ISSN 1346-5597

NII National Institute of Informatics

NII Technical Report

Aikuchi: Marking-based Social Navigation System

Yuki Matsuoka, Ryuuki Sakamoto, Sadanori Ito,
Hideaki Takeda, Kiyoshi Kogure

NII-2008-001E
Feb. 2008

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

Aikuchi: Marking-based Social Navigation System

Yuki Matsuoka⋆◇‡ m-yuki@nii.ac.jp
Ryuuki Sakamoto‡ skmt@atr.jp
Sadanori Ito♯‡ sito@cc.tuat.ac.jp
Hideaki Takeda ⋆◇‡ takeda@nii.ac.jp
Kiyoshi Kogure‡ kogure@atr.jp

⋆The Graduate University for Advanced Studies, Shonan, Hayama, Kanagawa 240-0193 Japan
◇National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
‡Advanced Telecommunications Research Institute International, 2-2-2 Hikaridai, Seikacho, Soraku-gun, Kyoto 619-0288 Japan
♯Tokyo University of Agriculture and Technology, 3-8-1 Harumi-cho, Fuchu-shi, Tokyo 183-8538 Japan

Abstract

In this paper, we propose a social navigation system called "Aikuchi", which enables users to mark a string on a Web page and share it with others in a community. If users mark a string on their Web browser using their cursor, Aikuchi recommends links to other Web pages that were related to this string based on recommendation algorithms. When a user clicks a recommended link, Aikuchi automatically highlights the marked string as a link anchor. We offered this system as part of a conference support system. According to an analysis of the user logs, users preferred Web page recommendations based on the strings marked by other users to those based on similarity-based recommendation and collaborative filtering. As a result, we think that marked strings are appropriate for social navigation.

Keywords

Social Navigation, web browsing, marking

1. INTRODUCTION

In the physical world we use our interactions with others to guide our actions, to make choices, and to find things of importance or interest. For instance, when we are looking for a book in a library, we might see that one of the books on the shelf has more note marks or dog-eared pages than the others. This might suggest that lots of people have read it, and we may decide to check it out. Thus, the physical world is a social place and people often rely on social interactions to find information.

On the Web, we often spend time trying to find information and often get frustrated during the search process, since there are so many web pages. People use the web to find information all the time and their experiences are rarely captured and used to guide or inform others. Although, in a social navigation system, such as amazon.com [1], when users access a page for a item, they can get related items based on a history that others have purchased the requested item from. The term "social navigation" was originally introduced by P. Dourish and M. Chalmers [4], to describe how a user's navigation through an information space is guided and structured by the activities of others within that space. The social navigation system enables users to take advantage of others' activities, become aware of others' interests and knowledge, get recommendations of relevant information based on others' opinions, and use web navigation as a kind of communication channel [2]. The social navigation system promotes knowledge and information exchange among users on the web.

We propose the social navigation system called "Aikuchi", which uses marking as the user interaction tool and recommends web pages based on it. It also provides interfaces for smooth navigation. In the following sections, we will introduce our system, describe its architecture and implementation, and present the findings from an analysis of the user logs.

2. OUTLINE OF THE SYSTEM

Some social navigation systems use an accounting of how many users pass through a link or visit a page and then recommend the most popular links and pages. The *Footprints* system [11] records the history of all users as they navigate on the Web. It presents different visualizations of the history information as maps, trails, and annotations allowing users to see where activity had taken place within a given page. IBM's WBI toolkit [7] observes people's paths through the Web and looks for recurring paths. However, we use marking as the users' interactions for page recommendations in our system, which we call Aikuchi. When people read books or literature, they often mark strings in pages that they found interesting. The marked strings in a page are different for each user, since they have different viewpoints. If they share their marked strings on pages, they may discover pages that they weren't able to find on their own. Our system enables users to mark strings in Web pages and share them with others. If a user marks a string in their Web browser using their cursor, Aikuchi recommends links to other Web pages based on shared marked strings, and when a user clicks a recommended link, the system shows a specified page and highlights the marked string. Users can move from the marked page to the other page smoothly using this interface. The highlighted string is called a footprint and works as a link anchor that can jump to other web pages. When users access a web page that includes footprints, they can get other pages using them. The system navigates to other web pages when users mark a string on a web page or places their cursor over the footprints.

2.1 How it works

We designed Aikuchi to be used at a conference. A lot of presentation pages, along with a schedule, is normally provided on conference web sites. When conference participants seek information concerning presentations, they often click links to presentations from the schedule page. If there are many presentations, they often have to browse through these pages, or may not find papers. In many cases, a community consisting of conference participants who are interested in the same field exists. If they share knowledge in this community, they may find the appropriate presentations. We offered Aikuchi as a part of a support system at the Japanese Society for Artificial Intelligence 2006, which was held from June 7-9, 2006. This support system was operated as a Web system, and every conference participant could access it using a user ID and password. The target Web pages were comprised of 276 pages that included the authors' names and abstracts for the papers. The purpose of the system was to make it easier for users to find desired pages based on the marked strings.

The details of the system's operations are as follows: Users can mark a string in their Web browser using their cursor when they find some interest on a page (Fig. 1). Then a recommendation window pops up to display any recommendations from other pages called recommendation links (Fig. 2). If the user selects a recommendation link, the system shows the specified page, and once a recommendation link is selected, the system highlights the used string. We call this highlighted string a footprint (Fig. 3). It is shared; i.e., users can see the footprints added by all other users. If the user places her or his cursor over the footprint, the recommendation window pops up again. Figure 4 shows an example of the recommendation and footprint links. The latter includes the Web pages to which some users have jumped to from the marked string.

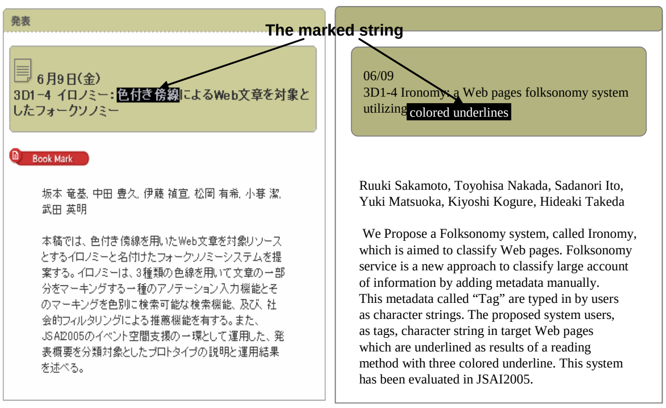

Fig. 1: *Marking on Web browser*
The left figure indicates the title, authors, and abstract for a paper. The right figure is a translation of the information on the left.

Fig. 2: *Popped-up recommendation window*
The right figure shows the translation of the recommendation links.

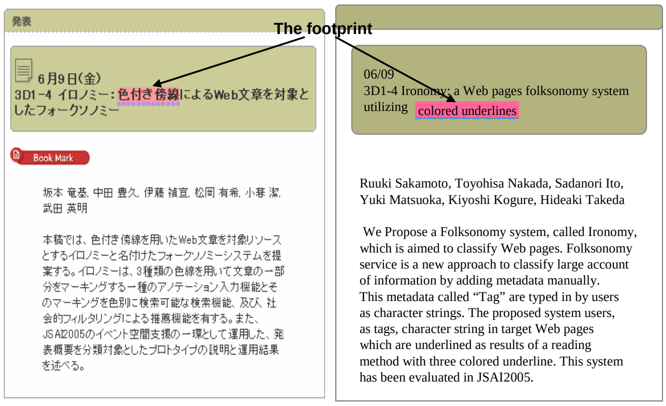

Fig. 3: *Example of Footprint*
The right figure is a translation of the information on the left.

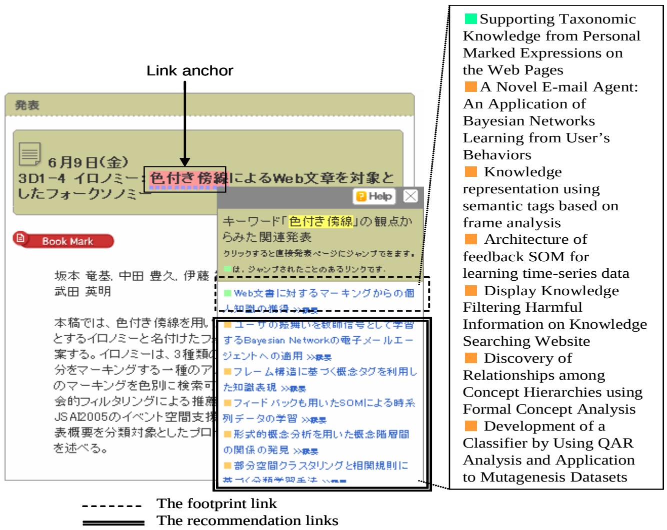

Fig. 4: *Recommendation window after cursor is placed over footprint*
The right figure shows a translation of the footprint and recommendation links.

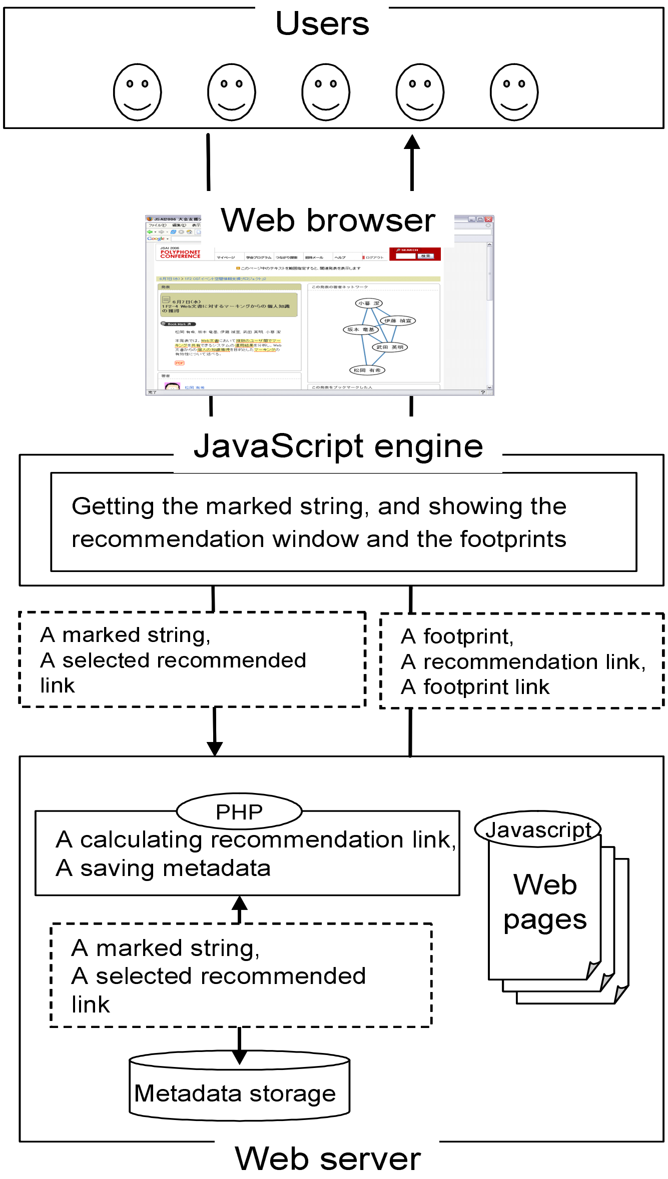

Fig. 5: *System structure*

2.2 Implementation

Figure 5 shows the system's structure. The system runs as a script on the user client site and on the Web server, and is implemented with JavaScript and PHP. We prepared web pages for papers with JavaScript on the web server. When a user accesses a Web page, the JavaScript engine asks the Web server if a footprint exists, and shows it if it does.

When a user marks a string in the Web browser, the JavaScript engine obtains the marked string and sends it to the Web server, and then the web server takes the marked string and calculates the recommendation links according to certain algorithms. Following the calculation, the Web server sends the recommendation links and the JavaScript engine shows the recommendation window to the user. When the user selects a recommended link, the JavaScript engine sends it and the Web server saves the following information related to the selected link as a text file in the metadata storage.

- Date
- User ID
- Marked string
- Position of the marked string on the Web page
- URL of the selected recommended link
- Selected recommendation algorithm

When a user places his or her cursor over a footprint, the JavaScript engine takes the footprint's string and sends it to the Web server. The Web server gets this string, calculates the recommendation links, and obtains the footprint links from the metadata storage. The Web server then sends both the footprint links and the recommendation links. Next, the JavaScript engine again displays the recommendation window to the user and if the user selects a footprint link or a recommended link, the JavaScript engine sends it and the Web server saves the information about it. Since we use Ajax asynchronous communication for the exchanges between the user client and the web server, users can get information smoothly without showing another page.

2.3 Recommendation algorithms

When a user marks a string or places the cursor over a footprint on a Web page, the system shows the recommendation links using four types of algorithms that explore Web pages to find useful pages based on the marked strings. For comparison, we prepared four recommendation algorithms; algorithm A is a similarity-based recommendation, algorithm B is a collaborative filtering, algorithm C is a marking-based matching, and algorithm D is a query-based matching. We will now describe each recommendation algorithm

A) Page similarity using TFIDF

B) Collaborative filtering using the number of footprints on a Web page

C) Word matching between a marked string and footprints' strings

D) Word matching between a marked string and the Web pages' strings.

In algorithm A, we used calculated page similarity based on TFIDF [10]. At first, we get words from each web page using Chasen [8] (Japanese morphological analysis system) and calculated the tfidf value t_{ij} for each word.

$$t_{ij} = tf\,(i,j) \log \left(\frac{N}{df\,(j)} \right)$$

where $tf\,(i,j)$ is the frequency appearance of words w_j in a web page, A_i and $df\,(j)$ is the frequency appearance of web pages including w_j, and N is the total number of web pages. In addition, we calculate the web page vectors $\vec{A_i}$ using the TFIDF value.

$$\vec{A_i} = (t_{i1}, t_{i2}, , , t_{ik})$$

where k is the total number of words in all web pages. So, we calculate the page similarities $simi_{i,j}$ between two page vectors $\vec{A_i}, \vec{A_j}$.

$$simi_{ij} = \frac{\vec{A_i} \cdot \vec{A_j}}{|\vec{A_i}||\vec{A_j}|}$$

If a user marks a certain string on a Web page, the system recommends high-similarity pages to it regardless of the marked string and footprints.

In algorithm B, the system recommends pages with collaborate filtering [9], using the number of footprints on a Web page as the users' evaluation of that page. It calculates the predictive value by using the following collaborate filtering. We use an algorithm based on the nearest neighbor method. The user similarity is calculated by using the Pearson's correlation coefficient between the evaluation values. The similarity $S_{a,u}$ between users a and u is as follows.

$$S_{a,u} = \frac{\sum_{i=1}^{I}(r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i=1}^{I}(r_{a,i} - \overline{r}_a)^2 \sum_{i=1}^{I}(r_{u,i} - \overline{r}_u)^2}}$$

where $r_{a,i}$ is the evaluation value of a web page by user a, $\overline{r}_a$ is the average value of the evaluation value by user a, and I is the total number of web pages. In addition, the system selects the neighboring n users with a high similarity to the user selected string. It calculates the predictive value $P_{a,i}$ using the evaluation value of the neighboring users.

$$P_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \overline{r}_u)S_{a,u}}{\sum_{u=1}^{n} S_{a,u}}$$

where $P_{a,i}$ is the predictive value to a web page i by user a and $S_{a,u}$ is the similarity between users a and u, and n is the number of neighboring users around user a. Therefore, the system recommends Web pages with a high predictive value using the number of footprints.

In algorithm C, if a word in the marked string matches one in the footprint's string on other pages, the system recommends the matched page.

In algorithm D, if a word in the marked string matches one in another page's string, the system recommends the matched page.

Table 1 shows a comparison of the recommendation algorithms with respect to whether they use a marked string and footprints.

Recommendation algorithm	Marked string	Footprints
A	×	×
B	×	○
C	○	○
D	○	×

Table 1: *Comparison of recommendation algorithms*

Algorithm A makes recommendations based on the page similarity between Web pages without using a marked string or footprints. Algorithm B, meanwhile, uses the number of footprints on a Web page to look for neighboring users. This is assumed to be an ordinary recommendation. Algorithms C and D use a marked string as a query to search for Web pages with matching words. The difference between them is the search target: footprints' strings (Algorithm C) or a Web pages' strings (Algorithm D). The system recommends up to two Web pages for each algorithm and presents them in random order. Algorithm priority was applied so that the same link was calculated in the order of C, D, B, then A. The system did not inform users about these algorithms. When users clicked a recommended page link, we determined that they preferred the respective recommendation algorithm.

3. ANALYSIS

After completing the experiment, we had obtained 324 footprints and 172 links. An analysis of the results revealed that there were 45 users who marked strings one or more times, 28 users who jumped from marked strings to other pages, 88 users who placed their cursors over footprints one or several times, and 33 users who jumped from footprints to other pages. In this section, we describe which algorithms were preferred by users when a user marked a string or placed their cursor over a footprint on a Web page.

3.1 Algorithm preference

When users clicked recommendation links, the system linked from the marked strings in the pages to other pages. We investigated the created link structure by using each algorithm. The link structure of algorithm B consisted of one cluster that all the pages linked (Figure on the right in Fig. 6) to. Since the cluster was linked from the pages to specific pages, we found that the same pages were recommended by algorithm B and that the users didn't like these recommended pages. Since a given user marked strings on the same page many times, the system recommended the same pages. Otherwise, algorithms A, C, and D consisted of several clusters. (Figures on the left in Figs. 6 and 7). The pages recommended by these algorithms had no problems.

Surrounded pages by line include out of three pages.

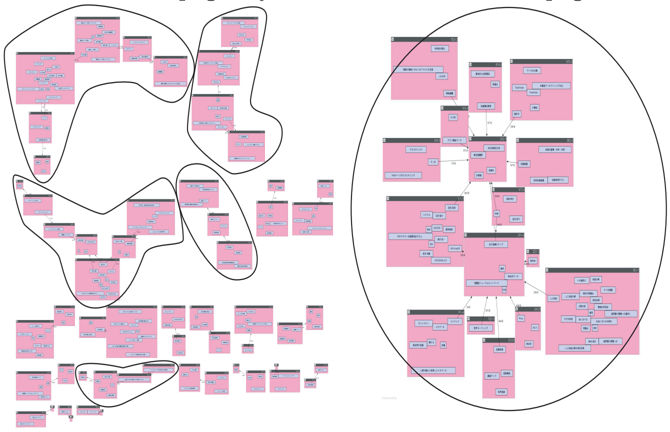

Fig. 6: *Link structure by algorithms A and B*

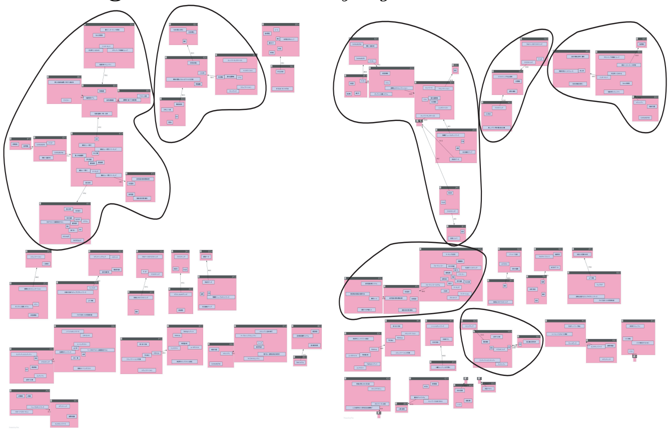

Fig. 7: *Link structure by algorithms C and D*

Figure 8 shows a comparison of the number of selected recommendation algorithms when users marked a string on a Web page. Before the conference, algorithm A was most commonly selected and there were fewer selections when using B and C because there were fewer footprints at the beginning of the conference (Fig. 9). The number of selections for D was fewer than that of A before the conference. This means that users preferred the recommended links based on their similarity without relation to the marked strings or footprints. As the number of footprints increased during the conference, the number of selections from C and D increased and the number of selections from B decreased, because if Aikuchi recommended the same page links over and over again, then the users may have tired of selecting the links recommended by B. The number of selections for algorithm A was also lower during the conference. This means that users preferred the recommendation algorithms based on the word in the marked string. Therefore, we found that over time users preferred recommendations based on words rather than similarity. We also found that footprints are effective for recommending links from the fact that the number of selections for algorithm C increased.

3.2 Footprints

When users place their cursor over a footprint, they could obtain not only the recommendations from the algorithms, but also the pages that a previous user had jumped from this marked string to before. We investigated which algorithms were preferred by the users in such cases (Fig. 10). Algorithm Z denotes the selection of jumped pages by users. Although the number of recommended links was fewer for algorithm Z than for algorithms A or D before the conference (Fig. 11), Fig. 10 shows that algorithm Z had many selections, since many users tried to use the system at first. The number of selections for algorithm B was fewer during the conference than before it. This means that users would become tired of selecting the links recommended by algorithm B as time passes. During the conference, algorithm Z was also most selected. Therefore, we found that users preferred to select links by using footprints.

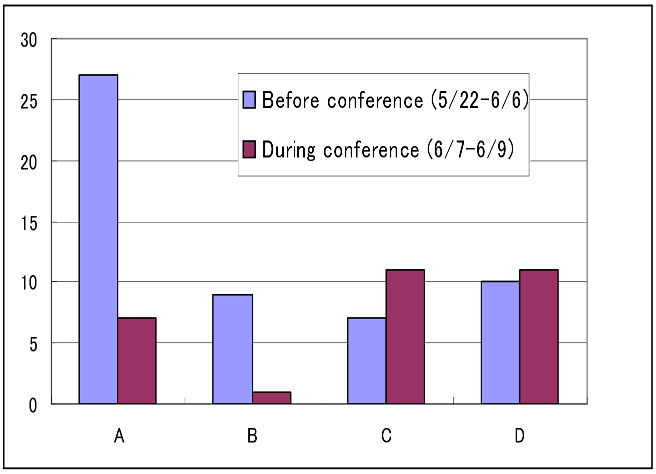

Fig. 8: *No. of selected recommendation algorithms, when users marked a string on a Web page.*

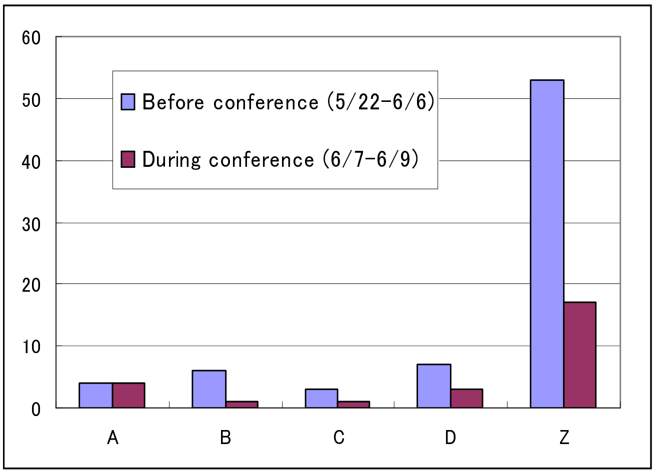

Fig. 10: *No. of selected recommendation algorithms, when users placed their cursor over a footprint.*

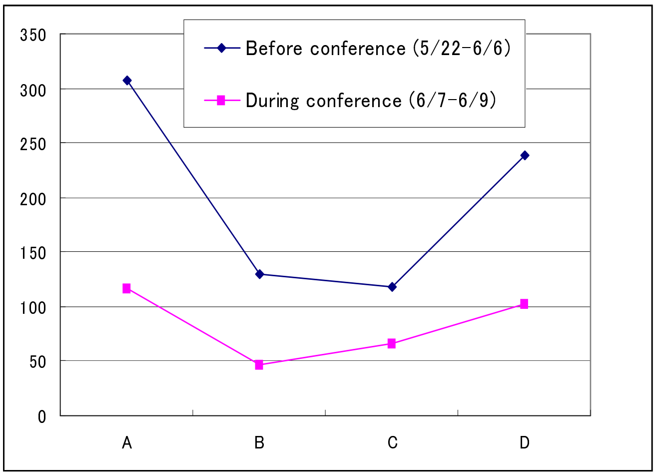

Fig. 9: *Recommended number by each algorithm, when users marked a string on a Web page.*

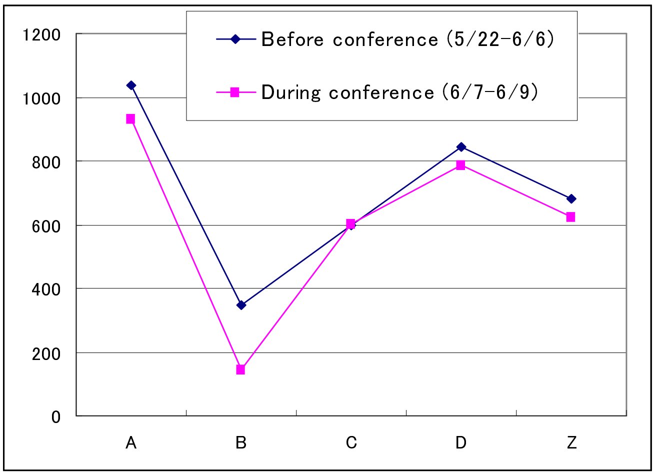

Fig. 11: *No. of selected recommendation algorithms, when users placed their cursor over a footprint.*

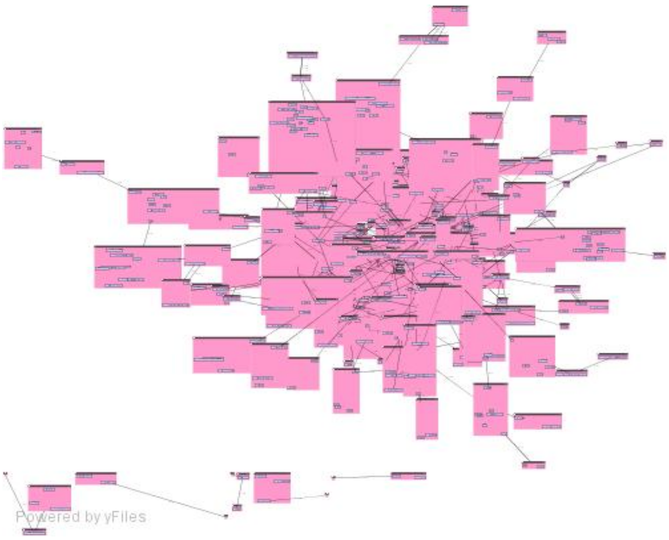

Fig. 12: *Link structure by algorithm A,B,C and D*

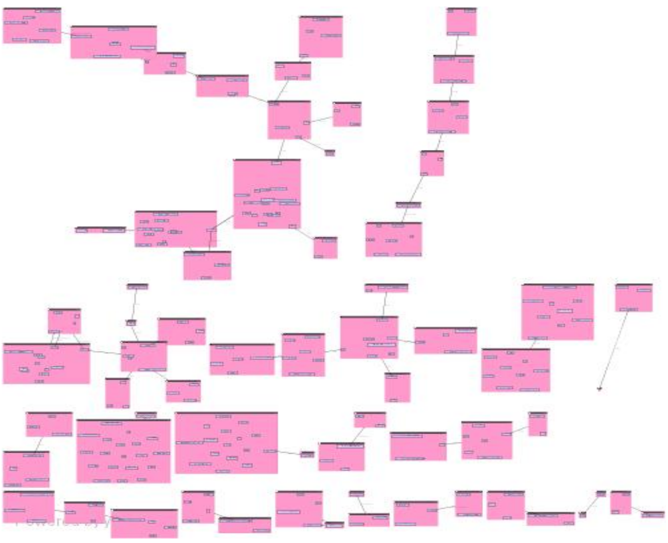

Fig. 13: *Link structure by algorithm Z*

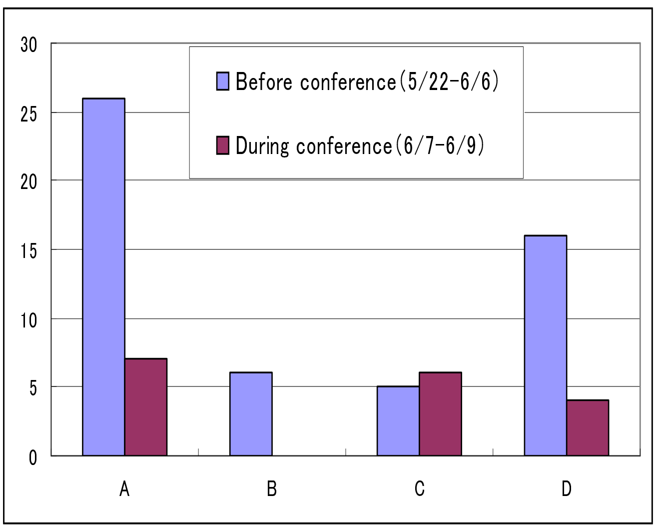

Fig. 14: *Number of selected recommendation algorithms in Z*

Figure 12 shows the link structure by algorithm for A, B, C, and D. Since there is a big cluster, we found that Aikuchi users had similar interests. Figure 13 shows that the link structure by algorithm Z consists of some clusters. It consists of the link structures by algorithms A, B, C, and D. We also investigated which recommended algorithms were preferred in the selection of algorithm Z (Fig. 14). Before the conference, the number of recommended algorithms was the same as when the users marked a string on a Web page. Otherwise, the largest number of recommended algorithm came from algorithm A during the conference. Since Fig. 11 shows that the numbers of recommended algorithms by algorithm C were fewer than it by algorithm A during the conference, users preferred those made by algorithm C.

4. RELATED WORK

There are some social navigation systems that use annotation. Walden's Paths [6] is a pedagogical application for social trail mapping. It allows teachers to provide additional context for pages through annotation. By providing text or other annotations in addition to the content of the page, the teacher may provide a rhetorical structure to the path as a whole, create transitions to fill in any informational gaps between pages, and create emphasis to particular aspects of the materials. Thus, students and teachers can associate the web pages in a linear fashion in order to create a path through the information space of the Web. Our system links from marked strings in pages to other pages.

CoWeb [3] annotates all links inside it with activity markers. These markers indicate when the page behind the link was last modified and also whether it was recently accessed. The last point is a major change in the user experience, because users leave traces of their activities in a CoWeb even if they are not modifying the content. Simply looking at a page causes activity markers to appear. In addition, CoWeb provides access to the log file and allows users to check for recent activity in the CoWeb on the whole without causing activity markers to appear. The annotation indicates a history of the page itself, not of the link. This point is different from our system.

Knowledge Sea [5] encourages users to annotate pages they are reading in the form of writing notes or highlighting parts of the page that they find important. These annotations appear to users as icons, which have different format: a thumbs up, a question mark, or a sticky note. Users use these as implicit indicators of a page's relevance. Our marking is for navigating to other pages.

5. CONCLUSION AND FUTURE WORK

We proposed using marking to note users' interactions for recommending pages, and developed a system called Aikuchi with which users can mark strings and share them as footprints. As our footprints work as link anchors, users can get information just to see the web page that includes them. Aikuchi recommends links based on a variety of algorithms when a user marks a string on a Web page. Based on an analysis of the user logs, we found that users preferred recommendations based on words in marked strings rather than page similarity. We also found that footprints are useful for users to select links, because links based on footprints are frequently selected by users. Since the algorithm Z was selected most, we were able to find that footprints were effective in leaving a trace that moved to other pages. Consequently, we believe that strings marked by users on Web pages can be useful when users have the same interests or purposes in a community.

We proposed a simple social navigation system based on marking. Users share all their marking data and get page recommendations based on same recommendation algorithms. Otherwise, users will need personal navigation. We'd like to find user models from their marking activities in order to recommend the appropriate pages and show only the appropriate footprints.

Acknowledgment

This research was supported by the National Institute of Information and Communications Technology.

References

[1] Amazon.com. http://www.amazon.com/.

[2] M. Barra, P. Maglio, A. Negro, and V. Scarano. Gas: Group adaptive system. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 47–57, 5 2002.

[3] A. Dieberger and M. Guzdial. Coweb: Experiences with collaborative web spaces. In *Springer-Verlag*, pages 155–166, 2003.

[4] P. Dourish and M. Chalmers. Running out of space: Models of information navigation. In *Proceedings of the Conference on Human Computer Interaction*, 8 1994.

[5] R. Farzan and P. Brusilovsky. Social navigation support through annotation-based group modeling. *Lecture Notes in Computer Science*, 3538:463–472, 2005.

[6] R. Furuta, I. Frank M. Shipman, C. C. Marshall, D. Brenner, and H. wei Hsieh. Hypertext paths and the world-wide web: experiences with walden's paths. In *HYPERTEXT '97: Proceedings of the eighth ACM conference on Hypertext*, pages 167–176, 1997.

[7] P. Maglio and R. Barrett. Intermediaries personalize information streams. *Commun. ACM*, 43(8):96–101, 2000.

[8] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi, and T. Imamura. Japanese morphological analysis system chasen manual. *NAIST Technical Report NAIST-IS-TR97007*, 1997.

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[10] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[11] A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 270–277, 1999.