Personalized Web-Document Filtering Using Reinforcement Learning

Byoung-Tak Zhang and Young-Woo Seo
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
E-mail: {btzhang, ywseo}@scai.snu.ac.kr

Abstract- Document filtering is increasingly deployed in Web environments to reduce information overload of users. We formulate online information filtering as a reinforcement learning problem, i.e. TD(0). The goal is to learn user profiles that best represent his information needs and thus maximize the expected value of user relevance feedback. A method is then presented that acquires reinforcement signals automatically by estimating user's implicit feedback from direct observations of browsing behaviors. This "learning by observation" approach is contrasted with conventional relevance feedback methods which require explicit user feedbacks. Field tests have been performed which involved 10 users reading a total of 18,750 HTML documents during 45 days. Compared to the existing document filtering techniques, the proposed learning method showed superior performance in information quality and adaptation speed to user preferences in online filtering.

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 [Belkin and Croft, 1996]. 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 manually follow or browse the hyperlinks of the documents by a user himself. However, these methods have some drawbacks. Since Web-index services are based on general purpose indexing methods, much of the retrieval results may be irrelevant to user's interests. In addition, manual browsing involves much time and efforts. High-quality information services require to capture the personal interests of individual users during the interaction with the information retrieval systems.

Several methods have been proposed to reflect user preferences. A classical approach is the Rocchio method [Rocchio, 1971] and its variants. This is a batch algorithm that modifies the original query vector by the vectors of the relevant and irrelevant documents. However, the batch algorithms tend to put large demands on memory and are slow in adaptation, thus not well suited to on-line applications. Recently, several on-line learning algorithms have been used for information retrieval and filtering. These include the Widrow-Hoff rule [Lewis et al., 1996] and the exponentiated gradient algorithm [Callan, 1998]. These algorithms learn training examples one at a time and thus more appropriate for learning in online fashion. However, all these methods have a drawback that the user has to provide explicit relevance feedback for the system to learn. Since providing relevance feedbacks is a tedious process and users may be unwilling to provide them, the learning capability of the filtering systems may be severely limited.

In this paper, we present a personalized information filtering method that learns user's interests by observing his or her behaviors during the interaction with the system. First, the system is trained on the explicit feedback from the user. After this learning phase, the system estimates the relevance feedback implicitly based on the observations of user actions. This information is used to modify the user profiles. We regard filtering as a goal-directed learning process based on interactions with the environment. The objective is to maximize the expected value of the cumulative relevance feedback it receives in the long run from the user. This process is formulated as TD(0) learning, a general form of reinforcement learning [Sutton and Barto, 1998]. In this formulation, filtering is viewed as an interactive process which involves a generate-and-test method whereby the agent try actions, observe the outcomes, and selectively retain those that are the most effective. The advantage of TD(0) over other reinforcement learning methods is that it can learn without excessive delay of rewards. This is an important property in real-time interactions with the user in Web browsing environments. Additional feature of our approach is that it is learning by experimentation, in contrast to learning by instruction as adopted in most supervised learning methods. The method was implemented as WAIR (Web Agents for Information Retrieval), a platform for Web-based personalized information filtering services [Seo and Zhang, 2000].

The paper is organized as follows. In Section 2, we review previous approaches to information filtering and relate them to our work. Section 3 describes the overall structure of WAIR and presents the reinforcement learning formulation of the information filtering problem. Section 4 details the process for detecting and learning user preferences to personalize Web-document filtering. Section 5 provides the experimental results and compares them with those of existing methods. Section 6 discusses the results and further work.

2 Related Work

A general model for information retrieval is the vector space model that represents queries and documents as vectors [Salton, 1989]. Most of the relevance feedback methods in the vector-space model are based on the Rocchio's algorithm [Rocchio, 1971]. Here, the original query is modified by increasing the weights of terms that appear in the relevant documents and by decreasing the weights of terms that appear in the irrelevant documents:

$$\mathbf{q}' = \mathbf{q} + \frac{1}{|D_R|}\alpha \sum_{i \in D_R} \mathbf{x}_i - \frac{1}{|D_I|}\beta \sum_{j \in D_I} \mathbf{x}_j, \tag{1}$$

where $\mathbf{q}$ is the vector for the initial query, D_R (resp. D_I) is the index set for relevant (resp. irrelevant) documents, and α and β are Rocchio's weights. $\mathbf{x}_i$ is the vector for relevant document *i*, $\mathbf{x}_j$ is the vector for irrelevant document *j*, and the summation symbol denotes vector summation. However, the Rocchio algorithm updates the queries in batch mode, i.e. the update is based on a collection of documents. Batch learning requires a large amount of memory and is slow in adaptation, and thus not very appropriate for on-line information services on the Web.

To overcome these drawbacks of batch algorithms, on-line incremental algorithms have recently been proposed. Examples are WH (Widrow-Hoff) and EG (exponentiated gradient) [Lewis et al., 1996][Callan, 1998] algorithms. The LMS or WH is a supervised learning algorithm that learns classification of documents into prespecified classes. Given a set of document and relevance-label pairs, $(\mathbf{x}_i, r_i)$, it searches the weight vector representing the classification rule. WH is a gradient descent procedure that tries to minimize the squared error of classification: $\| \mathbf{x}_i - r_i \|^2$. The learning rule is given as:

$$w'_k = w_k - 2\eta(\mathbf{w} \cdot \mathbf{x}_i - r_i)x_{i,k}, \tag{2}$$

where w_k, $k = 1, ..., d$, is a component of the weight vector $\mathbf{w}$, $\mathbf{x}_i$ is the *i*th document vector, and r_i is the correct class of document i. The parameter $\eta > 0$, usually called the learning rate, controls how quickly the weight vector $\mathbf{w}$ 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 $\mathbf{w}$ and runs through training examples one at a time. With EG, however, the components of weight vector are restricted to be non-negative and sum to one. The weight update rule is expressed as:

$$w'_k = \frac{w_k \exp\{-2\eta(\mathbf{w} \cdot \mathbf{x}_i - r_i)x_{i,k}\}}{\sum_{k=1}^{d} w_k \exp\{-2\eta(\mathbf{w} \cdot \mathbf{x}_i - r_i)x_{i,k}\}}, \quad (3)$$

where $\eta = \frac{2}{3K^2}$ and K is a value that satisfies the constraint $K \geq (\max_i\ x_{i,k} - \min_k\ x_{i,k})$. EG has the characteristic that the terms that have large errors are exponentially reflected in weight modification.

Since the WH and EG algorithms can learn a linear classifier in online fashion, it is useful to apply these algorithms to information filtering. But these methods have some drawbacks. One is that their learning is inherently iterative and typically requires a large number of cycles. In addition, since all the terms in the retrieved documents are used for document representation, and these supervised learning methods tend to use all the given terms, a large number of documents are required to distinguish relevant terms from irrelevant terms with respect to the user's interest. In contrast, the profile update method adopted in WAIR restricts the size of profile and directly reflects user's opinion in the profile by explicitly adding new terms, removing existing terms, and updating term weights.

All the methods described above have drawbacks. One is the user has to participate in relevance feedback himself. The more a filtering system gets user's opinions, the less convenient the system is to use. The other is the general assumption that they usually concern the document set of static nature. But, the nature of Web is dynamic rather than static. In this case, it is more useful to introduce the concept of filtering than retrieval. WAIR presents a method that gets user's potential preferences by observing his behaviors during the interaction with the information filtering system.

Several studies have been made to release the burden of explicit user's participation in finding the information on the Web. Letizia [Lieberman, 1995], which is an assistant for browsing the Web, traced the user behavior in the conventional Web browser. It analyzes his (or her) behaviors, such as following-up the hyperlinks in an HTML document. And then it estimates his interests by parsing the document and recommending HTML documents. ANTAGONOMY [Kamba et al., 1997] suggested methods by which user preferences for the electronic news articles can be learned from user behaviors. They have exploited two types of inference: one using explicit feedback and the other using implicit feedback. In the explicit relevance feedback, the users rate all articles according to their relevance. In the implicit, the users read articles by performing scrolling and enlarging the articles, and the system infers from the behaviors how much the user was interested in each article. [Morita and Shinoda, 1994] exploited a heuristic, which uses behavior monitoring to capture the user's interests in information, for filtering the news articles. They have determined whether a user is interested in an article or not by measuring the time to read it. MAXIMS [Lashkari et al., 1994] classifies the stream of e-mail after observing how a user chooses to deal with e-mail.

3 Personalized Filtering as Reinforcement Learning

3.1 Information Filtering in WAIR

WAIR (Web Agents for Information Retrieval) was originally designed as a platform for the development of personalized information services on the Web. WAIR consists of three agents: an interface agent, a retrieval agent, and a filtering agent. The interaction between the agents is illustrated in Figure 1. The overall procedure is summarized in Figure 2.

Initially, the user provides the system with a profile (Step 1). Typically, the initial profile consists of a few keywords. Then, the retrieval agent constructs a query using the profile and get N URLs (Step 2). Existing Web search engines are used to obtain the relevant URLs. The documents for the URLs are then

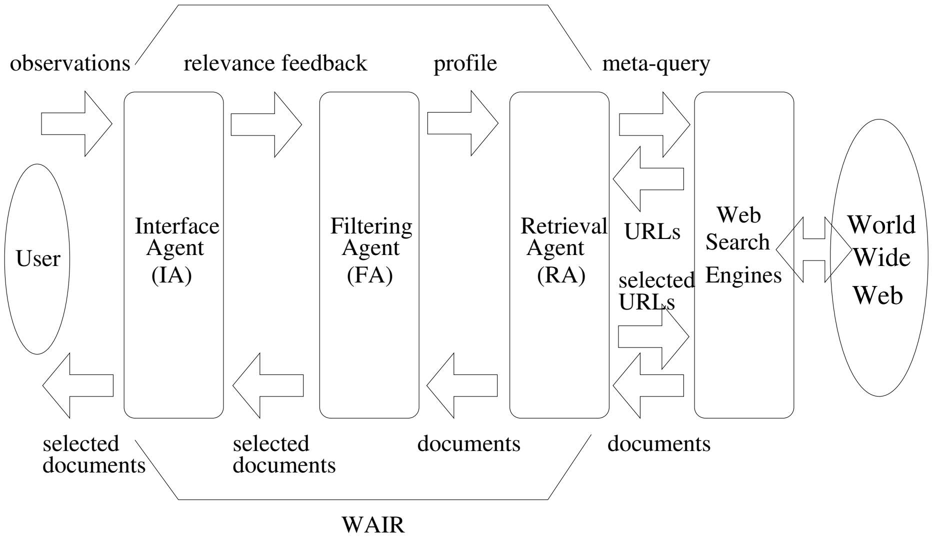

Figure 1: System architecture of WAIR.

1. Get the initial profile from the user. Set $t \leftarrow 0$.
2. (Retrieval) Generate a query from the profile to retrieve N URLs.
3. (Filtering) Evaluate the relevance of documents.
   Rank the N documents and present M of them to the user.
4. (Interface) Get the feedback by observing user behavior.
5. (Learning) Update the user profile.
6. Set $t \leftarrow t + 1$. Go to step 2.

Figure 2: The overall procedure of WAIR.

retrieved and preprocessed, and their relevance values are estimated. The N documents are ranked, and M of them are filtered and presented to the user (Step 3). To balance exploration and exploitation, WAIR chooses the highest-ranked documents most of the time, but occasionally (with probability ϵ) it filters lower-ranked documents.

The interface agent observes user behavior and measures user feedback (Step 4). Two different types of user feedbacks are distinguished in WAIR. One is the “explicit” feedback in the form of scalar values to evaluate the relevance of the documents. This is provided by the user during the initial learning phase. A second type of feedback is the “implicit” feedback. This is not provided by the user, but estimated by the interface agent in WAIR . That is, the users read filtered HTML documents by performing normal browsing behaviors, such as scrolling thumb up and down, bookmarking an URL, following the hyperlinks in the filtered document, and the WAIR infers from the behaviors how much the user was interested in each filtered document with a multi-layer neural network. This process is described in detail in the next section. The feedback information is then used to update the user profile (Step 5). Basically, this consists of inserting new terms, removing existing terms, and adjusting term weights of profile terms using the terms in the relevant/irrelevant documents. Then, the revised profile is used to get new documents by going to the retrieval step. Note that, the user provides only an initial query and then WAIR automatically retrieves and filters documents by oberving user behaviors implicitly.

3.2 Filtering as Reinforcement Learning

The task of information filtering in WAIR is formulated as a reinforcement learning problem. Reinforcement learning is about learning from interaction how to behave in order to achieve a goal. The reinforcement learning agent and its environment interact over a sequence of discrete time steps. The actions are the choices made by the agent. The states are the basis for making the choices. The rewards are the basis for evaluating choices. In WAIR, actions are defined as the decision-making as to whether to present the document to the user or not. States are defined as the pairs of the profile and the document to be filtered.

The policy is a stochastic rule by which the agent selects actions as a function of states. Formally, a policy π is a mapping from each state s and action a to the probability $\pi(s, a)$ of taking action a when in state s. We use an ϵ-greedy policy for choosing an action given a state. That is, most of the time WAIR chooses the highest-ranked documents, but with probability ϵ, it chooses lower-ranked documents too. The rationale behind this policy is that it combines exploitation and exploration of search behavior. The selection of documents with the highest relevance value corresponds to exploitation of known information, while selecting random documents encourages exploration of unknown regions to find interesting documents which are unexpected by the user. An advantage of the ϵ-greedy method is that, in the limit as the number of actions increases, the probability of selecting the optimal action converges to greater than $1-\epsilon$, i.e., to near certainty [Sutton and Barto, 1998].

The filtering agent's objective is to maximize the amount of reward it receives over time. The return is the function of future rewards that the agent seeks to maximize. Value functions of a policy assign to each state, or state-action pair, the expected return from that state, or state-action pair, the largest expected return achievable by any policy. The agent tries to select actions so that the sum of the discounted rewards it receives over the future is maximized. In particular, it chooses action a_t to maximize the expected discounted return:

$$\begin{aligned} R_t &= r_{t+1} + \gamma r_{t+2} + +\gamma^2 r_{t+3} \cdots \\ &= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \end{aligned}$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the discount rate.

To make decisions on whether or not filter the documents, it is necessary to estimate value functions, i.e., functions of states that estimate how good it is to be in a given state. The notion of how good here is defined in terms of future rewards that can be expected, i.e. in terms of expected return. Value functions are defined with respect to particular policies. Informally, the value of a state s under a policy π, denoted $V^\pi(s)$, is the expected return when starting in s and following π thereafter. We can define $V^\pi(s)$ as

$$\begin{aligned} V^\pi(s) &= E_\pi\left\{R_t \mid s_t = s\right\} \\ &= E_\pi\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s\right\} \\ &= E_\pi\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_t = s\right\} \\ &= E_\pi\left\{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\right\}, \end{aligned}$$

where $E_\pi\{\}$ denotes the expected value given that the agent follows policy π. Temporal difference (TD) learning, a form of reinforcement learning, uses an estimate of $(V^\pi(s))$ as a target. Because $V^\pi(s_{t+1})$ is not known, it uses the current estimate $V_t(s_{t+1})$ instead. In procedural form, the update rule for the state-value function is expressed as:

$$V_{t+1}(s_t) \quad = \quad V_t(s_t) + \alpha[r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)],$$

where s_t is the state, γ is a discount factor which determines the present value of the expected future reward, and r_{i+1} denotes the immediate reward due to filtering document i. This recurrence relationship indicates the theoretical target that the WAIR learning procedure has to attempt to reach. That is, the equation reaches a fixed point when $r_{t+1} + \gamma V_t(s_{t+1})$ equals to $V_t(s_t)$, i.e., the sum of the reward and the discounted expected reward of the next state becomes the same as the value of the current state.

It should be mentioned that WebWatcher [Joachims et al., 1997] learns the user interests using reinforcement learning like in WAIR. In WebWatcher, it is assumed that the information space is linked with hyperlinks. While the retrieval agent seeks the relevant documents, it is directed by the value of reinforcement learning:

$$Q_{t+1}(s,a) = R(s') + \gamma \max_{a' \in actions\_in\_s'} [Q_t(s',a')].$$

Here, *Q*-value is the discounted sum of the future rewards that will be obtained when the agent follows a hyperlink in an HTML document and subsequently chooses the optimal hyperlink. Note that WebWatcher is in contrast with WAIR in several points. While the objective of WebWatcher is to find interesting sites (a retrieval agent), the aim of WAIR is to filter a stream of documents that are relevant to user preferences (a filtering agent). Thus, the actions in WAIR are defined as the decision making whether or not to present documents to the user, while the actions in WebWatcher is the decisions as to follow the links or not. As shown above, the learning process in WAIR is formulated as TD(0) learning while WebWatcher is best formulated as Q-learning. While Q-learning primarily concerns selecting the most promising action in the given state, TD(0) is more general than Q-learning in that it deals with the value of the state. In WAIR, we seek the state of the profile that reflects the user's information needs well. Thus, our problem is more naturally formulated as TD(0).

4 Learning Profiles from Implicit Feedbacks

In this section, we first describe the retrieval of documents in WAIR. Then, the procedures for estimating user feedbacks and updating user profiles are described.

4.1 Document Retrieval

The task of the retrieval agent is to get a collection of candidate HTML documents to be filtered. The retrieved documents undergo preprocessing. We use standard term-indexing techniques, such as removing stop-words and stemming [Frakes and Baeza-Yates, 1992]. Formally, a document is represented as a term vector $\mathbf{x}_i$:

$$\mathbf{x}_i \quad = \quad (x_{i1}, x_{i2}, ..., x_{i,k}, ..., x_{i,d}), \tag{4}$$

where $x_{i,k}$ is the numeric value that term k takes on for document i, d is the number of terms used for document representation. In this work, we assume that $x_{i,k}$ represents the normalized term frequency, i.e. $x_{i,k}$ is proportional to the number of term k appearing in document i and $\| \mathbf{x}_i \| = 1$. This is contrasted with the usual *tf* · *idf* (term frequency · inverse document frequency) [Salton, 1989] based indexing method in conventional information retrieval. We use only tf information because we focus on information filtering from a stream of Web documents. In contrast to the conventional information retrieval environments where the collection of documents are static over a long period of time, our situation addresses a dynamically changing environment. In this dynamic environment, the inverse document frequency (which is computed with respect to a static collection of documents) is not significant.

The ultimate goal of WAIR is to filter documents that best reflect user's preferences. This is done by learning the profiles of users. A user profile consists of one or more topics. Topics represent user's information needs. In this section, we assume for simplicity that a profile consists of a single topic. The method can readily be generalized to multiple topics for a user by maintaining multiple profiles. Formally, the profile p is represented as a weight vector $\mathbf{w}_p$:

$$\mathbf{w}_p \quad = \quad (w_{p,1}, w_{p,2}, ..., w_{p,k}, ..., w_{p,d}), \tag{5}$$

where $w_{p,k}$ is the weight of the kth term in the profile and $\| \mathbf{w}_p \| = 1$. d is the number of terms used for describing the profiles. Formally, it is the same as the number of terms for representing documents. In WAIR, however, the maximum number of non-zero terms in the profile is limited to $m < d$. This is useful for concise description of user interests. Initially, the profile $\mathbf{w}_p$ contains only a small number of non-zero terms that are contained in the original user query. The subsequent retrieval and user-feedback process expands and updates the number and weights of the profile terms, as described below.

WAIR searches the Web-documents by using existing Web-index services, i.e. AltaVista, Excite, and Lycos. That is, it formulates a query $\mathbf{q}_p$ that is forwarded to one or more Web search engines. Queries are constructed by choosing terms from the profile based on an ϵ-greedy selection method. The retrieval agent then selects *N* URLs from different engines and ranks them. The rank of document i for profile p is based on its similarity (or relevance) to the profile and computed as the inner product:

$$V(s_i) \quad = \quad \mathbf{w}_p \cdot \mathbf{x}_i = \sum_{k=1}^{d} w_{p,k} x_{i,k}, \tag{6}$$

where $w_{p,k}$ and $x_{i,k}$ are the kth terms in profile p and document, respectively. The candidate documents are then sorted in descending order of $V_i(s_i)$, and M of them are presented to the user. Note that since the term vectors are normalized to $\mathbf{w}_p = 1$ and $\mathbf{x}_i = 1$, the relevance value is equivalent to the cosine correlation, i.e.

$$V(s_i) \quad = \quad \frac{\mathbf{w}_p \cdot \mathbf{x}_i}{\| \mathbf{w}_p \| \| \mathbf{x}_i \|} \tag{7}$$

where $\| \mathbf{x}_i \| = \sqrt{\sum_{k=1}^{d} x_{i,k}^2}$.

4.2 Estimating Implicit Feedbacks

The interface agent presents the retrieval results to the user. It also observes user's behavior by "looking over his (or her) shoulder" [Maes, 1994] to learn his interstes. Figure 3 shows the user interface of the WAIR system. It has three window frames. Part A is the input board that gets user's query, filtering conditions and shows the status of filtering procedure. Part B is for presenting the filtering results and getting the user's explicit feedback. Part C is a repository of bookmarks. And Part D a browser where the agent observes the user's behavior.

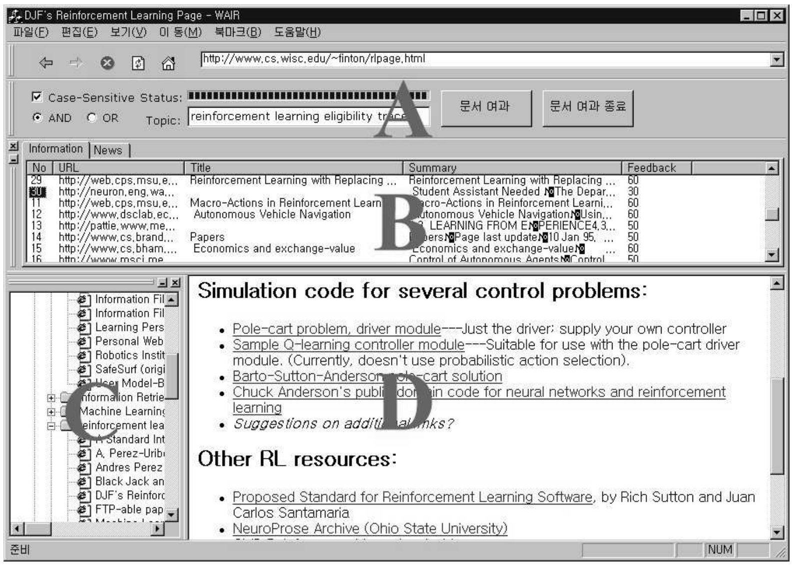

Figure 3: The user interface of WAIR.

Once M documents are filtered and presented, the user reads or browses (or ignores) the documents. For a document $\mathbf{x}_i$ presented to the user, WAIR measures a scalar-valued feedback by observing user behaviors as:

$$r_i \quad = \quad \delta R_E(i) + (1-\delta) R_I(i), \tag{8}$$

where $R_E(i)$ is an explicit feedback and $R_I(i)$ is an implicit feedback for document i. δ is a regulating factor that adjusts the ratio of implicit and explicit feedback. If δ is zero, implicit feedback is only used. The values are normalized to $0 \leq R_E(i) \leq 1$ and $0 \leq R_I(i) \leq 1$. The parameter δ controls the relative contribution of each feedback.

The explicit feedback is provided by the user as a real value in interval $[0, 1]$ while or after he reads the document. This feedback type is used in an early stage of interaction between the user and WAIR. After some interactions with the user, WAIR transfers to an implicit feedback mode in which the user does not need to give explicit feedback for the presented documents. The implicit feedback is measured automatically by WAIR without explicit help from the user. This can be done by analyzing user's behaviors on the documents filtered. Several factors can be measured. In this work, we distinguish four factors: reading time (*rt*), bookmarking (*bm*), scrolling (*sc*), and following up (*fl*) the hyperlinks in the filtered documents. The total score of implicit feedback is computed as:

$$R_I(i) \quad = \quad \sum_{v \in F} c_v f_v(i), \tag{9}$$

where $F = \{bm,\ fl,\ rt,\ sc\}$ is the set of implicit feedback factors, and c_v were the weight for each factor. The weight values c_v are determined by explicit feedback sessions during pre-experiments.

4.3 Updating User Profiles

The filtering agent evaluates the similarity between the documents retrieved and the user preferences to choose a subset of documents that best reflects user interests. User's preference is represented as a profile as described above. The profile is updated by adding new terms, removing existing terms, and modifying term weights. The update is based on the reward r_i. Formally, all this process can be expressed as a single learning rule:

$$w_{p,k}^{(i+1)} = w_{p,k}^{(i)} + \beta r_i I(x_{i,k}), \tag{10}$$

where $w_{p,k}^{(i)}$ is the term weight used for retrieving the ith document. $I(x)$ is a linear threshold function defined as:

$$I(x) = \begin{cases} +1 & \text{if } x \geq \theta_H \\ 0 & \text{if } \theta_L \leq x < \theta_H \\ -1 & \text{if } x < \theta_L, \end{cases} \tag{11}$$

where θ_H and θ_L are thresholds with $\theta_H > \theta_L$. According to this rule, a profile term gets its weight increased by a factor of relevance score r_i if the term appears in the relevant document. On the other hand, the terms get its weight increased by a factor of relevance score r_i if the term appears in the non-relevant document. It should be noted that r_i may be the implicit feedback only, estimated by equation (8). In vector form, the profile is updated as

$$\mathbf{w}_p^{(i+1)} = \mathbf{w}_p^{(i)} + \beta r_i I(\mathbf{x}_i), \tag{12}$$

where $I(\cdot)$ is now defined for a vector argument.

5 Experimental Results

The performance of the proposed filtering method was experimentally evaluated. We made two different sets of experiments. The objective of the first experiment was to compare the performance of the proposed method with 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. For each topic, 100 HTML documents were filtered by different relevance feedback methods: Rocchio, WH, EG, and WAIR. All the methods used the same retrieval engine built in WAIR. The user was presented 10 new documents in each session, and a total of 10 sessions were repeated for each user. This results in 100 different HTML documents filtered in total for each topic.

Table 1 summarizes the parameter values used for each algorithm. We also compared the performance of "e-match". This is used as a baseline method in which no relevance feedback is obtained from the user. It only follows up the hyperlinks that exactly matches the terms in the user's initial query. We attempted to use the parameter values as fair as possible. The parameter values for the Rocchio algorithm were as recommended in the literature [Salton and Buckley, 1990].

Figure 4 and Table 2 shows the results of various relevance feedback methods when explicit feedback was used. The graphs clearly show that online learning algorithms, such as WH, EG, and WAIR, consistently better reflect user's preferences than the batch algorithms, such as Rocchio. Since there is no query expansion, the filtering accuracy of "e-match" is decreased during all the experiment. Among the online algorithms, WAIR consistently achieved better relevance evaluations from the users. One reason for the performance difference is that EG and WH use all the terms for query construction while WAIR chooses important terms from the profile to construct queries. Since EG and WH use the profile vector directly to match the candidate documents, the focus is highly distributed to all the terms. This might work well for long-term experiments, but not very appropriate for a dynamic environment which requires short-term adaptation. In contrast, WAIR uses only selected terms according to the ϵ-greedy selection, which rapidly adapts to the current interests of the user. To verify the statistical significance of the experimental results, we have conducted paired-t tests. Tables 3 reports several statistics for the results of explicit feedback experiments. The proposed method (WAIR) is compared with each of the conventional algorithms with 99 degrees of freedom. Since the performance is statistically significant with 5% of significance level, we can say that the performance of WAIR significantly differs from that of WH. The difference of WAIR and other methods are especially evident from the excessively small P-values.

We analyzed the different user behaviors on the estimation of the relevance of documents. Figures 5–7 show the correlation between each behavior and the relevance of documents retrieved. It can be seen that bookmarking reflects user's interest most strongly. Other results show that following-up the hyperlinks does not always mean that the document is relevant. Users tend to follow up every document before they finally decide if the document is relevant or not. Similarly, scrolling is not a very strong indicator for relevance of documents, though this is a stronger indicator than the following-up behavior. Reading time seems a good indicator for user's interest. Most of the users spent 10 to 30 seconds on relevant documents while they spent 6 to 20 seconds on irrelevant or neutral documents. Thus, reading a document for 20-30 seconds is a good indicator for relevance. However, there is some ambiguity around 10 seconds. In general, it can be said that there is 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 was spent were evaluated as "irrelevant".

To build a model of user's explicit relevance feedback, we trained a three-layer neural network. It consisted of 4 input units, 3 hidden units, and 1 output unit. Its weight vector was learned by using the data collected from the first experiment in which users provided explicit feedbacks.

The second experiment was performed to compare the performance of three online feedback methods: WAIR, WH, and EG. We measured the filtering accuracy and adaptation speed when the user does not provide explicit feedback. The learner should estimate the user interests by observing the browsing behaviors. This experiment involved five people, each on a topic. Each method was tested on a topic using 750 HTML documents. The total number of HTML documents used for this experiment was 3,750.

User relevance feedback was implicitly obtained using the neural network trained through the browsing history of the explicit feedback experiment. In each filtering step, each method was presented 10 HTML documents. Figure 9 and Table 4 show the results for the methods during the 25 filtering steps. Though the absolute performance was lower than for the case of explicit feedback, the result shows a similar tendency. WAIR achieved better relevance values than the other methods. Among WH and EG, WH was better than EG. The accuracy of implicit feedback was confirmed by asking the participants to evaluate the documents presented at the end of the trials. Table 5 shows the results of paired-t tests of the filtering task with im-

Learning methods	Parameters	Term expansion
Rocchio	$\alpha = 0.75, \beta = 0.25$	higher weights
WH	$\eta = 0.03$	higher weights
EG	$\eta = 0.03$	higher weights
WAIR	$\alpha = 1, \beta = 0.03, \gamma = 0.9$	$m - \epsilon m$: higher weights, ϵm: random

Table 1: Parameters used for the experiments.

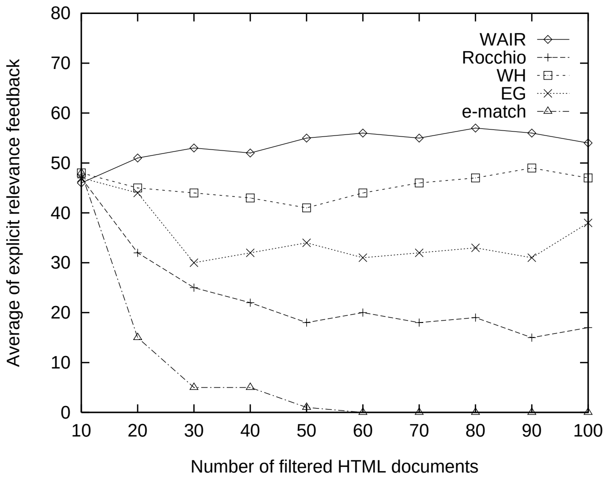

Figure 4: Results for the explicit feedback experiment. The X-axis shows the number of filtered documents. At each session, the user was presented 10 documents which were not presented in the previous sessions. The Y-axis denotes the average of explicit relevance feedback value scaled to [0, 100]. Each graph shows the evolution of the average explicit-feedback values as the filtering steps proceed. Each time 30 documents were retrieved using three different search engines, and 10 of them are filtered. The online learning algorithms, especially WAIR and WH, maintain a certain level of filtering performance though the number of filtering steps increased. In contrast, the filtering performance of the e-match and Rocchio algorithms tend to decrease rapidly as the filtering session goes on. See text for explanation of the results.

Feedback Iteration	average feedback ± standard deviation				
	WAIR	Rocchio	WH	EG	e-match
1	0.46±0.019	0.47±0.024	0.48±0.025	0.47±0.033	0.048±0.028
2	0.51±0.024	0.32±0.021	0.45±0.036	0.44±0.019	0.15±0.032
3	0.53±0.028	0.25±0.019	0.44±0.025	0.3±0.022	0.5±0.03
4	0.52±0.03	0.22±0.028	0.43±0.022	0.32±0.021	0.5±0.024
5	0.55±0.037	0.18±0.033	0.41±0.033	0.34±0.032	0.1±0.027
6	0.56±0.025	0.2±0.022	0.44±0.034	0.31±0.03	0
7	0.55±0.028	0.18±0.032	0.46±0.032	0.32±0.023	0
8	0.57±0.029	0.19±0.028	0.47±0.026	0.33±0.033	0
9	0.56±0.031	0.15±0.032	0.49±0.031	0.31±0.026	0
10	0.54±0.03	0.17±0.03	0.47±0.026	0.38±0.037	0

Table 2: Results for the explicit feedback experiments.

Learning methods	Average	Standard deviation	Number of documents	t-statistics	P(T $\leq$ t)
WAIR	0.533	0.0499	3000	–	–
Rocchio	0.234	0.0678	3000	0.1507	5.13×10^{-17}
WH	0.454	0.0678	3000	0.3608	2.01×10^{-2}
EG	0.352	0.0672	3000	0.0748	1.82×10^{-6}
e-match	0.074	0.0286	3000	0.5176	2.92×10^{-28}

Table 3: Paired-*t* test for the results of explicit feedback.

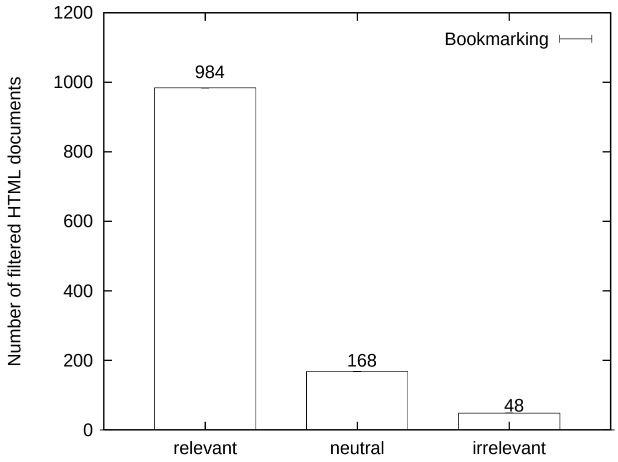

Figure 5: Correlation between the bookmarking behavior and the relevance of filtered documents. Bookmarking was observed 1200 times out of 15000 documents. This bar graph shows that many of the documents were relevant when the user bookmarked them.

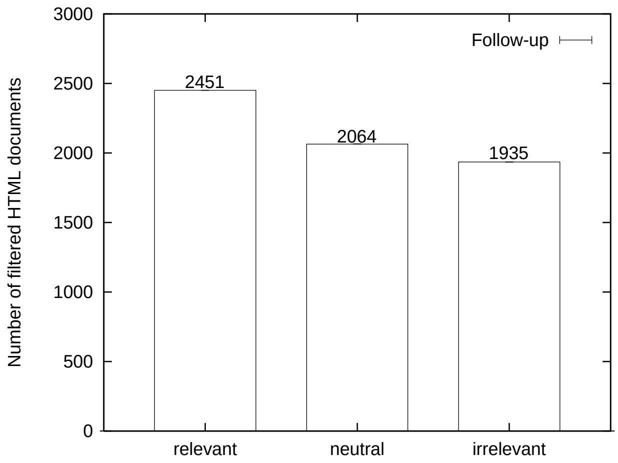

Figure 6: Correlation between the follow-up behavior and the relevance of filtered documents. The follow-up behavior was observed 6450 times out of 15000 documents. The results indicate that the users tend to follow-up every document irrespective of its relevance.

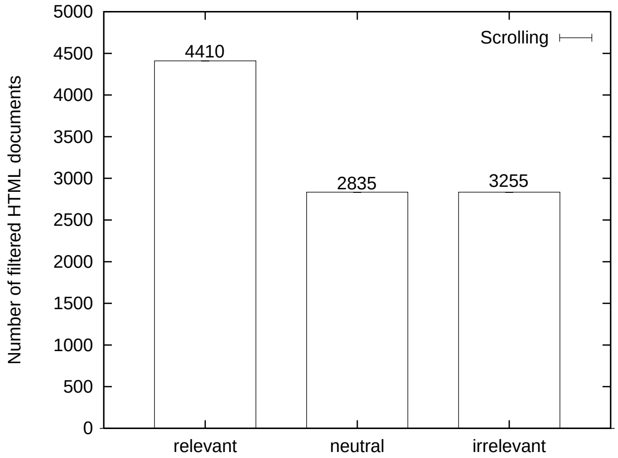

Figure 7: Correlation between the scrolling behavior and the relevance of filtered documents. This behavior was observed 10500 times out of 15000 documents. The result shows the tendency that relevant documents are scrolled more often than the others.

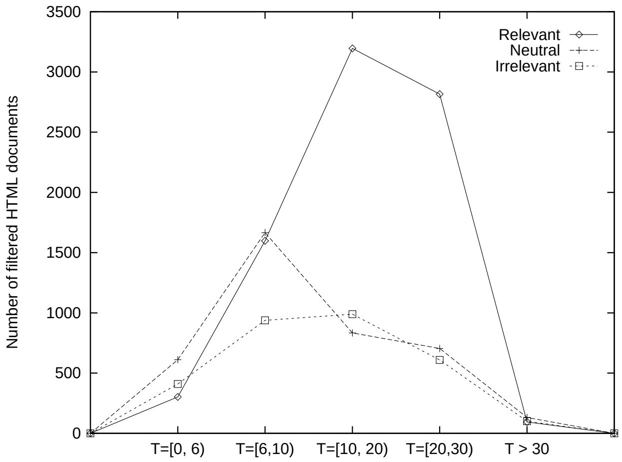

Figure 8: Correlation between the reading time and the relevance of filtered documents. This result indicates that the users spent more time on reading relevant documents than irrelevant ones. However, it also suggests that large reading time (10 or more seconds) was occationally spent on neutral and irrelevant documents.

Feedback Iteration	average feedback±standard deviation		
	WAIR	WH	EG
1	0.44±0.029	0.48±0.033	0.47±0.027
2	0.39±0.034	0.36±0.032	0.22±0.024
3	0.34±0.045	0.29±0.049	0.14±0.03
4	0.35±0.024	0.22±0.051	0.16±0.032
5	0.37±0.038	0.19±0.047	0.7±0.028
6	0.41±0.021	0.21±0.045	0.11±0.034
7	0.42±0.043	0.15±0.029	0.14±0.034
8	0.39±0.018	0.14±0.038	0.11±0.039
9	0.41±0.03	0.13±0.04	0.15±0.029
10	0.42±0.022	0.15±0.029	0.16±0.041
11	0.39±0.028	0.2±0.031	0.13±0.038
12	0.36±0.025	0.26±0.038	0.12±0.024
13	0.39±0.021	0.29±0.012	0.1±0.03
14	0.41±0.026	0.22±0.034	0.1±0.03
15	0.43±0.029	0.15±0.019	0.1±0.03
16	0.45±0.019	0.13±0.029	1±0.03
17	0.43±0.026	0.19±0.02	0
18	0.37±0.032	0.14±0.032	0
19	0.34±0.025	0.13±0.041	0
20	0.3±0.03	0.13±0.032	0
21	0.27±0.039	0.12±0.038	0
22	0.2±0.012	0.14±0.041	0
23	0.24±0.015	0.11±0.041	0
24	0.21±0.021	0.9±0.027	0
25	0.26±0.024	0.1±0.028	0

Table 4: Results for the implicit feedback experiments.

Learning methods	Average	Standard deviation	Number of documents	t-statistics	P(T $\leq$ t)
WAIR	0.359	0.0454	1250	–	–
WH	0.188	0.0419	1250	0.0564	3.46×10^{-20}
EG	0.080	0.0223	1250	0.3069	5.22×10^{-48}

Table 5: Paired-*t* test for the results of implicit feedback.

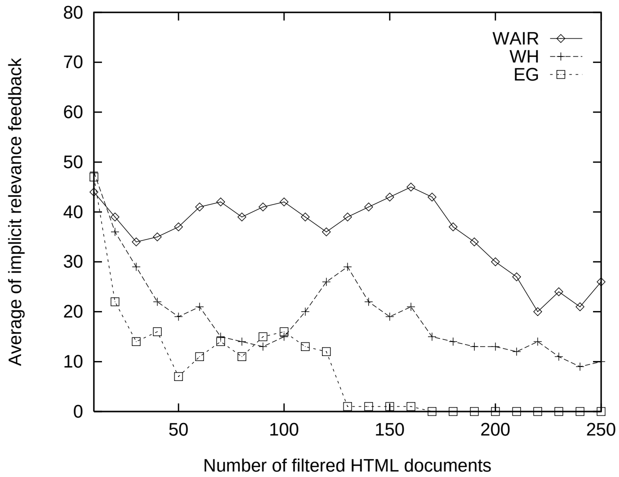

Figure 9: Results for the implicit relevance feedback experiment. Each graph shows the evolution of the average implicit-feedback values as the filtering session goes on. Compared are the three online learning algorithms. Though the overall performance for all the methods is lower than in the explicit feedback experiment, the general tendency looks similar to the previous experiment.

plicit feedback. As the small P-values indicate, the improvement of WAIR compared with WH and EG is statistically significant.

6 Conclusions

In this paper, we formulated the problem of information filtering as a TD(0) reinforcement learning problem, and presented a personalized Web-document filtering system that learns to follow user preferences from observations of his behaviors on the presented documents. A practical method was described that estimates the user's relevance feedback from user behaviors such as reading time, bookmarking, scrolling, and link-following actions.

Our experimental evidence from a field test on a group of users supports that the proposed method effectively adapts to the user's specific interests. This confirms that "learning from shoulders of the user" through self-generated reinforcement signals can significantly improve the performance of information filtering systems. In a series of short-term filtering environments, WAIR achieved superior performance when compared to the conventional feedback methods, including Rocchio, WH, and EG. In terms of adaptation speed, the proposed method converged to the user's specific interest faster than existing relevance feedback methods.

Our work has focused on personalizing information filtering based on existing Web-index services, i.e. AltaVista, Excite, and Lycos. Through the use of learning-based personalization techniques, WAIR could improve the quality of information service of the existing Web search engines. Since every search engine has its strengths and weaknesses, the meta-search approach of WAIR combines the strengths of different search engines while reducing their weaknesses. For the convenience of implementation, we used the conventional search engines directly. Using meta-search engines would further increase the final performance. Similar idea can be used to improve the quality of other Web information service systems.

The online nature of reinforcement learning makes it possible to approximate optimal action policies in ways that put more effort into learning to make good decisions for frequently encountered states, at the expense of less effort for infrequently encountered states. This is the key property that distinguishes reinforcement learning from other relevance feedback methods based on supervised learning. Our experimental results confirms this view: information filtering is dictated by online adaptation based on a small number of documents. The reinforcement learning formulation gave more emphasis on decision making as to filtering the documents rather than just to learn the mappings or profiles. This resulted in better performance than simple supervised learning methods in the dynamic environments. Our work suggests that reinforcement learning can provides a better framework for personalization of information service in the Web environments than conventional supervised learning formulation.

In spite of our success in learning the user preferences in the WAIR system, it should be mentioned 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 was 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 the Korea Ministry of Information and Telecommunications under Grants 00-102 through IITA, by KOSEF through AITRC, and BK21-IT Program.

Bibliography

[Belkin and Croft, 1996] Belkin, N.J. and Croft, W.B. 1992. Information filtering and information retrieval: Two sides of the same coin?, *Communications of the ACM*, 35(12):29-38.

[Boyan et al., 1996] Boyan J., Freitag D., and Joachims T. 1996. A machine learning architecture for optimizing Web search engine, In *Proc. AAAI Workshop on Internet-Based Information Systems*, pp. 324-335.

[Callan, 1998] Callan J. 1998. Learning while filtering documents, In *Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval* (SIGIR-98), pp. 224-231.

[Falk and Josson, 1996] Falk, A. and Josson, I.M. 1996. PAWS: An agent for WWW-retrieval and filtering, In *Proc. Practical Application of Intelligent Agents and Multi-agents Technology* (PAAM-96), pp. 169-179.

[Frakes and Baeza-Yates, 1992] Frakes, W.B. and Baeza-Yates, R. 1992. Stemming algorithms, In *Information Retrieval: Data Structures and Algorithms*, pp. 131-160, Prentice Hall.

[Hirashima et al., 1998] Hirashima, T., Matsuda, N., Nomoto, T., and Toyoda, J. 1998. Context-sensitive filtering for browsing in hypertext, In *Proc. Int. Conf. on Intelligent User Interfaces* (IUI-98), pp. 119-126.

[Joachims et al., 1997] Joachims, T., Freitag D., and Mitchell, T.M. 1997. WebWatcher: A tour guide for the World Wide Web, In *Proc. Int. Joint Conf. on Artificial Intelligence* (IJCAI-97), pp. 770-777.

[Kamba et al., 1997] Kamba, T., Sakagami, H., and Koseki, Y. 1997. ANATAGONOMY: A personalized newspaper on the World Wide Web, *Int. Jor. of Human-Computer Studies*, Vol. 46, pp. 789-803.

[Kindo et al., 1997] Kindo, T., Yoshida, H., Morimoto, T., and Watanabe, T. 1997. Adaptive personal information filtering system that organizes personal profiles automatically, In *the Proc. Int. Joint Conf. on Artificial Intelligence* (IJCAI-97), pp. 716-721.

[Lashkari et al., 1994] Lashkari, Y., Metral, M., and Maes, P., 1994. Collaborative interface agents, In *Proc. of the Twelfth National Conf. on Artificial Intelligence*, pp.444-450.

[Lewis et al., 1996] Lewis, D.D., Schapire, R.E., Callan, J.P., and Papka, R. 1996. Training algorithms for linear text classifiers, In *Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval* (SIGIR-96), pp. 298-306.

[Lieberman, 1995] Lieberman, H. 1995. Letizia: An agent that assists Web browsing, In *Proc. Int. Joint Conf. on Artificial Intelligence* (IJCAI-95), pp. 475-480.

[Maes, 1994] Maes, P. 1994. Agents that reduce work and information overload, *Communications of the ACM*, 37(7):31-40.

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

[Morita and Shinoda, 1994] Morita, M. and Shinoda, Y. 1994. Information filtering based on user behavior analysis and best match text retrieval, In *Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval* (SIGIR-94), pp. 272-281.

[Pazzani and Billsus, 1997] Pazzani, M. and Billsus, D. 1997. Learning and revising user profiles: the identification of interesting Web sites, *Machine Learning*, 27:313-331.

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

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

[Salton and Buckley, 1990] Salton, G. and Buckley, C. 1990. Improving retrieval performance by relevance feedback, *Journal of American Society for Information Science*, 41:288-297.

[Sakagami, 1997] Sakagami, H. and Kamba, T. 1997. Learning personal preferences on online newspaper articles from user behaviors, In *Hyper Proceedings of 6th Int. World Wide Web Conf.*, `http://decweb.ethz.ch/WWW6/Technical/Paper142/Paper142.html`.

[Seo and Zhang, 2000] Seo, Y.-W. and Zhang, B.-T. 2000. A reinforcement learning agent for personalized information filtering, In *Proc. Int. Conf. on Intelligent User Interfaces* (IUI-2000), pp.248-251.

[Sutton and Barto, 1998] Sutton, R.S. and Barto, A.G. 1998. *Reinforcement Learning: An Introduction*, MIT Press.