Bookmark-Agent: Information Sharing of URLs

Mikihiko MORI*and Seiji YAMADA*

November 28, 1998

Abstract

We propose a bookmark-agent system that shares information in bookmark files, which are registered user's preference URL addresses. The bookmark-agent we studied is more effective in searching information than existing search engines such as Yahoo, AltaVista and so on. When an user tries to find for certain URLs (i.e. the Web pages concerned with artificial intelligence) by browsing in the WWW, the agent can search for Web pages which he or she wants. Furthermore the agent requests for other agents to search their own bookmarks. As a result, the user can obtain similar pages by his agent as hyperlinks on a Web browser. Since the acquired information is filtered beforehand by users, the Bookmark-agent is able to output more precise Web pages than a search engine. We finally made experiments by six users, and found out the Bookmark-agent is a promising approach to share URLs in a small community.

1 Introduction

Growing rapidly of the internet, it is difficult to find particular pieces of information in the World Wide Web (WWW). Search engines have been developed. For search engines, an user gives keywords, then the search engines

*CISS, IGSSE, Tokyo Institute of Technology

can search Uniform Resource Locators (URLs)[3] suiting users' preference. But it often happens that the URLs were presented too much to see all.

In this paper, we propose a system named Bookmark-agent. The agent uses user's bookmarks to get user's interest. Bookmarks are a file for gathering favorite URLs of site which the user visited. In a small user group having the same interest, the bookmarks contains filtered information about URL, as it indicated their preference. The group have been gathered beforehand.

An Bookmark-agent assists an user to search sites which resembles a Web page looked by the user now. The agent refers user's bookmarks and others by communication between agents if their users allowed to open to public.

An user need not give his agent keywords because the agent can see a Web page looked by the user. The user can also explicitly give keywords if he or she wants.

Note that Bookmark-agent gives its user a smaller number of URL and more significant URL than ordinary search engines. Because the agent utilizes bookmarks which are filtered information of the WWW by users and the agents are only used in a small group having same interest.From this, it means that users' interest narrows the number of presenting URLs by keywords. Thus this system hopes for filtering more than ordinary search engines.

Siteseer[11] utilizes each user's bookmarks as an implicit declaration of the interest in the underlying content. It is difference from Bookmark-agent that *Siteseer* treats each user's folder in which it measures the degree of overlap URLs with other people's folders.

ShopBot[5] can parse product description and identify several product attributes, including price and operating system, for the products in HTML. *ShopBot* learns a vendor description for each merchant.

WebWatcher[1][7][8] indicates the Web pages which an user wants to see next as browsing in the WWW. *WebWatcher* observes and learns from its users' actions. The system differs in several key respects keyword-based search engines. It can learn that a term match a hyperlink even though these phrase share no words in common.

Letizia[9] also indicate the next Web pages while browsing. The system almost resembles *WebWatcher*. However *Letizia* is different from *WebWatcher* in terms that Letizia is located on a single user's machine and learn his current interest.

Syskill & Webert[10] learns user's interest from manual collected file as profile and can request LYCOS[1] to search. The system learns a separate profile for each topic of each user. The user can rate a page as is between hot and cold.

WebMate[4] learns user interest incrementally and automatically sends mails that match user interests. Moreover, it adds keywords from user profile for query to search for URLs of interesing sites. On the one hand WebMate monitors user browser to get user profile, on the other Bookmark-agent make profile bookmarks.

Beehive[6] designs a distributed system for social sharing and filtering of information on email system. Information of interest by an user is sent to members of registered group through Beehive system. The member who does not communicate is left from member list if frequeny of messages is lower than an threshold. Content of message is no concern of this system.

2 Bookmark-agent

This section describes the design of Bookmark-agent and how it assists its user in their search for information.

2.1 The Functions

Fig. 1 shows the architecture of the Bookmark-agent system. An Bookmark-agent is invoked per one user and then assists the searching for information in the WWW. Functions of Bookmark-agent are as the followings:

- **Extracting keywords from a new URL**: The agent monitors user's bookmarks. If a new URL is added, it will extract keywords from the Web page of the URL, and will store the keywords with title of the page and the URL in the keyword-database which the agent have.

- **Indicating URLs of similar pages**: The agent sugests URLs of similar pages in the WWW to its user if the user browses a new Web page following a hyperlink, or if the user explicitly gives keywords to it. In this system, the new page was obtained from its site through the internet, then the keywords were extracted from the page into keyword-database.

[1] `http://www.lycos.com/`

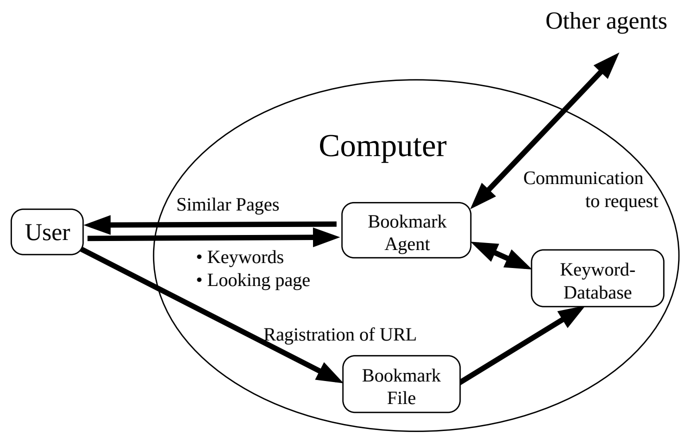

Figure 1: The architecture of the Bookmark-agent system

Here, we define a *similar page* as a page whose similarity is up to a threshold. The threshold can be changed by an user when too many/few shown URLs are indicated.

Searching for similar pages needs the following process:

- The agent searches the user's keyword-database.
- It communicates and requests to other agents in order to search for other users' keyword-databases. Replying search request can permitted each URLs by each users before the agent is invoked.

- **Accepting a request for search**: If the agent receives a request which another agent send for searching similar pages, it must search the user's keyword-database and send back the similar pages.

2.2 Keyword extraction and definition of similarity

Many methods of Extraction from text have been proposed.[12] Web pages are written in hypertext markup language (HTML),[2] thus we utilize tag structure of HTML describing following to extract keywords. A method of using structure need not a set of documents, therefore a method of using structure is more efficient than that of ordinary using TFIDF.[12]

We define a set of keywords for a Web page. The page is made from words in tags. The tags weighs each words. Their weight are added weight of same words. The keyword set is made of keywords which is not less high weight than fifth within the page.

The tags and weight within parenthesis are described follow:

- `CONTENT` attribute in `<META>` tag (10)
- `<TITLE>` tag (10)
- `<H`n`>` tag provided that $1 \leq n \leq 3$ $(6, 4)$
- `<EM>` tag (1)
- `<U>` tag (1)

We give a definition of *similarity*. The similarity between two Web pages is defined with the number of elements in the product set between the keyword sets of the Web pages. Therefore, the similarity is between 1 and 5.

3 Execution of Bookmark-agent

This section presents a execution of Bookmark-agent. In this example, Bookmark-agent was given 6 users' bookmarks. The users are been called from A to F. Table 1 shows the number of their URLs in their own bookmarks. We do not count same URL for the total of URLs.

Table 1: The number of users' URLs in each bookmarks

User's name	A	B	C	D	E	F	Total
URLs	10	356	137	85	167	71	782

Here, we describe two examples. The first example is that an user does not give keywords but that shows his looking page on his browser for the agent monitoring. The second example shows when an user give keywords to the agent and some search engines.

When an user invoked Bookmark-agent, the agent began to look his browsers. The user changes to another new Web page in Fig. 2. Then the agent indicates hyperlinks of similar pages in Fig. 3. In this example, the user looks the page of *The Multi-Agent Systems Laboratory*. It is found that the agent recommends the page of the *Learning in Multi-Agent Systems* and its own page of *The Multi-Agent System Laboratory*. On this advice of the agent, it is intuitively recognized that user looking page resemble agent advising pages.

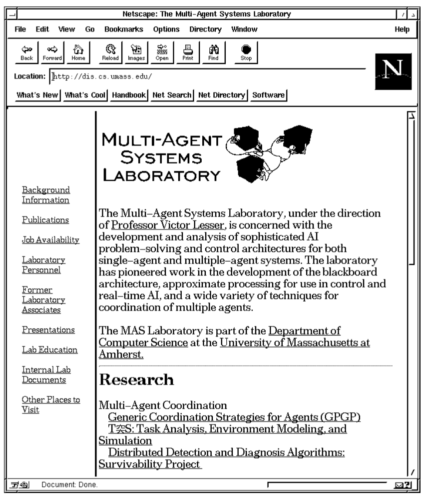

Figure 2: A Web page looked by an user at present time

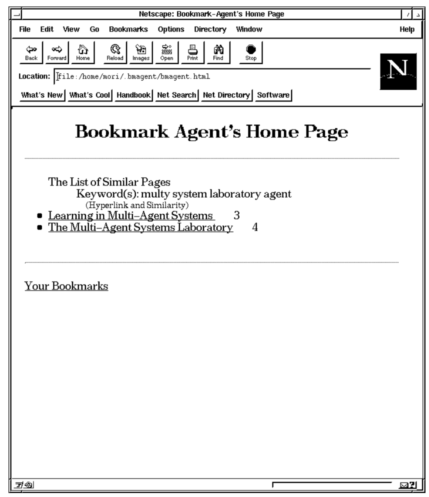

Figure 3: Bookmark-agent indicating page by implicit keywords

When an user explicitly gave keywords to Bookmark-agent, Fig. 4 shows similar pages which are indicated by Bookmark-agent. The links was recommended by the agent given the keyword 'agent' in this figure. Here, it is different that the agent indecates similar pages by the keywords from the result of searching the keyword-database by the keywords because the agent has been given the keywords by the way of a set of ketwords for similarity. On the other hand, the can give the same keyword to a search engine. Fig. 5 and 6 show the URLs that was indicated by Yahoo[2] and infoseek[3] , which are famous search engines. Yahoo gets keywords from Web pages which has been stored by their owners. In contrast, Infoseek uses a Web robot to retrieve keywords in the Web pages.

Note that Yahoo presents $2,550$ sites and that Infoseek presents $1,489,165$ sites, but that, in contrast, Bookmark-agent presents 2 sites. These search engines will present a large number of sites needed. This forces us to find out the target sites from a large number of URLs.

It would be difficult for him to get the URLs which the user really needed since the search engines recommends too general URLs to present a lot of URL: stravel agents, accountant agents etc. Therefore, if the user needs to find out the agent systems in the technical research area, the URLs presented were not often related to his interest.

Bookmark-agent, however, will suggests a small number of URLs which is filtered out kinds of travel agents if the user joins in a group of the technical research area of the agent systems. Hence the user obtains the information of needed URLs.

[2] `http://www.yahoo.com/`

[3] `www.infoseek.com`

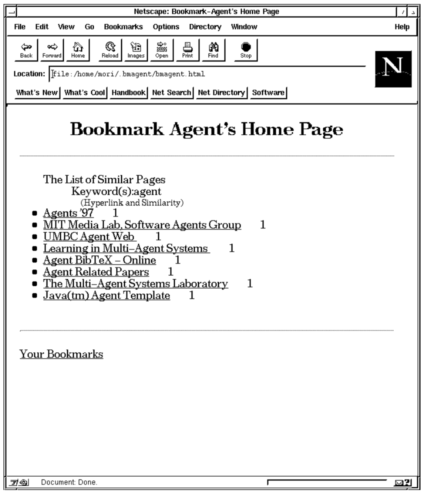

Figure 4: Bookmark-agent indicating page given keyword 'agent'

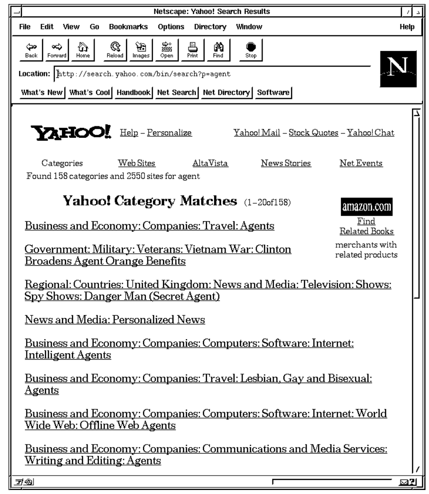

Figure 5: Answer from Yahoo

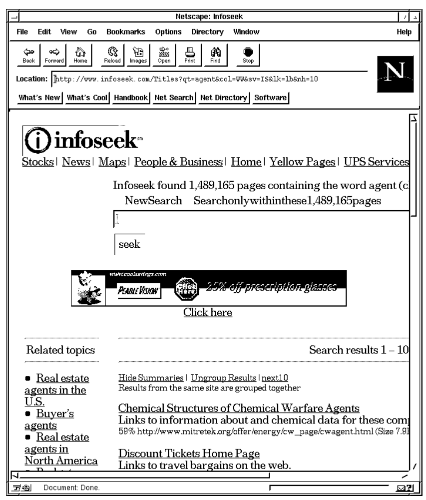

Figure 6: Answer from Infoseek

4 Experiments

In advance of experiments, Criteria for evaluating the property of Bookmark-agent are described in this section.

4.1 Details of Precision

In a field of Information retrieval, the performance of a retrieval system is evaluated in terms of its precision and recall. They are defined follow:[13]

- **recall**: The proportion of relevant material actually retrieved in answer to a search request.
- **precision**: The proportion of retrieved material that is actually relevant.

We use precision only since it is impossible that the all documents in the WWW as retrieved material are counted. Thus we had to give up to calculate recall. However precision can calculate because it is proportion of the user needing document to retrieved documents presented by search system for the WWW that searching system present from the WWW.

The user cannot frequently find out actually needed URLs when some search engines answered numerous URLs. But some search engines can present URLs that have been sorted by priority to user. So here, if the search engine answered the number of 20 URLs or more, we use the URLs between the top and 20th. And then we evaluate precision from the proportion of the retrieved documents that is actually relevant.

4.2 The Users for Subjects

For evaluating performance of Bookmark-agent, we prepared six users shown by Table 1 for subjects. These users used Bookmark-agent and had their own keyword-database. For the experiments, we can distinguish which of users owns oresented URL. Note that all of them use Linux and that study AI, robot, agent and so on and that they are not always interested in Perl and Java which are a programming language and that not interested in Sendmail which is a famous mail transfer program.

4.3 Experimental Result

Table 2, 3 and 4 show the results on precision when the keywords from one to three in th databases are given to an agent. We chose the keywords which are surely contained by total keyword-databases of all users.

For comparison with Bookmark-agent, the search engine Goo[4] is used since Goo answers a lot of URLs and uses a Web robot for retrieving. We thought that indicating a lot of URLs must have covered all area of ordinary interest.

In these experiments, the user C in Table 1 gave keywords to Bookmark-agent or the search engine Goo. Then he Judged presented sites of URLs whether relevant or not.

5 Discussion on The Experimental Results

We discuss experimental results in this section. It is different behavior between Bookmark-agent and Goo. As a result, then we find feature of Bookmark-agent.

5.1 Precision for Bookmark-agent

The characteristic of the keywords in the experiments influences the precision. The keyword which is easy to change int adjective takes lower precision than which is noun only. For instance in Table 2 and 3, the keywords 'internet', 'web' and 'english' can come to noun, but they come to adhectives when each of them made a phrase as "Intermet Program", "Web Agent" and "English Dictionary". Thus they take higher precision when they are given with other words than when they are given each theirself only. Evidence of this is that precision of phrases 'web, agent' and 'english, dictionary' took 100% but that precision of word 'web' took 35.7% and 'english' took 37.5%.

5.2 Difference from The Search Engine on Precision

A keyword which is used in a group having comon sense will be presented higher precision than which is without comon sense. For instance, 'agent'

[4] http://www.goo.ne.jp/

Table 2: Precision when 1 keyword was given

Keyword	Called URLs		Precision (%)	
	Agent	Goo	Agent	Goo
agent	8	1811187	100.0	5.0
robot	10	268591	100.0	30.0
intelligence	10	23622	100.0	10.0
ai	4	1134699	100.0	45.0
research	4	8465215	75.0	20.0
internet	15	13131781	26.7	15.0
web	14	128811	35.7	10.0
linux	22	1415551	100.0	70.0
unix	6	2708291	100.0	40.0
science	3	5477707	33.3	50.0
software	24	9113879	91.7	45.0
faq	7	3144108	100.0	50.0
perl	4	893256	75.0	75.0
java	16	2317347	100.0	45.0
sendmail	4	168982	100.0	65.0
search	13	13542226	84.6	55.0
mac	22	1974327	100.0	60.0
macintosh	16	1575040	100.0	85.0
weather	1	2705383	100.0	75.0
dictionary	6	11096	100.0	35.0
japanese	19	1400994	21.1	40.0
english	8	4262210	37.5	5.0
ml	4	540252	75.0	35.0
archie	5	282468	100.0	55.0
Average			81.5	44.2

Table 3: Precision when 2 keywords were given

Keywords		Called URLs		Precision (%)	
		Agent	Goo	Agent	Goo
software	agent	1	400238	100.0	30.0
web	agent	1	541780	100.0	15.0
linux	software	3	494729	100.0	35.0
free	software	1	2502799	100.0	10.0
english	dictionary	3	197152	100.0	70.0
java	software	1	687154	100.0	30.0
java	internet	1	766101	100.0	15.0
web	robot	1	91377	100.0	55.0
artificial	intelligence	6	264060	100.0	45.0
linux	application	1	129544	100.0	20.0
tokyo	university	1	207367	100.0	50.0
mailing	list	4	2453832	100.0	75.0
fj	news	1	40508	100.0	60.0
sports	soccer	1	374736	100.0	35.0
Average				100	38.9

Table 4: Precision when 3 keywords were given

Keywords			Called URLs		Precision (%)	
			Agent	Goo	Agent	Goo
ai	artificial	intelligence	1	75011	100.0	45.0
agent	robot	autonomous	1	5106	100.0	85.0
linux	tool	application	1	40581	100.0	40.0
ai	lab	mit	1	40946	100.0	90.0
english	japanese	dictionary	2	35977	100.0	35.0
mailing	list	japan	2	111290	100.0	70.0
tokyo	institute	technology	1	72575	100.0	75.0
mac	news	information	1	303161	100.0	30.0
Average					100	58.8

usually means that the person who takes user's place or give service for users who cannot do. However it is clear that the 'agent', in the group, means agent system computing on AI.

It is sure that the precision of keyword 'agent' in the search engine Goo became half or less than average in Table 2. However, the precision of 'agent' in Bookmark-agent scarcely changed. Therefore it is reasonable to suppose that users have common sense at a keyword is considered as that the users share preference.

Here, we gave a rank sum test between precision of each keyword from Bookmark-agent system and that from Goo. The test tests on each case of one keyword of presision to three. As a result, we found a significant difference on significance level 5% by each tests. Thus the methodof search engines did not have ability of method of Bookmark-agent.

5.3 Whether Preference is Shared or Not?

First, we pay attention that the users share preference. The user in the experiments does not share preference at general science but that they have common sense at AI, robot and agent system. Therefore precision of 'ai', 'robot' and 'agent' is higher than that of 'science' in Table 2. The keyword at which they share preference presents more URLs than one at which they do not share.

Next, we pay attention that the users take different preference each other. The users have common sense at both of the keywords 'robot' and 'linux', therefore each of keywords showed 10 or 22 URLs. On the other hand, there is not common sense at 'perl' since we know that the user C only have preference. As a result, the keyword 'perl' showed only 3 URLs. A user ought to usually be indicated more URLs at 'unix' than that at 'linux' in Table 2 since UNIX is more general than Linux. The keyword 'linux', however, indicated more URLs because of the users using Linux.

6 Conclusion and Future Works

We proposed Bookmark-agent using bookmarks to support an users' searching information, and made experiments by six users. We consider bookmarks as the sharing information as URL. Bookmark-agent are clearly not different from general search engines on the point that these search engines often give a large number of URLs with no wanted URLs for user. Bookmark-agent which uses bookmarks is able to give a small number of URLs just wanted by user who shares preference with other users.

We will classify users into some groups by some method which will measure similarity among users in future. So far Bookmark-agent uses on condition that the users in a group of common sense are gathered beforehand, and the agent shares URLs in the premise. We will make a match-making system utilizing similarity among users.

References

[1] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T. : WebWatcher: A Learning Apprentice for the World Wide Web, *1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments* (1995)

[2] Berners-Lee, T., and D. Connolly : Hypertext Markup Language – 2.0, *RFC 1866* (1995)

[3] Berners-Lee, T., Masinter, L., and McCahill, M. : Uniform Resource Locators (URL), *RFC 1738* (1994)

[4] Chen, L., Sycara, K.: WebMate: A Personal Agent for Browsing and Searching (1997)

[5] Doorenbos, R. B., Etizoni, O. and Weld D. S. : A Scalable Comparison-Shopping Agent for the World-Wide Web, *Proceedings of 1st Autonomous Agents* (1997)

[6] Huberman, B. A., Kaminsky, M.: Beehive: A System for Cooperative Filtering and Sharing of Information (1996)

[7] Joachims, T., Mitchell, T., Freitag, D., and Armstrong, R. : WebWatcher: Machine Learning and Hypertext, *Fachgruppentreffen Maschinelles Lernen* (1995)

[8] Joachims, T., Freitag, D., and Mitchell, T. : WebWatcher: A Tour Guide for the World Wide Web, *Proceedings of 15th International Joint Conference on Artificial Intelligence*, pp. 770–775 (1997)

[9] Lieberman, H. : Letizia: An Agent That Assists Web Browsing, *Proceedings of 14th International Joint Conference on Artificial Intelligence*, pp. 924–929 (1995)

[10] Pazzani, M., Muramatsu, J. and Billsus, D. : Syskill & Webert: Identifying interesting web sites, *Proceedings of 13th National Conference on Artificial Intelligence and 8th Conference on Innovative Applications of Artificial Intelligence*, pp. 54–61 (1996)

[11] Rucker, J. and Polanco, M. J. : Siteseer: Personalized Navigation for the Web, *Communications of ACM*, vol. 40, No. 3, pp. 73–75 (1997)

[12] Salton, G. and McGill, M. J. : Introduction to modern information retrieval, *McGraw-Hill* (1983)

[13] van Rijsbergen, C. J. : Information Retrieval, *London: Butterworths* (1979)