page agent, and generates bidirectional links between the stickies placed on similar contents.

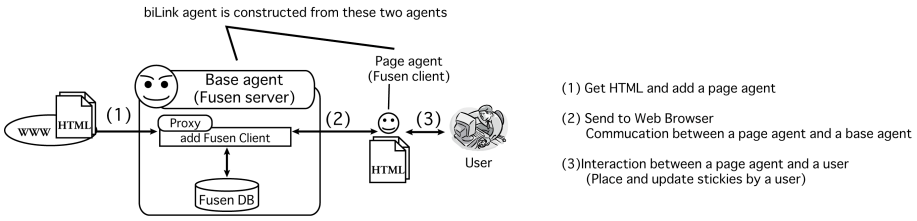

Fig. 1. The outline of the system

Figure 1 shows the outline of the system. The system comprises a 'Fusen client' and a 'Fusen server'.[1] The Fusen client, which is written in JavaScript, runs on the user's web browser and acts as a page agent. The Fusen client provides an interface for placing stickies on web contents. The Fusen server is the system which saves the stickies which users have placed. The Fusen server is a proxy server, which acts as a base agent, and a database for saving the properties of the stickies.

When a user requests a web page from the web annotation system, their web browser obtains the HTML source code of the web page from the proxy server, which contains the database with the properties of the stickies. The proxy server accesses the database and sends an inquiry regarding whether the HTML source code of the web page has been saved to the database. If the page has been saved, the proxy server sends the HTML code to the web browser. If not, the proxy server obtains the code from the relevant web server, adds a Fusen client to the HTML code, and sends the HTML code thus generated to the web browser. There are two methods for adding web services to an existing web page: one involves a proxy server[5], and the other is based on bookmarklets[6]. We adopted the proxy server method for the system. If a user places a new sticky or updates an existing sticky, the Fusen client updates the database on the Fusen server when the user leaves the web page.

3.2 Example of execution

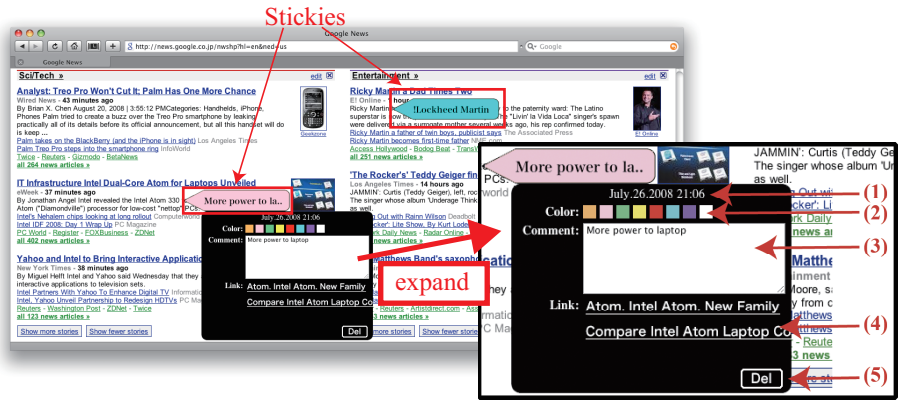

Fig. 2. An execution example of the system

Figure 2 shows a screenshot of a web page on which a user has used the system to place stickies. The browser is Safari, which has been developed by Apple, and the stickies were placed on two articles in Google News.

Users can place annotation stickies on web contents as shown in Figure 2. When a user double-clicks an annotation sticky, they can see detailed information

[1] Fusen means 'paper annotation sticky' in Japanese.

about the sticky in a dedicated popup window, which can be closed by double-clicking on the sticky again. In the example, the user has placed two annotation stickies on the page and has opened the popup window of the lower sticky. Regarding points (1) to (5), which are shown in the popup window in Figure 2, (1) indicates the date when the user placed the sticky, (2) is used when the user wishes to change the color of the sticky, (3) is a comment to the referenced content (this comment is also displayed on the sticky image), (4) shows links to similar contents (when a user clicks on a link, they can see the stickies which have been placed on similar contents), and (5) is used for deleting the sticky.

4 Deciding the place of a sticky Using a DOM tree

If an annotation sticky is displayed based on absolute coordinates, the sticky will not move when a window size or a font size of a web browser is changed in spite of the fact that the absolute coordinates of the content might change. As a result, the position of the sticky ceases to match the position of the content which the sticky refers to(e.g., Internote::Firefox Add-ons[2]). In order to avoid the problem, our system appends a HTML `<img>` tag to show a sticky image to the DOM node clicked by the user. By using the DOM-based method, both the sticky and the content it refers to are displayed at the same position even if the absolute coordinates of the content changes.

The method produces a new problem related to the fact that when users wish to place a sticky inside a very long text, the sticky does not appear at the desired place. In the system, if a DOM node where the user is attempting to place a sticky is a text node, the system divides the node into span nodes. If a text node is divided into span nodes that have only one character, the place where users can place a sticky can be chosen with an accuracy of one character, which enables users to place stickies practically anywhere on a web page. However, dividing text into span nodes of one character can create a new problem where the beginning of a new line might change drastically when compared with the unmodified web page. This is due to the differences in the implementation of rendering engines in web browsers. In our system, when users place a sticky inside a text, the system divides the text node into span nodes containing only one word. As the orthography of Western languages, such as English or French, demands a space to be left between words, the system can easily recognize words in such languages. Unfortunately, the orthography of Japanese does not leave a space between words, and therefore the system cannot easily recognize Japanese words. For this reason, the system needs to be able to recognize morphological units as words. The system performs a morphological analysis of a text node and divides the node into morphological span nodes. A page agent sends text on which the user has clicked to the Fusen server, which analyzes the text morphologically by using a Japanese Morphological Analyzer (MeCab[3]) and adds `<span>` tags between morphological units. The base agent then sends the divided

[2] https://addons.mozilla.org/firefox/addon/2011
[3] http://mecab.sourceforge.net/

nodes back to the page agent.The division of the nodes based on morphological units provides sufficient precision for placing stickies in the system.

5 biLink Agent

5.1 Bidirectional Links between Stickies

Weblogs implement a function called the 'trackback', which informs weblog authors about what kinds of web pages are linking to articles in the weblog. In this sense, the trackback feature makes weblogs bidirectional. In our system, a biLink agent generates links between stickies with a similar content based on the concept of trackback, and as a result users can traverse the stickies by using those links. We refer to the agent as 'biLink agent' and the links as 'bidirectional links'.

A biLink agent implements a function for automatic generation of bidirectional links between stickies. As mentioned in Section 3.1, a biLink agent constitutes a page agent and a base agent, which are used for keeping track of the stickies placed by the user. When a user places a sticky, the page agent extracts the text around the content where the sticky is placed by looking at the DOM tree. This process is based on heuristics, in other words, on the text around image files or flash files which describes those files[7]. The extracted text is sent to the base agent, which analyzes the text and classifies the sticky in accordance with a classification method explained in Section 5.3. Subsequently, the base agent automatically generates bidirectional links between stickies placed on similar content.

5.2 Operating principle of the biLink Agent

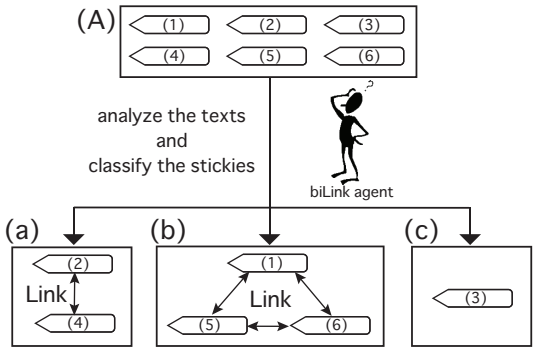

Fig. 3. Example of the operating principle of the biLink agent

Figure 3 illustrates the principle behind the biLink agent. The figure contains six stickies, (1) to (6), which the user has placed using the system. (A) indicates a universal set of stickies, and the biLink agent classifies those stickies. In Figure

3, the arrows indicate the process of classifying stickies. Let those six stickies be classified into three sets, (a) ={(2), (4)}, (b)={(1), (5), (6)}, and (c)={(3)}.

The biLink agent generates bidirectional links between stickies on the basis of the results of the classification. Set (a) contains two stickies, (2) and (4), and therefore the biLink agent generates bidirectional links between (2) and (4). Set (b) contains three stickies, (1), (5), and (6), and as a result the biLink agent generates bidirectional links between each pair of stickies (1), (5), and (6). As set (c) contains only one sticky, (3), the biLink agent does not need to generate any links.

5.3 Method for Classifying Stickies

Yang et al. examined some approaches to classify hypertext documents[8]. Glover et al. analyze the relative utility of document text, and the text in citing documents near the citation, for classification and description[9].

We present the method which the biLink agent uses to classify the stickies. The biLink agent uses MeCab to parse the web page containing the information on which the user has placed a sticky and decides the index terms of the web page. The biLink agent then uses the values of term frequency-inverse document frequency (TF-IDF) as evaluations of the index terms of the web page. The base document whose TF value is calculated by the biLink agent is the web page containing the information which the user has referenced with a sticky. Since the system is a web-based application, the biLink agent uses the total number of web pages which the Yahoo! API can search as the total number of documents, and the number of results which the Yahoo! API obtains appears as the number of index terms when the agent calculates the IDF.

The similarity between documents in classifying stickies is calculated by using a cosine measure based on the Vector Space Model. Each dimension of a document vector corresponds to a separate term, and each component corresponds to an evaluation of the term. However, the biLink agent performs the calculation by assigning a certain weight to the content which is referenced with a sticky. The term 'content' here indicates the nearest block-level element, where the tracing is in the direction from the node where the sticky is placed toward parent nodes.

Document vectors in the system are calculated by the following formula.

$$d_i = (w_{i1}, w_{i2}, \cdots, w_{iM})^T + \alpha(v_{i1}, v_{i2}, \cdots, v_{iM})^T$$

w_{ij} indicates an evaluation of term $t_j (j = 1, 2, \cdots, M)$ in a document number i, v_{ij} indicates an evaluation of term t_j in a content of a document number i, M indicates the number of different terms in a unit of documents, and α is the weight. Thus, by using document vectors calculated by assigning a certain weight to the content referenced with a sticky, the system can classify web pages containing multiple topics with a high degree of accuracy.

Next, the cosine measure is calculated by the following formula using two document vectors, d_1 and d_2.

$$cos(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1||\ ||d_2||}$$

Here, the lower the degree of the two vectors, the larger the cosine measure. When the system classifies stickies, a new cluster is created in which the average vector of document vectors present in the cluster is taken as the cluster vector.

```
input:
    a new sticky placed on a contents
        fusen
    existing clusters
        G = {g1, g2, .., gn}
    document vectors of existing clusters
        D = {d1, d2, .., dn}

procedure ClassifyStickies (fusen, G, D)

01. begin
02.   df ← the document vector of fusen
03.   // the number of existing clusters is zero
04.   if n == 0 then
05.     // add fusen to the new, first cluster
06.     g1 ← {fusen}
07.     d1 ← df
08.     G ← {g1}
09.     D ← {d1}
10.   else
11.     max ← −∞
12.     s ← 0
13.     // find the largest cosine measure
14.     for each di ∈ D do
15.       if max < cos(di, df) then
16.         max ← cos(di, df)
17.         s ← i
18.       endif
19.     enddo
20.     if max > threshold then
21.       // add to the cluster
22.       // update the document vector
23.       gs ← gs ∪ {fusen}
24.       ds ← the average vector of gs
25.     else
26.       // add fusen to a new cluster
27.       gn+1 ← {fusen}
28.       dn+1 ← df
29.       G ← G ∪ {gn+1}
30.       D ← D ∪ {dn+1}
31.     endif
32.   endif
33. end.
```

Fig. 4. The procedure for classifying

Figure 4 outlines the procedure used by the biLink agent to classify stickies. The biLink agent analyzes the content on which the user has placed the first sticky, calculates the document vector, and generates the first cluster which contains only the first sticky. After that, as more stickies are placed on parts of the page with different contents, the biLink agent analyzes the content in those parts, and calculates the document vectors as well as the similarity between the document vectors and the document vectors of the clusters which already exist. If the similarity is greater than a predefined threshold, the biLink agent adds the sticky to the cluster and updates the document vector of the cluster with the average vector. If the similarity is lower than the threshold, the biLink agent generates a new cluster and adds the sticky to the new cluster.

The weight α and the similarity threshold are decided on the basis of the results performed by a person. A certain number of web pages are collected at random and classified manually, after which the agent also classifies them. In order to match the results of the manual classification with the results of the classification performed by the agent, we adjusted α and the threshold. Eventually, the most satisfactory level of conformance was attained when α was 25 and the threshold was 0.15.

If a sticky is classified into an existent cluster, the biLink agent generates bidirectional links between the sticky and all other stickies in the same cluster, and the final result is that the bidirectional links form a complete graph.

6 Experimental Result

6.1 Speed of Placing Stickies

We measured the change in the processing time in relation to the increase of the text length. More specifically, we evaluated amounts of text data in the range of 200 bytes to 2000 bytes, where the step of increase was at the rate of 200 bytes. We placed stickies 10 times on each text and measured the time from when the mouse button was clicked to when the web browser received the results of the analyzed morphological parts from the Fusen server. We also measured the time from when the mouse button was clicked to when the sticky was displayed.

In order to avoid the influence of inherent network delays, the base agent and the page agent were executed on the same computer. In other words, the Fusen server and the web browser were running on the same computer. The relevant specifications of the computer system used in the experiment are outlined below.

- CPU: Intel Core Duo 1.83GHz
- Memory: 1.5GB
- OS: Mac OS X 10.5.3
- Web Browser: Safari 3.1.1

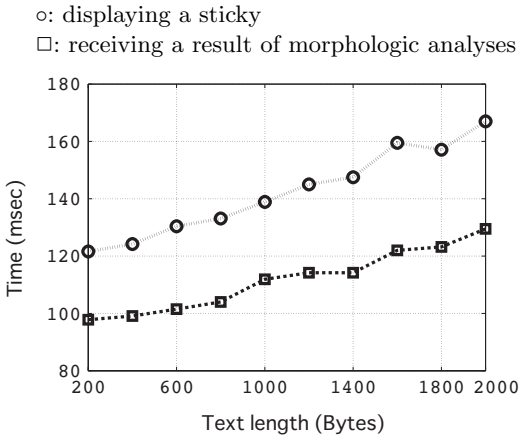

Fig. 5. Processing time of the proposed method

Figure 5 shows the processing time for the morphological analysis and for displaying the sticky. The horizontal axis shows the text length in bytes, while the vertical axis shows the processing time. The graph plotted with ○ (upper graph in Figure 5) is the processing time needed for displaying the sticky, and the graph plotted with □ (lower graph in Figure 5) is the processing time of the client receiving the results of the morphological analysis from the server.

Figure 5 shows that the longer the text length, the longer the processing time. However, the fact that the system processes the information in 170 ms when the text length is 2000 bytes is strong proof that the method is very fast.

Since the content used in the system is part of a web page, a text length of 2000 bytes is sufficiently long for practical purposes.

Thus, the experiment shows that stickies can be placed on the page very quickly, and that the proposed method has a potential for practical use.

6.2 Bidirectional Links

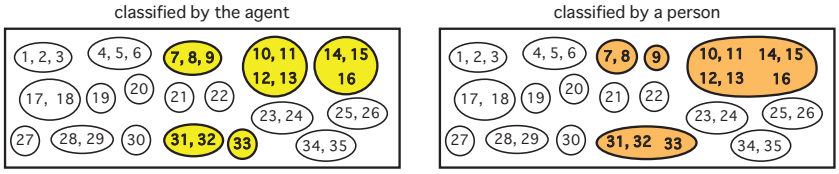

Fig. 6. Results of the classification by the biLink agent (left) and a person (right)

In order to check that the agent generates correct bidirectional links, we evaluated the classified stickies. We placed 35 stickies on different parts of web pages with multiple contents categories at random, and compared the results of the manual classification and that performed by the agent. All web pages used in the experiment were written in Japanese.

Figure 6 shows the results of the classification. The left image in Figure 6 presents the results of the classification performed by the agent, and the image on the right corresponds to the manual classification. Each number (1 to 35) in Figure 6 represents an identification number of a sticky. Stickies which are located inside the same circle have been classified into the same cluster.

The results of the classification performed by the agent are remarkably similar to those of the manual classification. The agent classifies the 35 stickies into 18 clusters, and 13 clusters are exactly the same as the clusters classified by a person, which amounts to an accuracy of 72.2%. In addition, the 5 clusters which differ in their classification still closely resemble the clusters classified by a person.

7 Summary and Future Work

We implemented a system which allows the placement of annotation stickies on web pages by using a web agent. The system was implemented using a DOM-based positioning method for placing the stickies. In the method, stickies correctly point to the specific information which the user wishes to mark as important, and the experimental results show that the method is sufficiently fast for practical purposes.

In the system, users can generate links to any type of contents in web pages without editing the web pages themselves. In the past, when users collected information from many web pages, they had to cut the relevant information and paste it into a new web page. However, the cut-and-paste method is prone to copyright infringement issues. By using our system, generating links to the

content of interest enables users to browse the contents without the need for copying and pasting information. Therefore, users can freely handle and reference different types of information, which has the potential to make the Internet even more useful with respect to searching for information[4].

A web agent in the system automatically generates bidirectional links between annotation stickies in order to show the relations between the stickies which reference similar contents. Users can traverse the referenced content in a two-way fashion by using the links, thus coming across previously unknown web pages. Furthermore, bidirectional links are useful in that they can contain more information than social tags. The next step of the research is to study the concept of new search engines and contents recommendation systems which are based on information provided by annotation stickies.

One issue related to sharing stickies involves the fact that the biLink agent must be given explicit permission to share stickies in order to protect the privacy of the user. Therefore, a new protocol defining the process which biLink agents must follow when sharing stickies with other biLink agents needs to be studied, and must include the allotment of privacy protection schemes.

References

1. Koivunen, M.R.: Annotea and semantic web supported collaboration. Invited talk at Workshop on User Aspects of the Semantic Web (User-SWeb) at European Semantic Web Conference (May 2005)
2. Kahan, J., Koivunen, M.R.: Annotea: an open rdf infrastructure for shared web annotations. In: WWW '01: Proceedings of the 10th international conference on World Wide Web, ACM Press (May 2001) 623–632
3. Tennis, J.: Social tagging and the next steps for indexing. In: Proceedings of the 17th SIG/CR Classification Research Workshop. (Nov 2006)
4. Fukagaya, Y., Ozono, T., Ito, T., Shintani, T.: Mispider: a continuous agent on web pages. In: WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web, ACM Press (May 2005) 1008–1009
5. Sakamoto, S., Kita, H., Takase, H., Hayashi, T.: Platform for web services using proxy server. The Special Interest Group Notes of IPSJ **2002**(31) (2002) 7–12
6. Tanabe, M., Ozono, T., Itoh, T., Shintani, T.: Web servise addition system by bookmarklet using user browsing domain. In: Proceedings of the 68th National Convention of IPSJ. (Mar 2006)
7. Chen, Z., Wenyin, L., Zhang, F., Li, M., Zhang, H.: Web mining for web image retrieval. Journal of the American Society for Information Science and Technology **52** (2001) 831–839
8. Yang, Y., Slattery, S.A.: A study of approaches to hypertext categorization. Journal of Intelligent Information Systems **18** (2002) 219–241
9. Glover, E.J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D.M., Flake, G.W.: Using web structure for classifying and describing web pages. In: WWW '02: Proceedings of the 11th international conference on World Wide Web, ACM Press (May 2002) 562–569

[4] We believe that proxy servers do not violate copyrights