Adaptive Web Search Based on User's Implicit Preference

Kazunari SUGIYAMA[†], Kenji HATANO[†], Masatoshi YOSHIKAWA[††], and Shunsuke UEMURA[†]

† Nara Institute of Science and Technology 8916–5 Takayama, Ikoma, Nara 630–0192, Japan
†† Nagoya University Furo, Chikusa, Nagoya, Aichi 464–8601, Japan

Abstract Web search engines help users find useful information on the World Wide Web (WWW). However, when the same query is submitted by different users, typical search engines return the same result regardless of who submitted the query. Generally, each user has different information needs for his/her query. Therefore, the search results should be adapted to users with different information needs. In this paper, we first propose several approaches to adapting search results according to each user's need for relevant information without any user effort, and then verify the effectiveness of our proposed approaches. Experimental results show that search systems that adapt to each user's preferences can be achieved by constructing user profiles based on modified collaborative filtering.

Key words WWW, Information retrieval, Information filtering, Relevance feedback, Search process, User modeling

1. Introduction

With the rapid spread of the Internet, anyone can easily access various information by using personal computers, cellular phones, Personal Digital Assistants (PDAs), and such devices. Since information resources on the WWW continue to grow, it has become increasingly difficult for users to find information on the WWW that satisfies their individual needs. Under these circumstances, Web search engines help users find useful information on the WWW. However, when the same query is submitted by different users, most search engines return the same results regardless of who submits the query. In general, each user has different information needs for his/her query. For example, for the query "Java," some users may be interested in documents dealing with the programming language, "Java," while other users may want documents related to "coffee." Therefore, Web search results should adapt to users with different information needs. In order to predict such information needs, there are several approaches applying data mining techniques to extract usage patterns from Web logs [27], [6], [7], [32]. However, the discovery of patterns from Web usage data by itself is not sufficient for performing the personalization tasks. Furthermore, Shahabi et al. [26] have pointed out that the item association generated from Web server logs might be wrong because Web usage data from the server side are not reliable. Therefore, these techniques are not so appropriate for Web personalization. Another novel information systems designed to realize such adaptive systems have been proposed that personalize information or provide more relevant information for users. As far as we know, three types of Web search systems provide such information: (1) systems using relevance feedback [23], (2) systems in which users register their interest or demographic information, and (3) systems that recommend information based on users' ratings. In these systems, users have to register personal information such as their interests, age, and so on beforehand, or users have to provide feedback on relevant or irrelevant judgements, ratings on a scale from 1 (very bad) to 5 (very good), and so on. These types of registration, feedback, or ratings can become time consuming and users prefer easier methods. Therefore, in this paper, we propose several approaches that can be used to adapt search results according to each user's information need. We, then, compare the retrieval accuracy of our proposed approaches. Our approach is novel because it allows each user to perform more fine-grained search by capturing changes of each user's preferences without any user effort. Such a method is not performed in typical search engines.

2. Related Work

As described in Section 1., there are several types of search systems that provide users with information more relevant to their individual needs. For example, we review hyperlink-based personalized Web search, personalized Web sites, and recommender systems.

2.1 Hyperlink-Based Personalized Web Search

The field of Web information retrieval focuses on hyperlink structures of the Web, for example with Web search engines such as Google[(1)][5] and the CLEVER project [12]. To address several problems with these engines, i.e., (1) the importance for a Web page is merely defined, and (2) the relativity of contents among hyperlinked Web pages is not considered, we proposed several approaches to refining the TF-IDF scheme for Web pages using their hyperlinked neighboring pages [28], [29]. In personalized Web searches, the hyperlink structures of the Web are also becoming important. The use of personalized PageRank to enable personalized Web searches was first proposed in [21], where it was suggested as a modification of the global PageRank algorithm, which computes a universal notion of importance. The computation of (personalized) PageRank scores was not addressed beyond the original algorithm. Haveliwala [9] used personalized PageRank scores to enable "topic sensitive" Web searches. Experiments in this work concluded that the use of personalized PageRank scores can improve a Web search. However, no experiments based on a user's context such as browsing patterns, bookmarks, and so on were conducted. Therefore, it is not clear if search results obtained by this approach actually satisfy information needs that is different user by user.

2.2 Personalized Web Sites

Link topology and the structure and contents of Web pages are often used in the construction of a personalized Web site. In this section, we review the framework of these systems with regard to "Link Personalization," and "Content Personalization."

2.2.1 Link Personalization

This scheme involves selecting the links that are more relevant to the user and changing the original navigation space by reducing

[(1)] http://www.google.com/

or improving the relationships between Web pages. E-commerce applications use link personalization to recommend items based on the buying history of clients or some categorization of clients based on ratings and opinions. Users who give similar ratings to similar objects are presumed to have similar preferences, so when a user seeks recommendations about a certain product, the site suggests those recommendations that are most popular for his/her class or those that best correlate with the given product for that class. At the E-commerce site for Amazon.com(2)(the largest bookstore on the Internet), this approach has been taken to an extreme by constructing a "New for you" home page and presenting it to each user, with new products that the user may be interested in. In a recent study, Tsandilas et al. [31] proposed a system that automatically adapts links in the browsed pages based on their relevance to the weighted topics specified by sliders that users can manipulate.

2. 2. 2 Content Personalization

In general, content personalization is done when pages present different information to different users. The difference between this and the link personalization described in Section 2. 2. 1 is subtle because part of the contents (i.e., the link anchors) present different information when links are personalized. However, content personalization is referred to when substantial information in a Web page is personalized, unlike link anchors. For example, My Yahoo!(3)[18] or My Netscape(4)filters the information that is relevant to the user, showing only sections and details in which the user may be interested. The user may explicitly indicate his/her preferences, or preferences may be inferred (semi-) automatically from his/her profile or from his/her navigation activity. At these sites, users choose a set of "modules" from a large set including weather, news, music and so on, and further personalize these modules by choosing a set of attributes of the module to be perceived. The approach followed in these applications is that the users should be able to "construct" their own pages and even the layout may be customized. However, user preferences or demographic information are acquired based on the prior questionnaire. Therefore, these sites have two problems: (1) the users' loads become high because these systems heavily rely on the users' inputs; and (2) these sites cannot adapt to the changes in users' preferences unless the users change their previously registered preferences by themselves.

2. 3 Recommender Systems

It has become increasingly difficult to search for useful information on the Web because the amount of information on the Web continues to grow. Therefore, we get the feeling of being overwhelmed by the number of choices. This situation is often referred to as "information overload." As one of the most promising approaches to alleviate this overload, recommender systems have emerged in domains such as E-commerce, digital libraries, and knowledge management. These systems provide personalized suggestions based on user preferences. Recommender systems collect user feedback in the form of ratings for items in a given domain and exploit similarities and differences among profiles of several users in determining how to recommend an item. There are two prevalent approaches to constructing recommender systems – collaborative filtering-based and content-based recommendation.

2. 3. 1 Collaborative Filtering-Based Recommendation

Collaborative filtering-based recommendation is the most successful recommendation technique to date. The term collaborative filtering was coined by Goldberg et al. [8]. Collaborative filtering means that people collaborate to help one another perform filtering by recording their reactions to documents they read. Based on this concept, Goldberg et al. developed a system called Tapestry which filter email and it allows users to annotate messages. The collaborative filtering provided by Tapestry was not automated, and users were required to formulate complex queries in a special query language designed for the task. The GroupLens research system [22], [15], which filters Usenet news, first introduced a rating-based automated collaborative filtering system using the k-nearest neighbor-based algorithm. In this algorithm, a subset of appropriate k users is chosen based on their similarity to the active user, and a weighted aggregate of their rating is used to generate predictions for the active user. GroupLens then recommends Usenet news articles to these active users.

While the Tapestry and GroupLens rely on explicit ratings, some systems rely on implicit ratings. For example, Morita et al. [20] exploit "time-spent-reading" as a measure of implicit ratings. PHOAKS (People Helping One Another Know Stuff) [30] also uses implicit ratings to construct a recommender system by examining Usenet news postings to find "endorsements" of Web sites. It then creates a listing of the top Web sites endorsed in each newsgroup. Some recommender systems also explore user preferences transparently without any extra user effort like the recommender systems relying on implicit ratings described above. For example, Letizia [16], [17] and WebWatcher [14] infer user preferences by observing user-browsing behavior. However, the main shortcomings in Letizia and WebWatcher are that they maintain persistent and slowly-changing user models and overlook the fact that different browsing sessions by the same user or even a single session may involve different user interests and goals.

Furthermore, at the E-commerce sites such as Amazon.com, CDnow.com (the largest CD store on the Web) and MovieFinder.com (one of the most visited movie sites), automated collaborative filtering systems have been used with considerable success.

2. 3. 2 Content-Based Recommendation

A content-based approach provides recommendations by comparing representations of content contained in an item with representations of content that the user is interested in. In this approach, a model of user ratings is first developed. Algorithms in this category use probabilities and envision the collaborative filtering process by computing the expected value of a user prediction given the user's ratings on other items. The model building process is performed by three different machine learning algorithms: (1) Bayesian network, (2) clustering, and (3) rule-based models. The Bayesian network model [4] constructs a probabilistic model for a collaborative filtering problem. Clustering model addresses collaborative filtering as a classification problem [3], [4]. It works by clustering similar users in the same class and estimating the probability that a particular user is in a particular class; from there it computes the conditional probability of ratings. The rule-based model applies association rule discovery algorithms to find associations between co-purchased items. It then generates item recommendations based on the strength of the association between items [24].

The systems described in Section 2. 3. 1 only provide recommendations based on collaborative filtering. However, some systems provide better recommendations by combining collaborative filtering with content information. Fab [2] uses relevance feedback to simultaneously construct a personal filter along with a communal "topic" filter. Web pages are initially ranked by the topic filter and then sent to user's personal filters. The user then provides relevance feedback for that Web page, and this feedback is used to modify both the personal filter and the originating topic filter. Basu et al. [3] integrate content and collaboration in a framework where they treat recommendation as a classification task. Melville et al. [19] overcome drawbacks of collaborative filtering systems in their recommender system by exploiting content information of items already rated. In recent study on recommender systems, Schafer et al. [25]

(2) http://www.amazon.com/

(3) http://www.my.yahoo.com/

(4) http://my.netscape.com/

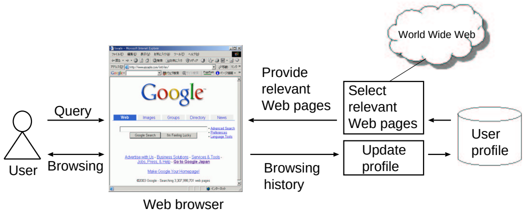


Figure 1 System overview.

introduce a new class of recommender system that provides users with personalized control over the generation of a single recommendation list formed from a combination of rich data using multiple information resources and recommendation techniques.

3. Our Proposed Method

As we described in Section 2. 1, hyperlink-based personalized search systems have a problem in that they do not clarify whether their search results actually satisfy each user's information need. This is because personalization based on user's context, i.e., browsing patterns, bookmarks, and so on is not performed. The personalized Web sites described in Section 2. 2 have the following shortcomings: (1) users have to rate items or adjust sliders to obtain relevant information in "Link Personalization" described in Section 2. 2. 1; and (2) in "Content Personalization" described in Section 2. 2. 2, the load on users becomes high because they have to answer questionnaires in advance to register their personal preferences or demographic information, and they have to change their registered information by themselves if their interests change. In addition, the recommender systems described in Section 2. 3, have the potential to provide serendipitous recommendations if users are only willing to rate items. However, in actuality, most users are unwilling to rate items even though user's ratings for items are key factors to achieving better recommendations. As a result, the accuracy of recommendations may be poor.

We do not necessarily believe that approaches based on user ratings provide users with more relevant information that satisfies each user's information need. Therefore, search system should directly and exactly capture the changes in each user's preferences without any user effort in order to provide more relevant information for each user. In order to construct such a system, we propose several approaches to adapting search results according to each user's information need. Unlike the research studies described in the previous section, our approach is novel because it allows each user to perform a fine-grained search by capturing the changes in each user's preferences without any user effort.

Figure 1 shows an overview of our system. When a user submits a query to a search engine through a Web browser, the search engine returns search results corresponding to the query. Based on the search results, the user may select a Web page in an attempt to satisfy his/her information need. In addition, the user may access more Web pages by following the hyperlinks on his/her selected Web page and continue to browse. Our system monitors the user's browsing history and updates his/her profile whenever his/her browsing page changes. When the user submits a query the next time, the search results adapt based on his/her user profile.

In the following sections, we explain how to construct a user profile in the update profile component illustrated in Figure 1. In our approach, the user profile is constructed implicitly. In other words, a user does not need to perform explicit efforts such as feedback, ratings and so on in order to construct his/her profile. We construct each user profile based on the following two methods: (1) Pure browsing history, and (2) Modified collaborative filtering.

3. 1 User Profile Construction Based on Pure Browsing History

In this method, we assume that the preferences of each user consist of the following two aspects: (1) persistent (or long term) preferences, and (2) ephemeral (or short term) preferences. In persistent preferences, the user profile is incrementally developed over time and it is stored for use in later sessions. The information exploited for constructing the profile usually comes from various sources, so it relies on different aspects of the user. On the other hand, in ephemeral preferences, the information used to construct each user profile is only gathered during the current session, and it is immediately exploited for executing some adaptive process aimed at personalizing the current interaction. Of course, users might do different task in one day. They may well do several searches and browsing activities in that time period. Therefore, it is one of the ideas that searches and browsing activities are dealt with separately without falling into one logical session. However, as described above, we assume that user's preferences are constructed by accumulating his/her past preferences. Therefore, we construct each user profile $\boldsymbol{P}$ considering both persistent preferences, $\boldsymbol{P}^{per}$, and ephemeral preferences, $\boldsymbol{P}^{today}$. $\boldsymbol{P}^{per}$ shows a user profile constructed exploiting the user's browsing history of Web page from N days ago (see Figure 2). Here, we introduce the concept of window size in order to construct $\boldsymbol{P}^{per}$, and define $S_j (j = 0, 1, 2, \cdots, N)$ as the number of Web pages the user browsed on the j^{th} day. "$j = 0$" means "today" as shown in Figure 2. In each day, $\boldsymbol{P}^{today}$ is constructed through the following process. First, we denote the feature vector $\boldsymbol{w}^{hp}$ of browsed Web page hp $(hp = 1, 2, \cdots, S_0)$ as follows:

$$\boldsymbol{w}^{hp} = (w^{hp}_{t_1}, w^{hp}_{t_2}, \cdots, w^{hp}_{t_m}),$$

where m is the number of unique terms in the Web page hp, and t_k $(k = 1, 2, \cdots, m)$ denotes each term. Using the TF (term frequency) scheme, we also define each element $w^{hp}_{t_k}$ of $\boldsymbol{w}^{hp}$ as follows:

$$w^{hp}_{t_k} = c^{hp} \cdot \frac{tf(t_k, hp)}{\sum_{s=1}^{m} tf(t_s, hp)}, \tag{1}$$

where $tf(t_k, hp)$ is the frequency of term t_k in each browsed Web page hp, and c^{hp} is a constant that shows to what extent our system reflects the contents of the Web page for each user profile. We define constant c^{hp} as follows:

$$c^{hp} = \begin{cases} 1; & dr \geqq Th \\ 0; & dr < Th \end{cases}$$

where dr denotes the time spent reading normalized by the number of terms in Web page hp. We define threshold Th as 0.317 based on our preliminary experiments. We then denote user profile $\boldsymbol{P}^{today}$ as follows:

$$\boldsymbol{P}^{today} = (p^{today}_{t_1}, p^{today}_{t_2}, \cdots, p^{today}_{t_m}),$$

and define each element $p^{today}_{t_k}$ as follows:

$$p^{today}_{t_k} = \frac{1}{S_0} \sum_{hp=1}^{S_0} w^{hp}_{t_k},$$

As described above, $\boldsymbol{P}^{today}$ shows a user profile constructed using the user's browsing history of today's Web page. Moreover, we set window size N $(N = 1, 2, \cdots, 30)$ to construct $\boldsymbol{P}^{per}$. We also denote $\boldsymbol{P}^{per}$ as follows:

$$\boldsymbol{P}^{per} = (p^{per}_{t_1}, p^{per}_{t_2}, \cdots, p^{per}_{t_m}),$$

and define each element $p^{per}_{t_k}$ as follows:

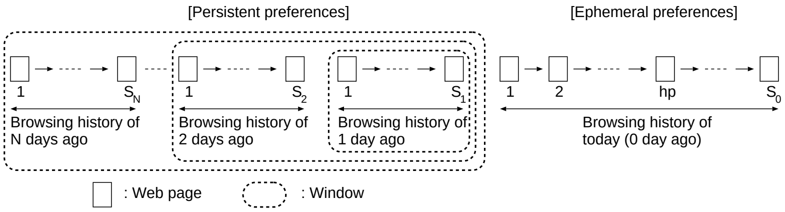
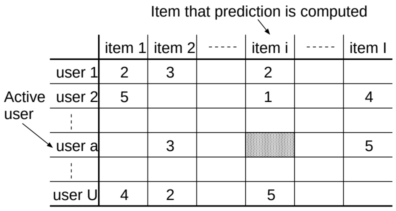

[Persistent preferences]
[Ephemeral preferences]
1
S_N
1
S_2
1
S_1
1
2
hp
S_0
Browsing history of N days ago
Browsing history of 2 days ago
Browsing history of 1 day ago
Browsing history of today (0 day ago)
: Web page
: Window

Figure 2 Window size for constructing persistent user profile.


Item that prediction is computed
Active user

	item 1	item 2	-----	item i	-----	item I
user 1	2	3		2		
user 2	5			1		4
⋮						
user a		3		▨		5
⋮						
user U	4	2		5		

Figure 3 User-item ratings matrix for collaborative filtering.

$$p_{t_k}^{per} = \frac{1}{S_N} \sum_{hp=1}^{S_N} w_{t_k}^{hp} \cdot e^{-\frac{\log 2}{hl}(d - d_{t_{k\_init}})}, \qquad (2)$$

where $e^{-\frac{\log 2}{hl}(d - d_{t_{k\_init}})}$ is a forgetting factor under the assumption that user's preferences gradually decay as days pass. In this factor, $d_{t_{k\_init}}$ is the day when term t_k initially occurs, d is the number of days following to $d_{t_{k\_init}}$, and hl is a half life span parameter. We set the half-life span hl to 7. In other words, the intuition behind this assumption is that user's preferences reduce by 1/2 in one week. Let us assume that each user browsed S_N pages on each day. Of course, this value of S_N, the number of browsed Web pages, differs user by user. Therefore, we normalize $p_{t_k}^{per}$ using S_N as shown in Equation (2). Using these parameters, we finally construct the user profile $\boldsymbol{P}$ as defined in the following equation:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + b\boldsymbol{P}^{today}, \qquad (3)$$

where a and b are constants that satisfy $a + b = 1$.

3. 2 User Profile Construction Based on Modified Collaborative Filtering Algorithm

In this section, we first briefly review the pure collaborative filtering algorithm, especially neighborhood-based algorithms, and then describe how to construct user profiles using the modified collaborative filtering algorithms.

3. 2. 1 Overview of the Pure Collaborative Filtering Algorithm

Collaborative filtering can be represented as the problem of predicting missing values in a user-item ratings matrix. Figure 3 shows a simplified example of a user-item ratings matrix.

In the neighborhood-based algorithm [11], a subset of users is first chosen based on their similarity to the active user, and a weighted combination of their rating is then used to produce predictions for the active user. The algorithm we use can be summarized in the following steps:

(i) Weight all users with respect to similarity to the active user. This similarity between users is measured as the Pearson correlation coefficient between their rating vectors.

(ii) Select n users that have the highest similarity with the active user. These users form the *neighborhood*.

(iii) Compute a prediction from a weighted combination of the neighbor's ratings.

In step (i), $S_{a,u}$, which denotes similarity between users a and u, is computed using the Pearson correlation coefficient defined below:

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

where $r_{a,i}$ is the rating given to item i by user a, and $\bar{r}_a$ is the mean rating given by user a, and I is the total number of items.

In step (ii), i.e., neighborhood-based methods, a subset of appropriate users is chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user in the next step (iii).

In step (iii), predictions are computed as the weighted average of deviations from the neighbor's mean:

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

where $p_{a,i}$ is the prediction for active user a for item i. $S_{a,u}$ is the similarity between users a and u as described in Equation (4), and n is the number of users in the neighborhood.

3. 2. 2 User Profile Construction using Modified Collaborative Filtering Algorithm

In the pure collaborative filtering algorithms described in Section 3. 2. 1, we considered a user-item ratings matrix. Similarly, in the construction of a user profile, we can consider a user-term weights matrix like that shown in Figure 4(a). In addition, based on the pure collaborative filtering algorithms described in Section 3. 2. 1, we can apply their predictive algorithms to predict a term weight in each user profile. In other words, since each user profile is computed based on term weights in a Web page the user browsed and the browsed pages are different according to each user, the profile is constructed in the form of a user-term weights matrix with missing values, as illustrated in Figure 4. This is very analogous to the user-item ratings matrix used in the pure collaborative filtering algorithms. Therefore, we expect that a more accurate user profile considering similar user's preference is constructed since these missing values can be predicted using the algorithms in collaborative filtering. In this approach, we propose the following two methods: (1) user profile construction based on the static number of users in the neighborhood, and (2) user profile construction based on dynamic number of users in the neighborhood.

(1) User Profile Construction Based on the Static Number of Users in the Neighborhood

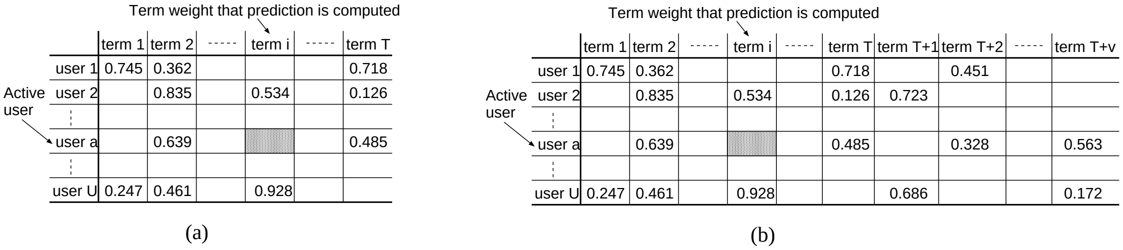
In this method, our proposed algorithms are explained in the following steps (note the similarity to the collaborative filtering algorithms described in Section 3. 2. 1):

(i) Weight all users with respect to similarity to the active user. This similarity between users is measured as the Pearson correlation coefficient between their term weight vectors unlike the rating vectors described in Section 3. 2. 1.

(ii) Select n users that have the highest similarity to the active user. These users form the neighborhood.

(iii) Compute a prediction from a weighted combination of the neighbor's term weights

In step (i), $S_{a,u}$, which denotes similarity between users a and u,


Term weight that prediction is computed
Active user
Term weight that prediction is computed
Active user

	term 1	term 2	-----	term i	-----	term T
user 1	0.745	0.362				0.718
user 2		0.835		0.534		0.126
⋮						
user a		0.639		▨		0.485
⋮						
user U	0.247	0.461		0.928		

(a)

	term 1	term 2	-----	term i	-----	term T	term T+1	term T+2	-----	term T+v
user 1	0.745	0.362				0.718		0.451		
user 2		0.835		0.534		0.126	0.723			
⋮										
user a		0.639		▨		0.485		0.328		0.563
⋮										
user U	0.247	0.461		0.928			0.686			0.172

(b)

Figure 4 User-term weights matrix for modified collaborative filtering [(a) when each user browsed k Web pages, (b) when each user browsed $k+1$ Web pages].

is computed using the Pearson correlation coefficient, defined below:

$$S_{a,u} = \frac{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a) \times (w_{u,i} - \bar{w}_u)}{\sqrt{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a)^2 \times \sum_{i=1}^{T}(w_{u,i} - \bar{w}_u)^2}},$$

where $w_{a,i}$ is the weight of term i regarding user a computed based on term frequency in a browsed Web page defined by Equation (1), and $\bar{w}_a$ is the mean term weight regarding user a, and T is the total number of terms.

In step (ii), i.e., neighborhood-based methods, a subset of appropriate users is chosen based on their similarity to the active user, and a weighted aggregate of their term weights is used to generate predictions for the active user in the coming step (iii). In this step, the number of selected users is fixed to n for any user. That is why we call this method "static."

In step (iii), predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{w}_a + \frac{\sum_{u=1}^{n}(w_{u,i} - \bar{w}_u) \times S_{a,u}}{\sum_{u=1}^{n} S_{a,u}},$$

where $p_{a,i}$ is the prediction for the active user a for weight of term i, $S_{a,u}$ is the similarity between users a and u, as described in Equation (4), and n is the number of users in the neighborhood.

(2) User Profile Construction Based on Dynamic Number of Users in the Neighborhood

In this method, our proposed algorithms are explained in the following steps (note the similarity to the collaborative filtering algorithms described in Section 3. 2. 1, and the aforementioned static approach):

(i) Generate clusters of users by means of the k-Nearest Neighbor algorithms [13]. The similarity between user a and these clusters is measured as the Pearson correlation coefficient between their term weight vectors.

(ii) Select n clusters that have higher similarity to the active user than the threshold. We consider the centroid vectors of these selected clusters as the neighborhood of the active user.

(iii) Compute a prediction from a weighted combination of the term weights using centroid vectors of clusters.

In step (i), $S_{a,g}$, which denotes similarity between users a and centroid vectors of clusters g, is computed using the Pearson correlation coefficient, defined below:

$$S_{a,g} = \frac{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a) \times (w_{g,i} - \bar{w}_g)}{\sqrt{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a)^2 \times \sum_{i=1}^{T}(w_{g,i} - \bar{w}_g)^2}},$$

where $w_{a,i}$ is the weight of term i regarding user a computed based on term frequency in a browsed Web page defined by Equation (1), and $\bar{w}_a$ is the mean term weight regarding user a, and T is the total number of terms.

In step (ii), several clusters are chosen based on their similarity to the active user, and a weighted aggregate of their term weights is used to generate predictions for the active user in the next step (iii). In this step, the number of selected clusters is different user by user. That is why we call this method "dynamic." Therefore, it is expected that this method allows each user to perform more fine-grained search.

In step (iii), predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{w}_a + \frac{\sum_{g=1}^{n}(w_{g,i} - \bar{w}_g) \times S_{a,g}}{\sum_{g=1}^{n} S_{a,g}},$$

where $p_{a,i}$ is the prediction for the active user a for term weights i, $S_{a,g}$ is the similarity between users a and centroid vectors of clusters g, and n is the number of centroid vectors of clusters in the neighborhood.

4. Experiments

4.1 Experimental Setup

We conducted experiments in order to verify the effectiveness of the three approaches: (1) relevance feedback and implicit approaches, (2) user profiles based on pure browsing history as described in Section 3. 1, and (3) user profiles based on the modified collaborative filtering algorithm described in Section 3. 2. While users have to provide feedback explicitly in relevance feedback, users do not have to provide any effort in our proposed methods (2) and (3) since our system implicitly captures changes in user's preference. The experiments were implemented using Perl on a workstation (CPU: UltraSparc-II 480MHz× 4, Memory: 2GBytes, OS: Solaris8). We used 50 query topics that were employed as test topics in the TREC WT10g test collection [10]. Note that we only used query topics of the test collection, and did not use the contents of the test collection. In our experiments, we observed the browsing history of 20 subjects for 30 days. The subjects browsed 12 Web pages in one day on average. In addition, the number of terms in user profiles accumulated during the 30 days is about 810,000. In the following, let the h^{th} Web page in the search results and the user profile as defined by Equation (3) be rp_h and $\boldsymbol{P}$, respectively, Then, the feature vector of the h^{th} Web page rp_h in the search results, $\boldsymbol{w}^{rp_h}$, is defined as follows:

$$\boldsymbol{w}^{rp_h} = (w_{t_1}^{rp_h}, w_{t_2}^{rp_h}, \cdots, w_{t_m}^{rp_h}),$$

where m is the number of unique terms in the Web page rp_h, and t_k $(k = 1, 2, \cdots, m)$ denotes each term. We also define each element $w_{t_k}^{rp_h}$ of $\boldsymbol{w}^{rp_h}$ based on the TF (term frequency) scheme as follows:

$$w_{t_k}^{rp_h} = \frac{tf(t_k, rp_h)}{\sum_{s=1}^{m} tf(t_s, rp_h)},$$

where $tf(t_k, rp_h)$ is the frequency of term t_k in the rp_h. The similarity $sim(\boldsymbol{P}, \boldsymbol{w}^{rp_h})$ between the user profile $\boldsymbol{P}$ and the feature vector of the h^{th} Web page in search results $\boldsymbol{w}^{rp_h}$ is computed by

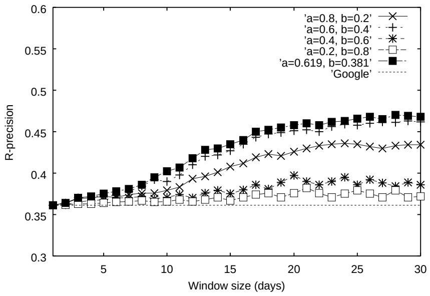


Figure 5 R-precision obtained by relevance feedback-based user profile ($FB = 1$).

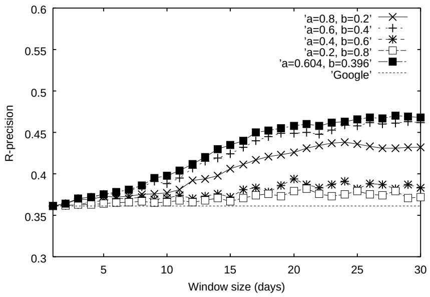


Figure 6 R-precision obtained by relevance feedback-based user profile ($FB = 2$).

the following Equation.

$$sim(\boldsymbol{P}, \boldsymbol{w}^{rp_h}) = \frac{\boldsymbol{P} \cdot \boldsymbol{w}^{rp_h}}{|\boldsymbol{P}| \cdot |\boldsymbol{w}^{rp_h}|}. \quad (4)$$

Based on the value obtained in Equation (4), the search results are adapted to each user according to his/her profile. These results are compared with the search results of Google [5]. We then evaluate the retrieval accuracy using R-precision [1]. We employed 30 as the value of R because users tend to take a look at the first 30 documents retrieved.

4. 2 Experimental Results

4. 2. 1 User Profile Based on Relevance Feedback

Relevance feedback [23] is the most popular query reformulation strategy. In a relevance feedback process, the user is presented with a list of the retrieved documents and marks those that are relevant after examining them. The basic idea is to reformulate the original query vector $\boldsymbol{Q}^{org}$ into new query vector $\boldsymbol{Q}^{new}$ such that it gets closer to the term-weight vector space of the relevant documents. In our experiment, we use the Rocchio formulation defined as follows:

$$\boldsymbol{Q}^{new} = \alpha \boldsymbol{Q}^{org} + \frac{\beta}{|D_r|} \sum_{\boldsymbol{d}_j \in D_r} \boldsymbol{d}_j - \frac{\gamma}{|D_n|} \sum_{\boldsymbol{d}_j \in D_n} \boldsymbol{d}_j,$$

where D_r and D_n are the set of relevant and non-relevent documents as identified by the user among the retrieved documents, respectively, and $|D_r|$ and $|D_n|$ are the number of documents in the sets D_r and D_n, respectively. We set α, β and γ that are tuning constants to 1, 1 and 0, respectively.

We believe that the new query vector $\boldsymbol{Q}^{new}$ obtained by the user's judgement, whether the retrieved documents are relevant or not, reflects the user's preferences. Therefore, we treat $\boldsymbol{Q}^{new}$ as $\boldsymbol{P}^{today}$ defined by Equation (3), and employ $\boldsymbol{Q}^{new}$ as an initial preference of a user to construct a user profile. In this case, using Equation (3), the user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + b\boldsymbol{Q}^{new}, \quad (5)$$

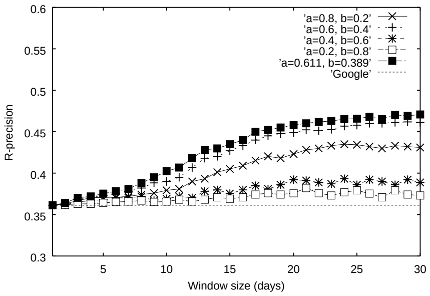


Figure 7 R-precision obtained by relevance feedback-based user profile ($FB = 3$).

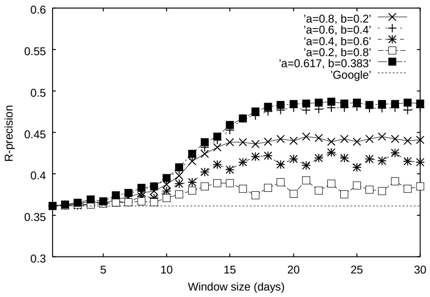


Figure 8 R-precision obtained by pure browsing history-based user profile.

We asked each subject to judge if the search results returned by the search engine are relevant, and constructed the user profile $\boldsymbol{P}$ based on Equation (5). In this experiment, we varied the number of feedbacks FB that each subject provided from 1 to 3. Figures 5 to 7 show the R-precision when the values of a and b are varied such that these values satisfy $a + b = 1$ under the condition that the numbers of feedbacks are 1, 2, and 3.

4. 2. 2 User Profile Based on Pure Browsing History

In this approach, each user profile is constructed as mentioned in Section 3. 1. The user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + b\boldsymbol{P}^{today}.$$

Figure 8 shows the R-precision when the values of a and b are varied such that these values satisfy $a + b = 1$.

4. 2. 3 User Profile Based on Modified Collaborative Filtering

In this approach, when the user browses a new Web page, new terms are added to his/her user profile. However, other users do not always browse the same pages, so missing values occur in the user-term weights matrix as illustrated in Figure 4. These missing values are predicted using the algorithms described in Section 3. 2. 2, and then the matrix is filled. We consider that this user-term vector reflects the user's preferences. Let this user-term vector with predicted value be $\boldsymbol{V}^{pre}$. We treat $\boldsymbol{V}^{pre}$ as $\boldsymbol{P}^{today}$ defined by Equation (3), and employ $\boldsymbol{V}^{pre}$ as an initial preference of a user to construct a user profile. In this case, using Equation (3), the user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + b\boldsymbol{V}^{pre}, \quad (6)$$

Figures 9 to 12 show the R-precision of static approaches when the values of a and b are varied such that these values satisfy $a + b = 1$ under the condition that the numbers of neighbors n are 5, 10, 15, and 20, respectively. In addition, Figure 13 shows the R-precision of dynamic approaches.

4. 3 Discussion

In this section, we discuss the results obtained using each approach discussed in Section 4. 2. Note that, in Figures 5 to 13, the

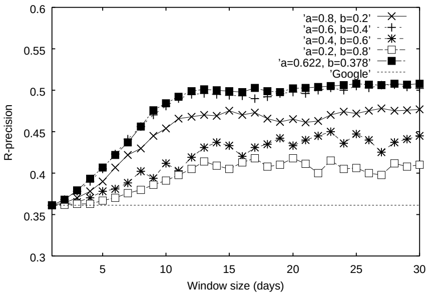


Figure 9 R-precision obtained by modified collaborative filtering-based user profile (static, $n = 5$).

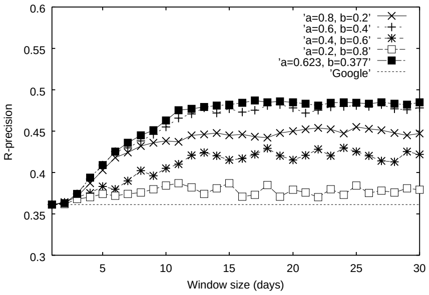


Figure 10 R-precision obtained by modified collaborative filtering-based user profile (static, $n = 10$).

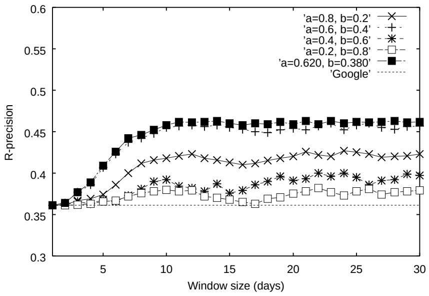


Figure 11 R-precision obtained by modified collaborative filtering-based user profile (static, $n = 15$).

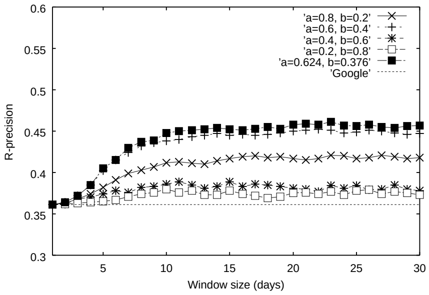


Figure 12 R-precision obtained by modified collaborative filtering-based user profile (static, $n = 20$).

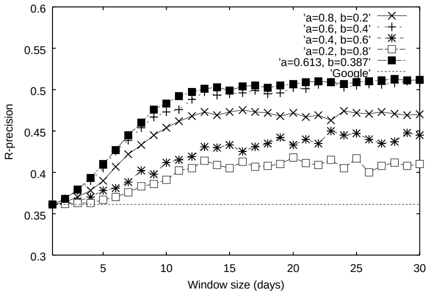


Figure 13 R-precision obtained by modified collaborative filtering-based user profile (dynamic).

R-precision of Google is constant because it does not depend on the window size.

In the relevance feedback-based user profile shown in Figures 5 to 7, we found that a user profile that provides search results adaptive to a user can be constructed when a window size with about 20 days is used regardless of the number of feedbacks. As mentioned in Section 4. 2. 1, we used query vector reformulated by relevance feedback as an initial preference of a user. However, we could not observed significant improvement in precision even if the number of feedbacks increases. We consider that this effect is caused because the initial preference of a user is absorbed by persistent preferences constructed using the window size. In addition, it is valid that we conducted experiments by examining the number of feedbacks from 1 to 3 since the precision is not improved largely in this range.

In the user profile based on pure browsing history shown in Figure 8, we found that a user profile that provides search results adaptive to a user can be constructed when a window size with about 15 days is used. This approach can achieve higher precision than the relevance feedback-based user profile, and the result shows that the user's browsing history strongly reflects the user's preference.

In addition, in the user profile based on modified collaborative filtering shown in Figures 9 to 13, we found that a user profile that provides search results adaptive to a user can be constructed when a window size with about 10 days is utilized. In user profile construction based on the static number of users in the neighborhood described in Section 3. 2. 2(1), the best precision is obtained in the case of $n = 5$ as illustrated in Figure 9; in other words, the 5 nearest neighbors of each user are taken. Therefore, as shown in Figures 9 to 12, we found that it is not so effective to adapt search results to each user even if more nearest neighbors are used. In addition, the user preferences of not only a certain user but also other users are exploited in this approach. We consider that this method obtained higher precision than the aforementioned approaches. In user profile construction based on the dynamic number of users in the neighborhood described in Section 3. 2. 2(2), we could obtain the best precision in all of our experimental results in the case of $a = 0.613$ and $b = 0.387$ in Equation (6) as shown in Figure 13. In this method, the neighborhood of each user is determined by the centroid vectors of clusters of users, and the number of the clusters is different user by user. Therefore, we believe that this method allows each user to perform more fine-grained search compared with the static method.

The best precision of any of the methods is obtained when a is larger than 0.6 and b is less than 0.4. This shows that search results that adapt to each user can be provided by weighting persistent preferences a little larger than ephemeral preferences. Moreover, the smaller the value of a and the larger the value of b, the larger the fluctuation in precision. Therefore, it is difficult to provide users with search results that adapt to each user when the user profile is constructed by weighting ephemeral preferences larger than persistent preferences. Furthermore, the precision obtained by each of our proposed methods can outperform the precision obtained by Google as shown in Figures 5 to 13. We believe that our proposed methods can perform a fine-grained search for each user that typical search engines can not perform.

5. Conclusion

In this paper, in order to provide each user with more relevant information, we proposed several approaches to adapting search results according to each user's information need. Our approach is novel in that it allows each user to perform a fine-grained search, which is not performed in typical search engines, by capturing changes in each user's preferences. We conducted experiments in order to verify the effectiveness of the approaches: (1) relevance feedback and implicit approaches, (2) user profiles based on pure browsing history, and (3) user profiles based on the modified collaborative filtering. We then evaluated the retrieval accuracy of these approaches. The user profile constructed based on modified collaborative filtering achieved the best accuracy. This approach allows us to construct a more appropriate user profile and perform a fine-grained search that is better adapted to each user's preferences. In the future, if broadband networks spread widely, information is expected to be provided in a variety of forms such as music, movies and so on. In addition, more information will be provided for mobile terminals such as cellular phones, PDAs, or terminals in cars for Intelligent Transportation Systems (ITS). We believe that the technique proposed in this paper can be applied to situations where users require more relevant information to satisfy their information needs. In future work, we plan to analyze each user's browsing behavior in one day in more detail and conduct experiments aiming at achieving adaptive Web search with higher precision for each user.

Acknowledgements

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan, under grants #14780325 and #15017243.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[2] M. Balabanovic and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, Vol. 40, No. 3, pp. 66–72, 1997.

[3] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98)*, pp. 714–720, 1998.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of the 14th Conference on Uncertanity in Artificial Intelligence (UAI '98)*, pp. 43–52, 1998.

[5] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. of the 7th International World Wide Web Conference (WWW7)*, pp. 107–117, 1998.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, Vol. 1, No. 1, pp. 5–32, 1999.

[7] X. Fu, J. Budzik, and K. J. Hammond. Mining Navigation History for Recommendation. In *Proc. of the 5th International Conference on Intelligent User Interfaces (IUI 2000)*, pp. 106–112, 2000.

[8] D. Goldberg, D. Nichols, B. M. Oki, and D. B. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, Vol. 35, No. 12, pp. 61–70, 1992.

[9] T. H. Haveliwala. Topic-Sensitive PageRank. In *Proc. of the 11th International World Wide Web Conference (WWW2002)*, pp. 517–526, 2002.

[10] D. Hawking. Overview of the TREC-9 Web Track. *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*, pp. 87–102, 2001.

[11] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 230–237, 1999.

[12] IBM Almaden Research Center. Clever Searching. http://www.almaden.ibm.com/cs/k53/clever.html.

[13] R. A. Jarvis and E. A. Patrick. Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Transactions on Computers*, Vol. C22, No. 11, pp. 1025–1034, 1973.

[14] T. Joachims, D. Freitag, and T. M. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, pp. 770–777, 1997.

[15] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, Vol. 40, No. 3, pp. 77–87, 1997.

[16] H. Lieberman. Letizia: An Agent That Assists Web Browsing. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pp. 924–929, 1995.

[17] H. Lieberman. Autonomous Interface Agents. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '97)*, pp. 67–74, 1997.

[18] U. Manber, A. Patel, and J. Robison. Experience with Personalization on Yahoo! *Communications of the ACM*, Vol. 43, No. 8, pp. 35–39, 2000.

[19] P. Melville, R. J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proc. of the 18th National Conference on Artificial Intelligence (AAAI2002)*, pp. 187–192, 2002.

[20] M. Morita and Y. Shinoda. Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pp. 272–281, 1994.

[21] L. Page. The PageRank Citation Ranking: Bringing Order to the Web. http://google.stanford.edu/~backrub/pageranksub.ps, 1998.

[22] P. Resnick, N. Iacovou, M. Suchak, and J. Riedl P. Bergstorm. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94)*, pp. 175–186, 1994.

[23] J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[24] B. M. Sarwar, G. Karypis, and J. A. Konstan. Analysis of Recommendation Algorithms for E-commerce. In *Proc. of the 2nd ACM Conference on Electronic Commerce (EC '00)*, pp. 158–167, 2000.

[25] J. B. Schafer, J. A. Konstan, and J. Riedl. Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations. In *Proc. of the 11th International Conference on Information and Knowledge Management (CIKM '02)*, pp. 43–51, 2002.

[26] C. Shahabi and Y.-S. Chen. An Adaptive Recommendation System without Explicit Acquisition of User Relevance Feedback. *Distributed and Parallel Databases*, Vol. 14, No. 3, pp. 173–192, 2003.

[27] M. Spiliopoulou and L. Faulstich. WUM–A Tool for WWW Utilization Analysis. In *Proc. of the International Workshop on the World Wide Web and Databases (WebDB'98)*, pp. 184–203, 1998.

[28] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. A Method of Improving Feature Vector for Web Pages Reflecting the Contents of their Out-Linked Pages. In *Proc. of the 13th International Conference on Database and Expert Systems Applications (DEXA2002)*, pp. 891–901, 2002.

[29] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. Refinement of TF-IDF Schemes for Web Pages Using their Hyperlinked Neighboring Pages. In *Proc. of the 14th ACM Conference on Hypertext and Hypermedia (HT '03)*, pp. 198–207, 2003.

[30] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. PHOAKS: A System for Sharing Recommendations. *Communications of the ACM*, Vol. 40, No. 3, pp. 59–62, 1997.

[31] T. Tsandilas and M. C. Schraefel. User-Controlled Link Adaptation. In *Proc. of the 14th ACM Conference on Hypertext and Hypermedia (HT '03)*, pp. 152–160, 2003.

[32] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin. Ranking User's Relevance to a Topic through Link Analysis on Web Logs. In *Proc. of the 4th ACM CIKM International Workshop on Web Information and Data Management (WIDM'02)*, pp. 49–54, 2002.