Learning User's Preferences by Analyzing Web-Browsing Behaviors

Young-Woo Seo
Artificial Intelligence Lab (SCAI)
School of Computer Sci. and Eng.
Seoul National University
Seoul 151-742, Korea
Phone: +82 2 880 1835
ywseo@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Sci. and Eng.
Seoul National University
Seoul 151-742, Korea
Phone: +82 2 880 1833
btzhang@scai.snu.ac.kr

ABSTRACT

This paper describes a method for an information filtering agent to learn user's preferences. The proposed method observes user's reactions to the filtered documents and learns from them the profiles for the individual users. Reinforcement learning is used to adapt the most significant terms that best represent user's interests. In contrast to conventional relevance feedback methods which require explicit user feedbacks, our approach learns user preferences implicitly from direct observations of browsing behaviors during interaction. Field tests have been made which involved 10 users reading a total of 18,750 HTML documents during 45 days. The proposed method showed superior performance in personalized information filtering compared to the existing relevance feedback methods.

Keywords

Information filtering agents, learning user's preferences, reinforcement learning, user modeling.

1. INTRODUCTION

With the rapid progress of computer technology in recent years, electronic information has been explosively increased. This trend is especially remarkable on the Web. As the availability of the information increases, the need for finding more relevant information on the Web is growing. Therefore, filtering appropriate information is a critical issue to relieve Web users from information overload [1].

Currently, there are two major ways of accessing information on the Web. One is to use Web index services such as AltaVista, Yahoo!, and Excite. The other is to repeatedly follow or browse the hyperlinks of the documents by a user himself. However, these methods have some drawbacks. When using the Web index services, much of the retrieved results may be irrelevant to user's interests. The method of manual browsing involves much time and efforts.

The low quality of index services is in part due to the fact that users have personal desires in mind, but the index service systems do not take into account the personal preferences. This is because they are based on general purpose indexing methods [14]. High quality service requires to catch the personal interests over the interaction with the information retrieval system.

In this paper, we describe a method for learning user's preferences by observing user behaviors during his interactions with the system. The method is based on reinforcement learning and implemented in a Web-based information filtering system called WAIR (Web-Agents for Information Retrieval).

Two different aspects are employed to filter more relevant Web-information. One is to acquire user's opinion exactly without extra efforts. Another is to learn user's profile. WAIR observes user's Web-browsing behaviors to the filtered documents, such as bookmarking, following the hyperlinks, scrolling up and down, and spending time in reading. And it gets user's relevance feedback implicitly. This information is used for learning the profiles, where reinforcement learning is applied.

Reinforcement learning is a goal-directed learning method based on interactions with the environment. We take the reinforcement learning method because it can learn on-line.

The paper is organized as follows. In the following section, we review existing methods for learning user preferences. Since we focus on information filtering on the Web, our survey is confined to relevance feedback methods developed for information retrieval. Then, we describe our method for learning the user interests along with the WAIR system and its filtering procedure. Finally, the method is evaluated through experiments.

2. LEARNING USER'S PREFERENCES IN INFORMATION FILTERING

We propose a method that learns user's preferences by analyzing Web-browsing behaviors and filters more relevant documents by using the learned profile. This section describes related works in view of relevance feedback and agent technology.

Agents 2000 Barcelona Spain

2.1 User Relevance Feedback

In information retrieval and filtering, users are usually not able to express their information needs in exact terms. But, they easily can evaluate a document on whether relevant or not to their information needs. The evaluation by a user is called user relevance feedback and is used for improving retrieval and filtering accuracy [2][4]. It should be required for expressing user's initial information need as more descriptive terms.

A general model of information retrieval is the vector space model [2] that represents queries and documents as vectors of terms. Query modification consists of re-weighting, adding or removing terms, and inserting structure to the queries. Rocchio has suggested a relevance feedback method as follows [3]:

$$Q' = Q + \frac{1}{n_1}\alpha\sum_{i=1}^{n_1} R_i - \frac{1}{n_2}\beta\sum_{i=1}^{n_2} S_i$$

where

Q : the vector for the initial query,

R_i : the vector for relevant document i,

S_i : the vector for irrelevant document i,

α, β : Rocchio's weights.

Q' is a modified vector of the original query plus the vectors of the relevant and the irrelevant documents. Rocchio's weights are the factor used for adjusting the ratio of each component. The Rocchio algorithm is the basis of relevance feedback in IR of vector space model [2]. The original query vector is expanded to the query which is the vector of the original query plus the vectors of the relevant and the irrelevant documents. Most of the relevance feedback methods in the vector-space model are based on Rocchio's method. But these methods have several drawbacks. One of them is that they do not discriminate well which term is more relevant to the user's initial need because they make use of all the terms in the retrieved documents. Another is that the cost of vector sums is very high in relatively dynamic document sets like the Web. Therefore, it is difficult to be applied to information filtering on the Web.

To overcome these drawbacks of batch algorithms, there are on-line incremental algorithms, called WH (Widrow-Hoff) and EG (exponentiated gradient) [6][7]. The LMS and WH are learning algorithms that search the weight vector representing the user information need. WH is viewed as a gradient descent procedure since the term $2(W \cdot X - y)x$ is the gradient of the square loss. Thus, WH tries to move in a direction in which this loss is decreasing the fastest.

$$w_{i+1,j} = w_{i,j} - 2\eta(W_i \cdot X_i - y_i)x_{i,j}$$

where

W_i : the weight vector at ith iteration,

$w_{i,j}$: jth term in the W_i,

X_i : the ith document vector,

$x_{i,j}$: jth term in the X_i,

y_i : relevance feedback to the X_i.

The parameter $\eta > 0$, usually called the learning rate, controls how quickly the weight vector W is allowed to change, and how much influence each new example has on it.

The EG algorithm is similar to WH in that it maintains a weight vector W_i and runs through training examples one at a time.

$$w_{i+1,j} = \frac{w_{i,j}\exp(-2\eta(W_i \cdot X_i - y_i)x_{i,j})}{\sum_{j=1}^{d} w_{i,j}\exp(-2\eta(W_i \cdot X_i - y_i)x_{i,j})}$$

where

$$\eta = \frac{2}{3R^2},\ R \geq (\max_i x_{i,j} - \min_j x_{i,j}).$$

Since they can learn a linear classifier with less data and in online fashion, it is useful to apply these algorithms to personalized information filtering. But they mainly consider weight modification, and thus the results are affected by other factors such as term selection. Because EG follows an exponentiated gradient, small variations in how the gradient is determined might have large effects on algorithm behavior [7].

We exploit a method that reflects directly user's opinion to the term in the profile. That is, if a term in the document estimated as "relevant" is included in the user's profile, the term receives user's feedback as follows:

$$w_{pk} \leftarrow w_{pk} + \alpha r_i, \qquad \text{if } k \in D_i$$

where

r_i : relevance feedback to the filtered document D_i

w_{pk} : weight of kth term in profile p

Here α is the learning rate that controls learning speed. If a weight of a term is less than the threshold, the term is removed from the user's profile. As the more relevant documents are filtered by a term, the importance of the term increases.

So far, we described methods that modify the initial user information needs into more descriptive terms. These methods have a drawback that the user participates in relevance feedback himself. The more a filtering system gets user's opinions, the less it is not convenient to use. In the next section, we describe methods that get user's potential opinions by analyzing his behaviors.

2.2 Learning User's Preferences by Analyzing His Behaviors

A concept of an intelligent agent was proposed that can reduce information and work overload on behalf of users [9]. An intelligent agent "looks over his (or her) shoulder" his behaviors and learns his preferences. And then it takes the place of user's works or delivers more interesting information.

Letizia [10], which is an assistant for browsing the Web, traced the user behavior in the conventional Web browser. It analyzed his (or her) behaviors, such as following-up of the hyperlinks in an HTML document. And then it estimated his interests by parsing the document and recommending HTML documents. It has some benefits that satisfy the potential information need without using the retrieval tools such as WAIS or Web Crawler, and using Web-index services such as Yahoo! or AltaVista. And it also helps the user to persist his interest. But, learning about a person takes much time and there are many variables associated with the learning.

WebWatcher [11] learns the user interests using reinforcement learning as in WAIR. In WebWatcher, it is assumed that the information space is linked with hyperlinks. While the retrieval agent seeks the relevant document, it is directed by the value of reinforcement learning. It chooses one of the hyperlinks in the visited HTML document as the next.

$$Q_{n+1}(s,a) = R(s') + \gamma \max_{a' \in actions\_in\_s'} [Q_n(s',a')]$$

Here, the value of Q is the discounted sum of future rewards that will be obtained when the agent follows a hyperlink in an HTML document and subsequently chooses the optimal hyperlink. γ is a discount factor that regulates a ratio of the future reward. The agent, which is a reinforcement learner, will seek a policy maximizing the expected value of the future reward.

In [12], a system is described that learns user's profiles by analyzing user behaviors to the filtered net-news. The net-news system implemented in [12] observes several behaviors, for example, spending time on reading and following the hyperlinks in the filtered documents. They successfully estimate explicit user feedbacks by analyzing user's behaviors.

As described above, we analyze user's behaviors on the Web browser and then estimate explicit user's feedback $R_E(i)$ by using analyzed information. We refer to this type of feedback as implicit feedback. Implicit feedback $R_I(i)$ is obtained by observing user's behavior on the filtered ith document. It is composed of several parts: time for reading (rt), bookmarking (bm), scrolling thumb up and down (sl), and following up (fl) the hyperlinks in the filtered documents. The total score of implicit feedback is computed as:

$$R_I(i) = \sum_{v \in F} c_v f_v(i)$$

where

$$F = \{rt, bm, sl, fl\}.$$

Here, c_v is a weight for each factor of the implicit feedback. It is learned by using a multi-layer neural network.

3. INFORMATION FILTERING BY ANALYZING USER'S WEB-BROWSING BEHAVIORS

In this section, we describe a method that analyzes user's Web-browsing behaviors and learns user's preference by using reinforcement learning.

The method is implemented in a Web-based personalized information filtering system called WAIR (Web Agents for Information Retrieval).

WAIR consists of three agents: a user-interface agent, a Web-document retrieval agent, and a profile learning agent. The user-interface agent, shown in Figure 1, directly interacts with the user. Part A is an input board that gets user's information need and shows the status of filtering. Part B is for presenting the filtering results and getting the user's explicit feedback. Part C is a repository of bookmarking. Part D is a browser where WAIR observes the user's behaviors.

The missions of each agent are as follows.

- User-interface agent: It observes user's behaviors, for example, bookmarking, time for reading, scrolling up/down thumb, and following up the HTML document by way of "looking over the shoulder" of the user.
- Web-document retrieval agent: Getting the user's interests from the user-interface agent, it retrieves a set of candidate HTML documents. It sets a starting-point of retrieval using the meta-search.
- Learning agent: It learns user's profile by using reinforcement learning. Information needed for learning are supplied by the user-interface agent. It guides the retrieval direction of the retrieval agent. This means that the learning agent supplies the retrieval agent with relevance criteria, which is a modified profile. And it estimates an explicit user relevance feedback with a multi-layer neural network.

The agents are closely related with each other for personalized information filtering, but we do not focus on their interactions in this paper. Figure 2 shows the overall procedure of Web-document filtering.

The profile for a user consists of one or more topics p. Each topic represents user's specific information need. The profile p is represented as a vector of terms: $W_p = (w_{p1}, \ldots, w_{pk}, \ldots, w_{pn})$, where w_{pk} is the weight of k-th term and n is the number of terms used for describing the profiles. Topics are initially information need itself but they are expanded through incremental feedback. The ultimate goal of WAIR is to learn the profile of the user to filter documents that best reflect his (or hers) preferences. WAIR filters the Web-document by using conventional Web-index services. That is, it queries user's information need to the Web-index services and receives N URLs. After each of the retrieved HTML documents (D_i) is preprocessed (i.e., remove the stoplists and HTML tags), it is evaluated as follows:

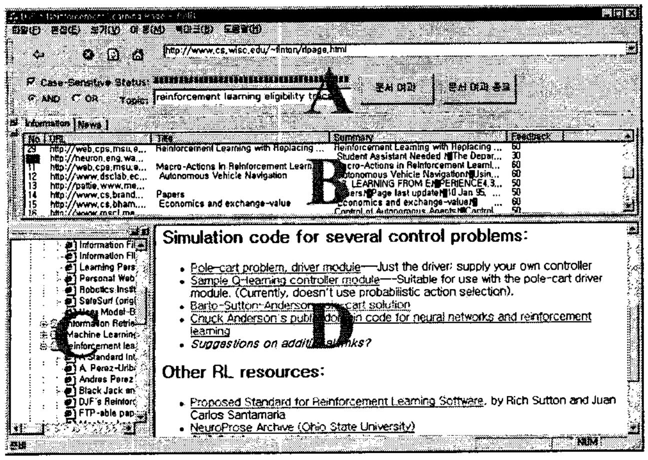

Figure 1: User-interface of WAIR.

$$RV_{D_i} = \sum_{k=1}^{n} tf_{ik} \times w_{pk}, \qquad \text{if } k \in D_i$$

where

tf_{ik} : term frequency of k th term in retrieved document D_i

k, w_{pk} : kth term and weight in the profile about p.

We do not use the general $tf \cdot idf$ (term frequency · inverse document frequency) [2] based indexing method in preprocessing the HTML documents. As we focus on filtering a dynamic information stream, it is difficult to define the static document sets. Since we also do not take a serious view of term's weight in the document, there are the term's weights only in the profile. The candidate documents are presented to the user, after sorted by the retrieval value (RV) in a descending order.

After the filtering, the user evaluates the results. WAIR acquires the user's explicit $R_E(i)$ or implicit $R_I(i)$ feedback by using equation in Section 2.2.

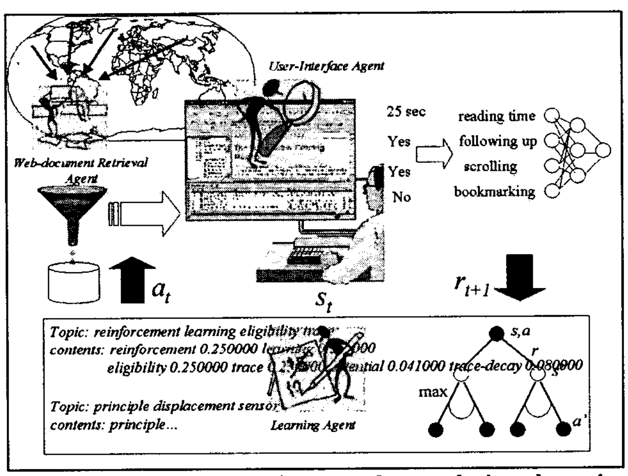

Figure 2: Learning preferences by analyzing browsing behaviors.

WAIR exploits reinforcement learning for filtering more personalized information. Reinforcement learning is a goal-directed learning method based on interactions with the environment [5]. The learner receives a scalar-valued feedback r_t, called reward, when it chooses and takes an action at given time t and a given state s_t. The objective is to maximize the expected value of the cumulative reward it receives in the long run from the environment [5][15]. According to the above description, the policy is the learner's way of behaving at a given time.

The goal of WAIR is to look for the best state of the profile. States in our problem are defined as a profile vector of terms and their weights. Accordingly, the best state of the profile can be interpreted as the most similar representation of the user's initial interests. Actions are defined as the method of picking up terms that participate in estimating the relevance between documents and profile. To find the state reflecting the user's interests well, we employ the ε -greedy terms selection method. Specifically, WAIR selects the m terms from the specific topic (p) in the profile, where $m - \varepsilon$ terms have been sequentially selected by the order of the highest weights and ε terms have been randomly selected. Since this selection method uses its current learned knowledge about user's interests (i.e. select the terms of highest weights), it can be described by the concept of exploitation [5]. And, it also gives a boost for profiles to find the terms to discriminate which documents are more relevant to the initial user's interests by the notion of exploration.

At time t, we define the scalar-valued feedback from the user as

$$r_i = \beta R_E(i) + (1-\beta) R_I(i),$$
$$0 \le R_E(i) \le 1, \quad 0 \le R_I(i) \le 1$$

where β is a regulating factor that adjusts the ratio of implicit and explicit feedbacks. At time t, the value $V_{p,t}$ of the profile about topic p is estimated by

$$V_{p,t} \leftarrow V_{p,t-1} + [R_t + \gamma \Delta V_{p,t}], \quad 0 \le \gamma < 1$$

where

$$R_t = \frac{1}{N}\sum_{i=1}^{N} r_i, \quad \Delta V_{p,t} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{n}(w_{pk,i} - w_{pk,i-1}).$$

γ is a discount factor that determines the present value of expected future reward. We approximate the value about the future reward as the change of term weights. The positive value of the change means that the current selected terms are good choices for representing the user's interest and the current contents of the profile will obtain the future positive feedback from the user.

At time t, the V value of profile p is the sum of the current value of evaluation from the user and the estimated value of future evaluation. That is, the learning agent bootstraps itself to more relevant point of the user's interests by considering the current reward and the estimated future reward. Thus, it directs the learning agent to the best point of the term space about user's specific interests.

4. EXPERIMENTAL RESULTS

Several experiments were carried out using the proposed method. The objective of the first experiment was to compare the performance of the proposed method with that of the conventional feedback methods. In this experiment, 10 people volunteered to suggest 30 topics. These 30 topics amount to a total of 15,000 HTML documents. 100 HTML documents were filtered per each of topics by using the four methods described Section 2.

Figure 3 and Table 1 show the result of the first experiment and parameters used. The method called "e-match" is a baseline method in which no relevance feedback is obtained from the user. It only follows up the hyperlinks, which are exactly anchored with user's initial query terms.

Feedback methods	Parameters	Term expansion
Rocchio	α=0.75, β=0.25	higher weight
WH	η=0.03	higher weight
EG	η=0.03	higher weight
WAIR	α=1, β=0.03, γ=0.9	m-ε: higher weight, ε: random

Table 1: Parameters used for various learning methods used in comparison. The parameter values were chosen from pre-experiments.

In the first experiment, the batch algorithms, such as Rocchio, did not learn well with less documents.

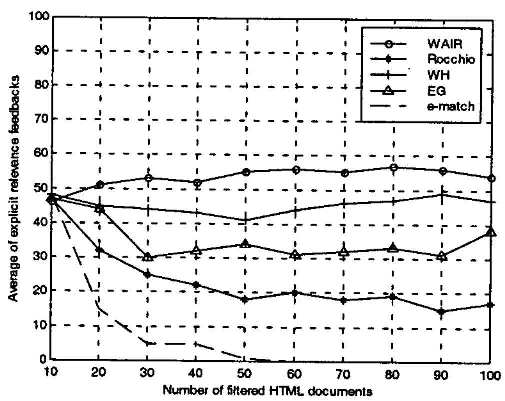

Figure 3: Comparison of the relevance with other relevance feedback methods by using explicit feedback. The X axis is the number of filtered documents and the Y axis is the average of explicit user relevance feedbacks.

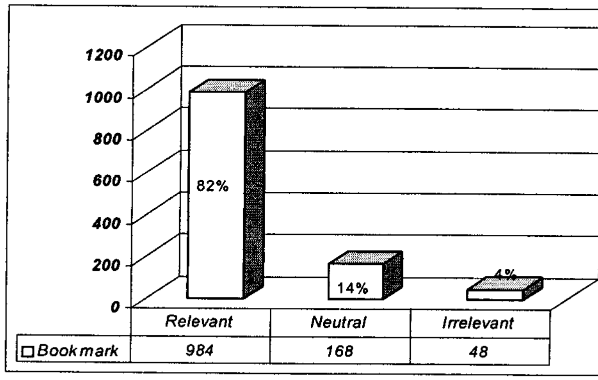

Figure 4: Correlation between bookmarking and relevance of filtered documents. Out of 15000 documents, a total of 1200 documents were related with bookmarking.

Because of using the final result after learning, they did not exploit experiences obtained during learning.

Since online incremental algorithms, such as WH, EG, and WAIR, learn through training examples one at a time, they outperformed batch methods. Setting up a goal as next state, the proposed method learns profile incrementally with experience. And it randomly selects terms that are components of the profile.

We analyzed the correlation of value's user behaviors to the relevance of documents. A statistical analysis of the results shows that the major factor of implicit feedback is the bookmarking of an HTML documents.

It means that the bookmarked URL reflects user's strong opinion of relevance. In Figure 5, we see that most participants scrolling up or down to evaluate the documents. Although the number of scrolled documents is important, scrolling itself does not affect user's opinion. There is also a tendency that the HTML documents on which the user spent a long time to read were rated as "relevant" and the documents for which only a short time is spent were evaluated as "irrelevant".

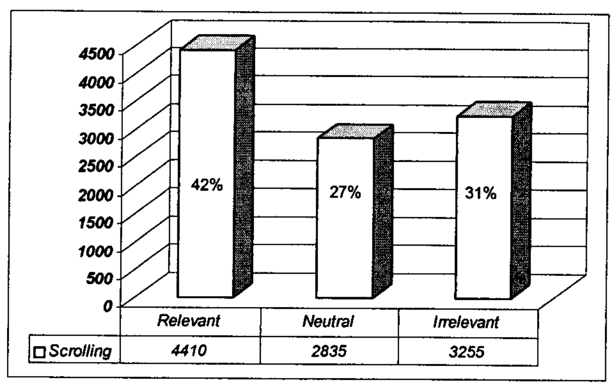

Figure 5: Correlation between scrolling and relevance of the filtered documents. Out of 15000 documents, a total of 10500 documents were related with scrolling.

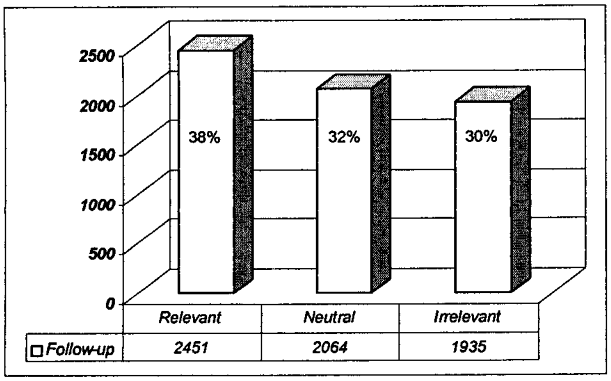

Figure 6: Correlation between follow-up and relevance of the filtered documents. Out of 15000 documents, a total of 6450 documents were related with follow-up.

To approximate user's explicit relevance feedback, we deployed a three-layer neural network. It consisted of 4 input nodes, 3 hidden nodes, and 1 output node. Its weight vector is trained by using the results of the first experiment to estimate user feedback. We performed next experiments to compare three online feedback methods taking sides view of the filtering accuracy and adaptation. This involved 5 people with each at a topic. 750 HTML documents were presented at a topic with each of tested methods. The number of HTML documents used for this experiment is 3,750.

As described earlier, user relevance feedback is implicitly obtained by a neural network. In a filtering trial, each method was presented 10 HTML documents. Figure 8 shows the results for 25 filtering trials. To verify the accuracy of implicit feedback, we asked to the participants to evaluate 25th filtered documents explicitly. The difference of feedback values, *Feedback$_{error}$*, is calculated by subtracting the implicit feedback from the explicit feedback.

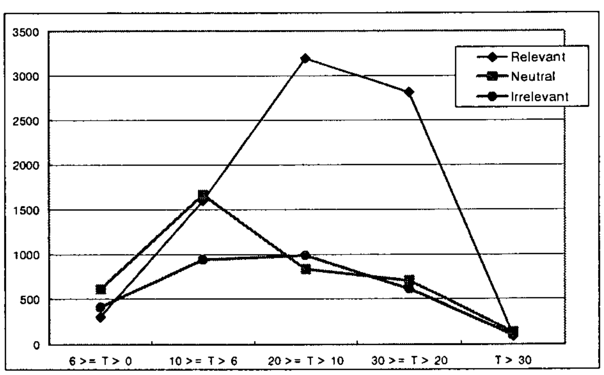

Figure 7: Correlation between reading time and relevance of the filtered documents. There is a tendency to spend a longer time to read HTML documents those evaluated as relevant.

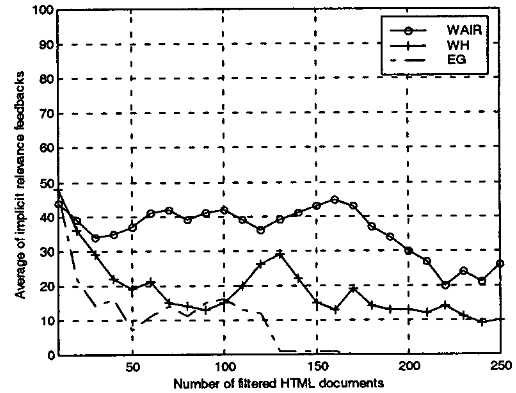

Figure 8: Comparison of the relevance with online incremental relevance feedback methods by using implicit feedback. The X axis is the number of filtered documents and the Y axis is the average of explicit user relevance feedback. *Feedback$_{error}$* = R_E - R_I, WAIR=-15, WH=-10, EG=0.

5. CONCLUSION

In this paper, we proposed a new method for learning user preferences that obtains the required information directly from user interactions. Through the experiments on a group of users, we verified that the method can provide documents which are more relevant to the user's specific interests when compared with other conventional feedback methods. It also effectively adapts to the user's specific interests with implicit feedback only. In terms of adaptation speed, the proposed method converged on the user's specific interest faster than existing relevance feedback methods. Based on the results, we can conclude that learning from the shoulders of the user can significantly improve the performance of personalized information filtering systems.

In spite of our success in learning the user preferences in the WAIR system, we should mention that the success comes in part from the environments where we made our experiments. One is that the topics used for experiments were usually scientific and thus the filtered documents contained relatively less-ambiguous terms than those that might be contained in other usual Web documents. Another reason might be that the duration of our experiments were not very long during which the user interests did not change very much. The adaptation to user's interests during a longer period of time in a more dynamic environment should still be tested.

From a more practical point of view, the response time is a crucial factor in the information retrieval and filtering. However, our focus in this paper has been confined to the relevance feedback. Learning from users to minimize their response time is one of our research topics in the future.

ACKNOWLEDGEMENTS

This research was supported in part by Korean Ministry of Information and Communication under Grant 98-199 through IITA (Institute of Information Technology Assessment) and by Korea Science and Engineering Foundation through AITRC (Advanced Information Technology Research Center).

REFERENCES

[1] Nicholas J. Belkin and W. Bruce Croft, Information filtering and information retrieval: two sides of the same coin?, *Communications of the ACM*, Vol. 35, No.12, pp. 29-38, 1992.

[2] G. Salton, *Automatic Text Processing*, Addison-Wesley, 1989.

[3] Rocchio, J. J, Relevance feedback in information retrieval, *The SMART Retrieval System*, pp. 313-323, Prentice Hall, 1971.

[4] Gerard Salton and Chris Buckley, Improving retrieval performance by relevance feedback, *Journal of American Society for Information Science* 41, pp. 288-297, 1990.

[5] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[6] David D. Lewis et al., Training algorithms for linear text classifier, In *Proceedings of Special Interest Group in Information Retrieval* (SIGIR-96), pp. 298-306, 1996.

[7] Jamie Callan, Learning while filtering documents, In *Proceedings of ACM Special Interest Group in Information Retrieval* (SIGIR-98), pp. 224-231, 1998.

[8] Masahiro M. and Yoichi S., Information filtering based on user behavior analysis and best match text retrieval, In *Proceedings of ACM Special Interest Group in Information Retrieval* (SIGIR-94), pp. 272-281, 1994.

[9] Pattie Maes, Agents that reduce work and information overload, *Communications of the ACM*, Vol. 37, No. 7, pp. 31-40, 1994.

[10] Henry Lieberman, Letizia: An agent that assists web browsing, In *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI-95), pp. 475-480, 1995.

[11] Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell, WebWatcher: A tour guide for the world wide web, In *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI-97), pp. 770-775, 1997.

[12] Masahiro M., and Yoichi S., Information filtering based on user behavior analysis and best match text retrieval, In *Proceedings of ACM Special Interest Group in Information Retrieval* (SIGIR-94), pp. 271-281, 1994.

[13] Foss, C. L., Tools for reading and browsing hypertext. *Information Processing and Management*, Vol. 25, pp. 407-418, 1989.

[14] Alexandros Moukas, Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem, In *Practical Applications of Intelligent Agents and Multi-Agents Technology* (PAAM-96), pp. 421-437, 1996.

[15] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.