

Collaborative Information Filtering by Using Categorized Bookmarks on the Web

Jason J. Jung¹, Jeong-Seob Yoon¹, Geun-Sik Jo²

¹ Intelligent E-Commerce Systems Laboratory,
Department of Computer Science and Engineering, Inha University, Incheon, Korea
{jayjung, jsyoon}@eslab.inha.ac.kr

² Department of Computer Science and Engineering, Inha University, Incheon, Korea
gsjo@inha.ac.kr

Abstract. A bookmark means the URL information stored for memorizing a user's own footprints and revisiting that website. This paper regards this bookmark as one of the evidences representing user preferences. An original bookmark indicating only address information is categorized for merging semantic meanings by using public web directory services. These categorized bookmarks are expressed in a hierarchical tree structure. However, most current web directory services cannot afford to normalize and manage the topic hierarchy. There are several kinds of structural incompleteness such as multiple references and heterogeneous tree structures. In order to extract user preferences, this paper proposes a method for driving these problems and the influence propagation methods based on Bayesian networks. Therefore, the preference maps of each user that are representing their preferences are also established as tree structures. In respect to the user clustering, an approximate tree matching method is used for mapping (overlapping) users' preference maps. It is possible to make queries and process them efficiently according to categories. Finally, this paper is applied to implement collaborative web browsing that can guide and explore the web efficiently and adaptively.

1. Introduction

In recent years the development of network technology and the Internet infrastructure have brought about many changes in our lives. As the Internet has developed, too much information has been flowing on it. There is no doubt that *Information overload*, which makes it hard for users to search for proper information, is one of the most serious problems [1]. Many kinds of information filtering methods have appeared in order to handle this problem. For example, the e-mail filtering system, Ringo, and WebHound can be called personalized information filtering systems [2], [3], [4], [5]. Therefore, user preference (or interest) is information that should be extracted first in order to increase the performance of filtering systems.

While the sparsity of information about users and dynamic characteristics of the web environment have caused difficulties in the acquisition of user preference, there are many kinds of information that make user preferences inferable, such as footprints on navigation, viewing/accessing time of web pages, and site access frequency. A bookmark that is stored for

revisiting a particular site and memorizing the URL information can also be included as one of these pieces of evidence. In this paper, a bookmark is regarded as the most representative information of user preferences. There is an information filtering system to which bookmarks are applied. BISAgent (Bookmark Information Sharing Agent) is a collaborative web browsing system based on a modified *TF-IDF* scheme without considering user preference [6], [7]. Nowadays web directory services like Yahoo [8] are offering users much more relevant answers than any other general keyword-searching engines, even if the web coverage provided by directories is very low [9]. Web directories are also called categories, yellow pages, or sometimes subject directories. Categories are hierarchical taxonomies that classify human knowledge or a general class of ideas, terms, or things that mark divisions or coordinations within a conceptual scheme. Not only Yahoo but also digital libraries, UseNet, and DMOZ (the open directory project) have served as a category service [10], [11]. This paper assumes that the most important role of a category is that it is possible to be used for extracting user preferences. Because categories on a hierarchical tree structure are mutually dependent, influences between them have to be considered. As hierarchical influences propagate based on Bayesian network, all user preferences can be detected as well as the fact that the favorite categories of each user can be extracted. Thus, preference maps of each user can be established as more than one disjointed tree, called a clique [12]. Approximate tree matching is a method used in molecular biology to compare vast amounts of RNA structures, natural language processing, or pattern recognition [13]. We can make queries to inquire about the user group interested in each category. Then, the similarities between users are calculated for clustering like-minded users by approximate tree matching. Through user clustering, we can apply this result to recommend and distribute relevant information adaptively to each user on collaborative browsing.

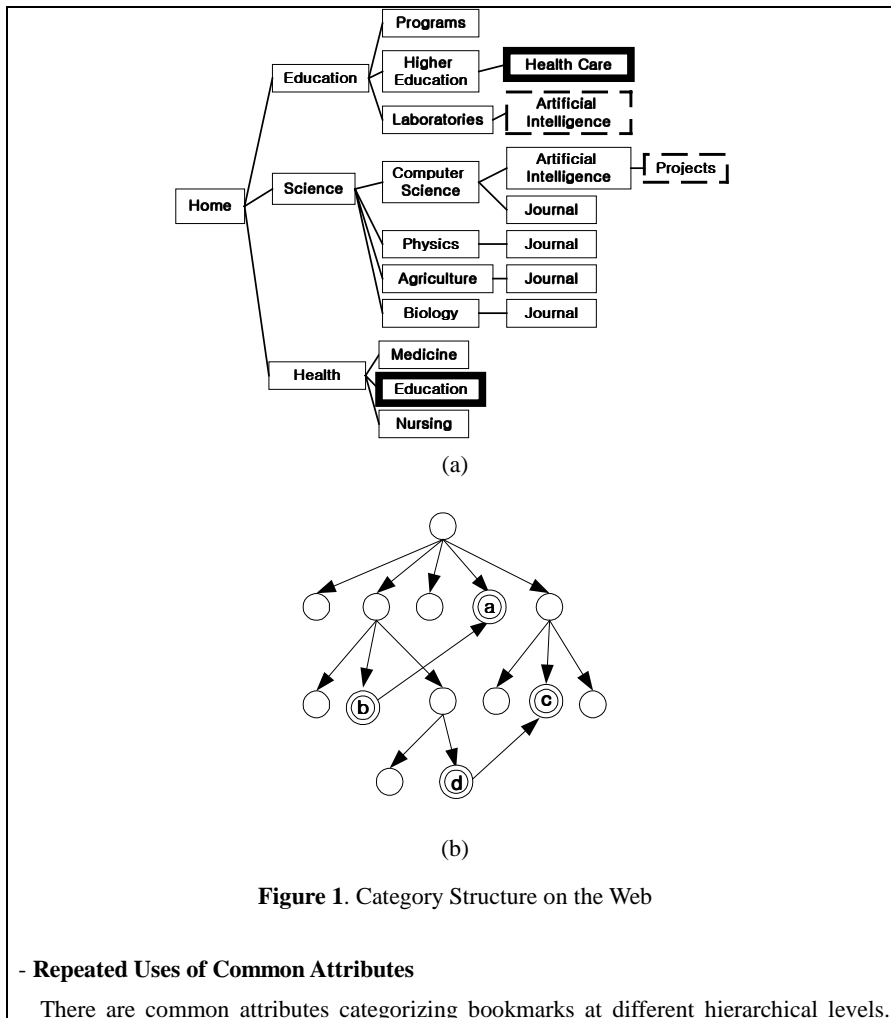
2. Related Work

Originally most web information retrieval systems made use of only the text information on the pages, ignoring valuable link information (not only hyperlinks). Moreover, heterogeneity, cross-domain links and the dynamic nature of the web used to make information retrieval extremely difficult. As ParaSite makes use of new kinds of information available, such as multiple independent categorizations, naming, and indexing of pages, it can mine structural information on the web by hyperlinks [14]. Contextual categorical information can be retrieved through instance inheritance. When a hierarchical structure-like category keeps semantic relationships between nodes, structural information can be derived by using inherited categorical properties [15]. There is research automatically constructing a category tree structure by analyzing bookmarks. This is why most web directory services are being managed by just the users. Therefore, some information cannot be guaranteed because of its ambiguity [21]. Currently some companies have developed Bayesian inference tools like HUGIN Expert. This system uses influence propagation on Bayesian networks [20]. To approach a computer vision problem, hierarchical tree structures constructed by palm prints of humans are matched by using TAG's (Tree Association Graphs) [10]. So far many companies have tried to manage personal bookmarks without information filtering. For instance, Linkify (www.linkify.com) and Backflip (www.backflip.com) are representative companies associated with bookmarks.

3. Preliminary Assumptions and Definitions

3.1 Bookmarks and Category Structures on the Web

All bookmarks in this paper are categorized by a well-organized directory service, like Yahoo and Cora (cora.whizbang.com). A topic hierarchical structure is an efficient way to organize, view and explore large quantities of information that would otherwise be cumbersome [19]. A category structure on the web is a connected, acyclic, and directed tree structure. Even though it is possible to make a cycle among some nodes, this paper ignores this. The characteristics of category information on the web are as follows:



For example, as shown in Figure 1 (a), the attribute 'Journal' can be used for categorizing bookmarks in most of the subcategories of 'Science,' such as 'Agriculture> Journal,' 'Biology> Journal,' and so on.

- Heterogeneous Category Structure

This means that all nodes do not connect only with their root node. In other words, a category can be pointed by a category on a totally different path. As in Figure 1 (b), nodes 'a' and 'c' are connected not only with their parent nodes but also with nodes 'b' and 'd,' respectively. Practically, most companies are forced to manage this non-generic tree structure in order to avoid a waste of memory space by the redundancy of information. As shown in Figure 1 (a), the category "Health_Care", one of the subcategories of "Home>Education>Higher_Education>," just points to "Home>Health>Education," which contains the same information.

- Multiple References

This means that a bookmark can be categorized into more than one category. For example, "Intelligent E-Commerce System Lab., Inha University" is not only one piece of information in "home>education>laboratories>artificial intelligence>" but also in "home>science>computer science>artificial intelligence>projects>."

3.2 Categorization of a Bookmark

As shown in Figure 2, by referring to a well-organized global category set, each user's bookmark will be categorized.

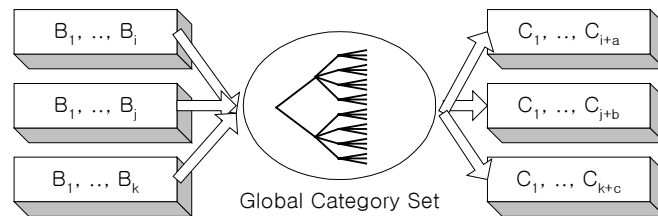


Figure 2. Categorization of a Bookmark

At this time, the size of each user's category set become larger than that of his bookmark set because of incomplete properties of the category structure, such as a multiple reference. Due to the heterogeneity of a tree, we have made a candidate category set. This is the reason why the coverage of user preferences should be improved. A global category set is a kind of dictionary that matches bookmarks with categories.

3.3 Notation

If the number of all categories is T, the global category set is represented by the following equation.

$$Global_Category = \{C_1, \dots, C_T\}$$

The following is a bookmark set and a category set of the i^{th} user.

$$Bookmark(User_i) = \{B_1, \dots, B_M\} \quad (M : \text{the size of the } i^{\text{th}} \text{ user's bookmark set})$$

for $k = 1$ to M do

$$Category(User_i) = Category(User_i) + Categorize(B_k)$$

if ($am_i_pointed(Categorize(B_k)) = \text{true}$) then

$$Candidate_Category(User_i) = Candidate_Category(User_i) + pointed(Categorize(B_k))$$

$$Category(User_i) = \{C_1, \dots, C_N\} \quad (N : \text{the size of the } i^{\text{th}} \text{ user's category set})$$

$$Candidate_Category(User_i) = \{C_1, \dots, C_\alpha\} \quad (\alpha: \text{the number of pointed categories})$$

The function *Categorize* looks up the global category set for matching bookmarks to proper categories. In addition, the function *am_i_pointed* and *pointed* are related to the heterogeneity of category structure. The function *am_i_pointed* can check whether a parameter is connected to more than two categories or not, including the parent category. The other function *pointed* can ask which categories are pointing to that category apart from its parent category. Categories in the user's category set will be called a 'hit' by the user's bookmark.

4. Extracting User Preference Based on Influence Propagation on Bayesian Network

Basically Bayesian networks are probabilistic models that allow the structured representation of a cognitive or decision process and are commonly used for decision tree analysis of business and the social sciences [17], [18]. The following is used to measure conditional probability.

$$P(H, E) = P(H | E) \times P(E)$$

The strengths of causal influences between categories are expressed by these conditional probabilities [9]. This is how categories reflect their causal relationship to parent nodes. The degree of user preference about Node *Parent_Category* is a measure of how much the evidential support nodes *Child_Categories* provide to Node *Parent_Category*.

$$\begin{aligned} &P(Parent, Childs) \\ &= \sum_i P(Parent | Child_i) \times P(Child_i) \end{aligned}$$

In order to extract user preference, first, all categories included in the category set are assigned a value of user preference (*VOP*), which expresses the degree of user interest. Then, their influences are propagated to their parent categories to assign *VOP*'s.

$$VOP(Parent) = \sum_i (Propagate[VOP(Child_i)] \times P(Child_i))$$

At this point, the following axioms define the rules used to assign *VOP*'s to each category.

Axiom 1. The initial *VOP*'s of each category are the number of 'hit' times of each category in the category set.

For all included categories, $VOP(Category_i) = \text{number of hit times}$

Axiom 2. As the 'hit' number of a category increases, it is inferred that the user is more interested in that category.

$\text{number of hit time} \propto VOP(Category_i)$

This means that the number of 'hit' times is in linear proportion to user preference for that field.

Axiom 3. As the number of subcategories of a parent is increased, each subcategory has less influence on the parent. This is why user interests are dispersed.

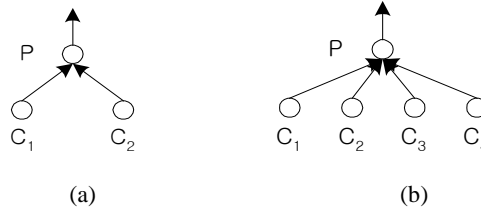


Figure 3. Influence dependent on the number of categories

The influence of node C_1 in Figure 3(a) is two times as strong as that of node C_1 in Figure 3(b), because the ratio between them is equal to two. Additionally, this criterion is reflected to the propagating function by calculating the variance of the hit number of categories. Less variance means a stronger evidential support to the parent category.

Axiom 4. Hierarchically the lower category is more influential in deciding user preferences than the upper one. This paper assumes that the influence from child nodes to parent node is exponentially decreased, as the distance between nodes increases. As shown in Figure 4, because node C_1 and C_2 are hit, node C_1' is influenced by C_1 . Especially, node C_2' is influenced by C_1' and C_2 . These nodes' *VOP*'s are presented in the following equations:

$$VOP(C_1') = Propagate[VOP(C_1)] \times VOP(C_1)$$

$$VOP(C_2') = Propagate[VOP(C_1')] \times VOP(C_1') + Propagate[VOP(C_2)] \times VOP(C_2)$$

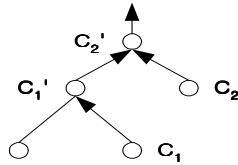


Figure 4. Influence propagation on Bayesian network

Axiom 5. The category set *Candidate_Category* is also caused by the same level of influences as that of the normal one.

Axiom 6. All categories' influences are propagated up to the vertex node.

Axiom 7. Categories whose *VOP*'s are over the threshold value can represent user preference after a normalization step. The threshold value controls how many categories will be extracted.

The function *Propagate* in *Axiom 4* is defined as a logarithm function in the following equation.

$$Propagate[VOP(C)] = \frac{\log_k(VOP(C) + 1)}{N}$$

$$k = \text{Variance}(VOP(\text{subcategories})) + 2 = \sigma^2 + 2$$

N = The number of subcategories of a parent category

For normalization, the rate of the average of *VOP*'s (μ) of all categories and the portions of each category has to be calculated. For the portion over the threshold value, these categories are regarded as the representative categories of user preference.

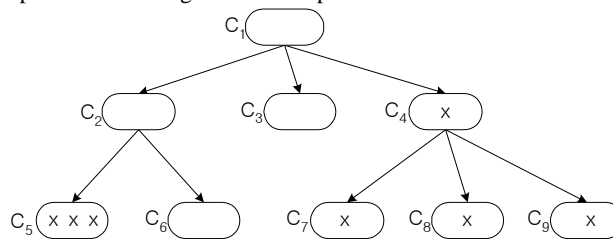


Figure 5. Example for extracting user preferences

As shown in Figure 5, 'X' expresses the bookmarks of a user in proper categories.

$$\text{Bookmark}(U) = \{B_1, \dots, B_7\}$$

$$\text{Category}(U) = \{C_4, C_5, C_5, C_5, C_7, C_8, C_9\}$$

For initial VOP's,

$$VOP(C_4) = 1, VOP(C_5) = 3, VOP(C_6) = 0, VOP(C_7) = 1, VOP(C_8) = 1, VOP(C_9) = 1$$

For means and variances of categories' VOP's,

$$\begin{aligned} \mu(\{VOP(C_5), VOP(C_6)\}) &= 1.5 & \sigma^2(\{VOP(C_5), VOP(C_6)\}) &= 4.5 \\ \mu(\{VOP(C_7), VOP(C_8), VOP(C_9)\}) &= 1 & \sigma^2(\{VOP(C_7), VOP(C_8), VOP(C_9)\}) &= 0 \end{aligned}$$

Therefore, C₂ and C₄ are propagated from their child categories.

$$VOP(C_2) = \sum_{k=1}^2 \text{Propagate}[VOP(C_k)] \times VOP(C_k) = \frac{\log_{6.5}(VOP(C_5) + 1)}{2} \times VOP(C_5) = 1.11$$

$$VOP(C_4) = 1 + \left(\frac{\log_2 2}{3} \times 1\right) \times 3 = 1 + 1 = 2$$

In the same way, we can get the VOP of C₁.

$$VOP(C_1) = \left(\frac{\log_{2.93}(1.11 + 1)}{3} \times 1.11\right) + \left(\frac{\log_{2.93}(2 + 1)}{3} \times 2\right) = 0.94$$

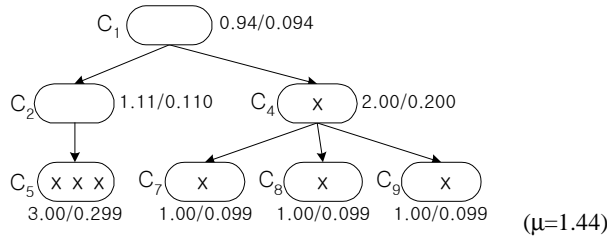


Figure 6. The result of VOP's and portions of each category

Next, the mean of all VOP's is 1.44 and Figure 6 illustrates the VOP and the rate of each category after normalization.

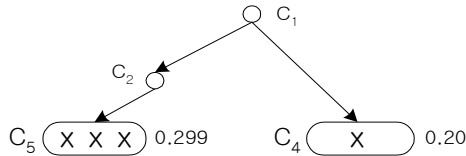


Figure 7. The high ranked categories for user preferences, PM

If the threshold value is 0.2, only C_4 and C_5 are extracted, as the most interested categories to the user as in Figure 7. Furthermore, we call this tree a ‘preference map (PM)’ of the user. Each user gets this PM and every time he inserts his bookmark, this PM has to be updated.

5. Approximate Tree Matching for User Clustering

Comparing preferences between users is a simple task for users who are interested in only one field. However, ordinary users are generally interested in more than one. The similarities (or distances) between users’ PM ’s have to be measured in order to recognize how much both users have similar preferences. Many algorithms have been studied and developed for exact tree comparison.

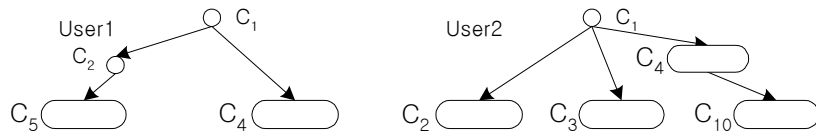


Figure 8. PM s of User1 and User2

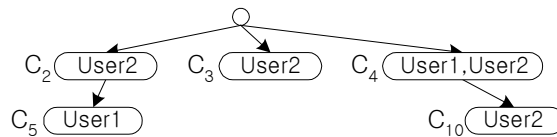


Figure 9. All categories labeled user information

This paper is based on the approximate tree matching algorithms presented in [13], [22]. The edit operations give rise to a *mapping* that is a geographical specification of which edit operations apply to each node in the two trees in order to determine the editing distance between sequences [23]. In this paper, *mapping* is simply regarded as overlapping both PM ’s. As shown in Figure 9, the PM of User1 and the PM of User2 in Figure 8 overlap for processing queries with respects to each category efficiently. Therefore, we can make queries, to investigate the user group interested in a particular category. For example, a query such as “Who is interested in the category home>computer>database>?” is available. Moreover, for this query, it is possible that not only users in the category but also users in its subcategories are replied to.

Next, the distance between two PM s is expressed as the following equation.

$$Similarity(User_1, User_2) = threshold - \sum_i \sum_j \min[dist(Category(User_1), Category(User_2))]_j$$

The variables i, j mean the size of the high ranked category set of user₁ and user₂, and i has to

be equal to or larger than j . If the distance among sibling nodes is equal to k , the distance to the parent node is $k/2$, which is a depth difference. We can calibrate the variable *threshold* and k to establish the threshold value as the standard and to control the number of user clusters.

6. Experiment

We made up a hierarchical tree structure as a test bed for “Home>Science>Computer Science>” from Yahoo [8]. This sampled tree consists of about 1300 categories and the maximum depth is 8. For gathering bookmarks, 30 users explored Yahoo directory pages for about 50 hours (2hours for 4 weeks) using this system. Whenever they found a site related to their own preference, they stored that URL information. As a result, 2718 bookmarks were collected. After 8.1 days 80% of the total bookmarks are inserted. We used these testing data to evaluate the performance of information filtering according to user preference. The evaluation measures ‘Recall’ and ‘Precision’ are utilized in order to present quantitative comparison between both cases.

6.1 Evaluation of user preference extraction

After resetting the bookmark set of all users, these users began to gather bookmarks again upon getting the system’s recommendations according to their own preferences.

1) The measure, Recall

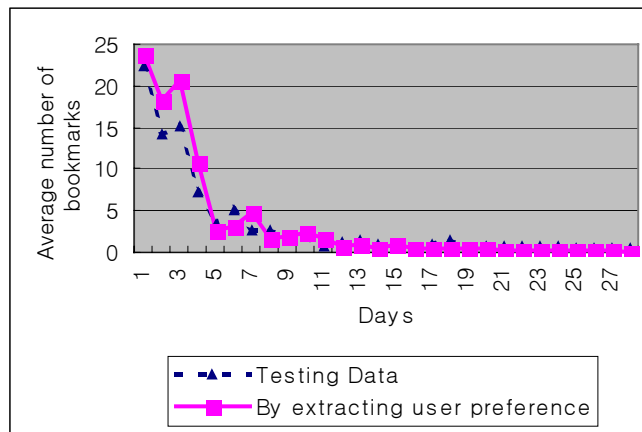


Figure 10. Evaluating with respect to Recall

During this time users were being recommended information retrieved from Yahoo based on user preference extracted up to that moment.

	Time for collecting 80% of total bookmarks (Days)
Testing Data	8.1
With recommendation of system	3.8

Table 1. The result of evaluating with respect to ‘Recall’

As a result, 80% of the total bookmarks were collected for 3.8 days.

As shown in Table 1, 53.1% of the total time could be saved with respect to recall.

2) The measure, Precision

The measure precision was measured by the rate of the inserted bookmarks among the recommended information set. In other word, this is the measurement for the accuracy of predictability. As time passes, the user preferences are changed according to the inserted bookmarks. At the beginning, the precision was especially low because the user preferences were not yet set up.

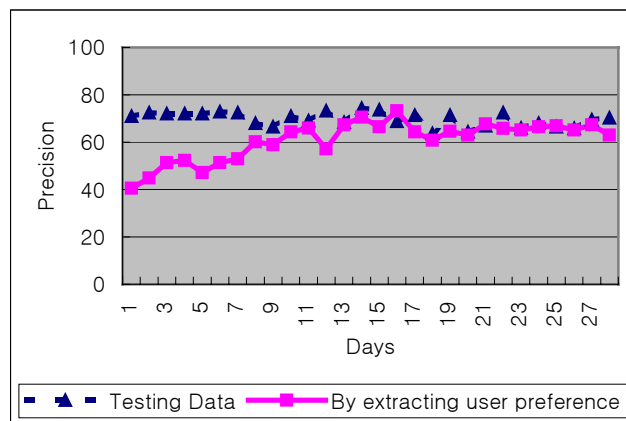


Figure 11. Evaluating with respect to Precision

While user preferences were being extracted in the first 6 days, the ‘Precision’ of recommended information tracked to that of the testing data quickly, as shown in Figure 11. For the final 15 days, it maintained the same level as that of the testing data.

6.2 Collaborative web browsing by user clustering

1) System architecture

Web browsing to search for relevant information is not only a difficult but also a boring task. Hence, collaborative web browsing through which users can share helpful information to increase the performance of information searching is used in digital libraries, and particularly,

for guiding the beginning users. Such systems are “Let’s Browse” and “ARIADNE” [24, 25]. The system architecture of the client-side and the server side is represented in Figure 12. All bookmarks are patched into the server in order to infer the preferences of each user. The user clustering module is for computing the similarities between users. Both query generators for user information and category information can ask the user information repository and the global category set in web directory service.

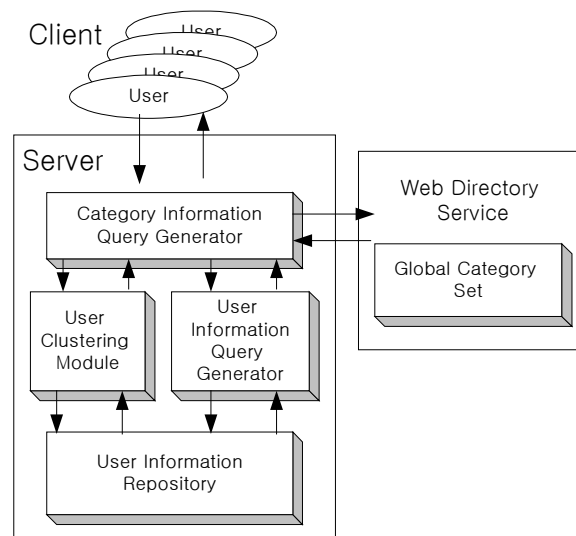


Figure 12. System architecture

2) User interface

As shown in Figure 13, the left side is just a window for viewing site information.

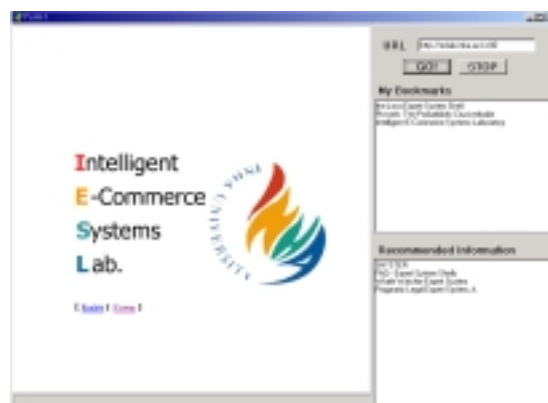


Figure 13. User interface for collaborative web browsing

The upper window on the right side is a space for checking the user's own bookmarks. The lower-right window shows the recommended information from the server.

7. Conclusions and Future Work

We have assumed that categorized bookmarks give us information which is more evidentially supportable than any other to extract user preferences. Incompleteness of practical directory services could be driven by this system. By propagating the influences on Bayesian networks, all possible and potential preferences of the user were covered and extracted. With this system 74% of the users were able to find the field that they are potentially interested in, according to experiments. The influence propagation proposed in this paper is confirmed to be reasonable. Additionally, the evaluation measures 'Recall' and 'Precision' were acquired. Moreover, the 'Recall' showed high-level values, which means that this system is covering as many user preferences as possible. In the beginning, the 'Precision' was low-level because of a deficient adaptation. However, by the end of the experiment period, the 'Precision' value also converged to as stable a level as that of testing data quickly. This means instantly updating user preference as accumulated information. Finally, information filtering based on influence propagation allows users to retrieve highly relevant information. To approach a practical system, collaborative web browsing was applied to this method and user clustering by using approximate matching. In conclusion, we have verified not only high performances of information retrieval but also the possibility of navigation.

For future work, because it is hard for multi-preference users to detect their preferences and to be clustered, we should investigate new criteria based on human factors. In addition, the visualization of a personal bookmark management system will be implemented

Reference

1. P. Maes. "Agents that Reduce Work and Information Overload," *Communications of the ACM* 37, 7 (July 1994): 31-41.
2. B. Sheth and P. Maes. "Evolving agents for personalized information filtering," In *Proceedings of the ninth IEEE Conference on Artificial Intelligence for applications*, 1993.
3. M. Pazzani, L. Nguyen, and S. Mantik. "Learning from hotlists and coldlists: towards a WWW information filtering and seeking agent," In *Proceedings of AI Tools Conference*, Washington, DC, 1995.
4. U. Shardanand and P. Maes. "Social Information Filtering: Algorithms for Automating Word of Mouth'," *Proceedings of CHI'95 Conference on Human Factors in Computing Systems*, ACM Press, 1995.
5. Y. Lashkari, "Webhound," Master's thesis, MIT Media Laboratory, 1995.
6. J. Jung, J. Yoon, and G. Jo, "BISAgent: Collaborative Web Browsing through Sharing of Bookmark Information," *16th IFIP World Computer Congress 2000 (IIP)*, Beijing, China, 2000.
7. G. Salton and M. E. Lesk, "Term-weighting approaches in automatic retrieval," *Information Processing & Management*, 24(5):513-523, 1988.
8. Yahoo, <http://www.yahoo.com>

9. R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, 1999.
10. M. Pelillo, K. Siddiqi and S.W. Zucker, "Matching hierarchical structures using association graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(11), Nov. 1999.
11. DMOZ, <http://dmoz.org/>
12. E. Horowitz and S. Sahni, "Fundamentals of Data Structures in Pascal," Computer Science Press, 1994.
13. J. Wang, K. Zhang, K. Jeong, and D. Shasha, "A System for Approximate Tree Matching," In IEEE Transaction On Knowledge and Data Engineering, volume 6, pages 559--570, August 1994.
14. E. Spertus, "ParaSite: Mining Structural Information on the Web," in The Sixth International World Wide Web Conference, 1997.
15. A. Analyti, N. Spyrtos, and P. Constantopoulos, "Deriving and Retrieving Contextual Categorical Information through Instance Inheritance," 1998, www.ics.forth.gr/proj/isst/Publications/
16. G. Navarro and R. Baeza-Yates. "Improving an algorithm for approximate pattern matching," Technical Report TR/DCC-98-5, Dept. of Computer Science, Univ. of Chile, 1998.
17. J. Pearl, "Probabilistic reasoning in intelligent systems," Morgan Kauffman Publisher, 1988.
18. J. Giarratano and G. Riley, "Expert systems principles and programming," PWS publishing company, 1994.
19. A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Building Domain-Specific Search Engines with Machine Learning Techniques," AAAI Spring Symposium, 1999.
20. "HUGIN EXPERT White Paper," <http://www.hugin.com>
21. S. Chakrabarti and Y. Batterrywala, "Mining themes from bookmarks," ACM SIGKDD KDD 2000 Workshop on Text Mining, 2000.
22. A. Apostolico and Z. Galil, (editors) "Pattern Matching Algorithms - Approximate Tree Pattern Matching (Chapter 14)," Oxford University Press, 1997.
23. R. S. Boyer and J. S. Moore, "A fast string matching algorithm," CACM, Vol. 20, 1977.
24. H. Lieberman, N. Dyke, and A. Vivacqua, "Let's Browse: A Collaborative Web Browsing Agent," International Conference on Intelligent User Interfaces, 1999.
25. Twidale, M. and Nichols, D. "Collaborative Browsing and Visualization of the Search Process," Electronic Library and Visual Information Research Conference (ELVIRA-96), Milton Keynes, England, May 1996.