User-Oriented Adaptive Web Information Retrieval Based on Implicit Observations

Kazunari Sugiyama[1], Kenji Hatano[1],
Masatoshi Yoshikawa[2], and Shunsuke Uemura[1]

[1] Graduate School of Information Science, Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{kazuna-s, hatano, uemura}@is.aist-nara.ac.jp
[2] Information Technology Center, Nagoya University,
Furo, Chikusa, Nagoya, Aichi 464-8601, Japan
yosikawa@itc.nagoya-u.ac.jp

Abstract. Web search engines help users find useful information on the 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 result 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. Experimental results show that search systems that adapt to a user's preferences can be achieved by constructing user profiles based on modified collaborative filtering.

1 Introduction

It has become increasingly difficult for users to find information on the WWW that satisfies their individual needs since information resources on the WWW continue to grow. Web search engines help users find useful information on the WWW. In order to achieve much better retrieval accuracy, hyperlink structures of the Web are focused on [3], [6], [13], [14]. However, when the same query is submitted by different users, these systems return the same result regardless of who submits the query. In general, each user has different information needs for his/her query. Therefore, Web search results should adapt to users with different information needs. Novel information systems designed to realize such systems have been proposed that personalize information or provide more relevant information for users: (a) systems using relevance feedback [2], (b) systems in which users register their interest or demographic information [8], and (c) systems that recommend information based on users' ratings [10], [7], [9], [12]. In these systems, users have to register personal information beforehand, or users have to provide feedback on relevant or irrelevant judgements, ratings, 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 by capturing changes of each user's preferences without any user effort.

J.X. Yu, X. Lin, H. Lu, and Y. Zhang (Eds.): APWeb 2004, LNCS 3007, pp. 636–643, 2004.

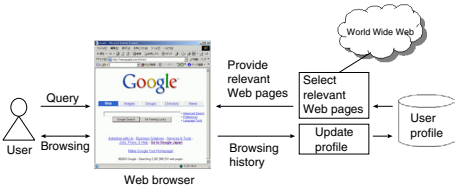

Fig. 1. Overview of our system.

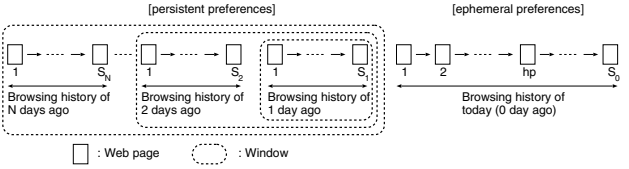

Fig. 2. Window size for constructing persistent user profile.

2 Our Proposed Method

Figure 1 shows an overview of our system. In the following sections, we explain how to construct a user profile in the update profile component illustrated in Figure 1. We construct each user profile based on the following two methods: (1) Pure browsing history, and (2) Modified collaborative filtering.

2.1 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. From these two factors, 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:

$$\boldsymbol{w}^{hp} = (w^{hp}_{t_1}, w^{hp}_{t_2}, \cdots, w^{hp}_{t_m}), \tag{1}$$

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:

$$w^{hp}_{t_k} = \frac{tf(t_k, hp)}{\sum_{s=1}^{m} tf(t_s, hp)}, \tag{2}$$

where $tf(t_k, hp)$ is the frequency of term t_k in each browsed Web page hp. We then denote user profile $\boldsymbol{P}^{today}$ as follows:

$$\boldsymbol{P}^{today} = (p_{t_1}^{today}, p_{t_2}^{today}, \cdots, p_{t_m}^{today}), \tag{3}$$

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

$$p_{t_k}^{today} = \frac{1}{S_0} \sum_{hp=1}^{S_0} w_{t_k}^{hp}, \tag{4}$$

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:

$$\boldsymbol{P}^{per} = (p_{t_1}^{per}, p_{t_2}^{per}, \cdots, p_{t_m}^{per}), \tag{5}$$

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

$$p_{t_k}^{per} = \frac{1}{S_N} \sum_{hp=1}^{S_N} w_{t_k}^{h_p} \cdot e^{-\frac{\log 2}{hl}(d - d_{t_{k\_init}})}, \tag{6}$$

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 occur, 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 (6). Using these parameters, we finally construct user profile $\boldsymbol{P}$ as defined in the following equation:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + cb\boldsymbol{P}^{today}, \tag{7}$$

where a and b are constants that satisfy $a + b = 1$, and c is a constant that shows to what extent our system reflect the contents of the Web page for each user profile. We define constant c as follows:

$$c = \begin{cases} 1;\ dr \geq Th \\ 0;\ dr < Th \end{cases} \tag{8}$$

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 preliminary experiments.

2.2 User Profile Construction Based on Modified Collaborative Filtering Algorithm

In the pure collaborative filtering algorithms, a user-item ratings matrix is usually considered [5]. Similarly, in the construction of a user profile, we can consider a user-term weights matrix like that shown in Figure 3(a). In addition, we can apply predictive algorithms in the pure collaborative filtering to predict missing term weights in each user profile. Our proposed algorithms are explained in the following steps:

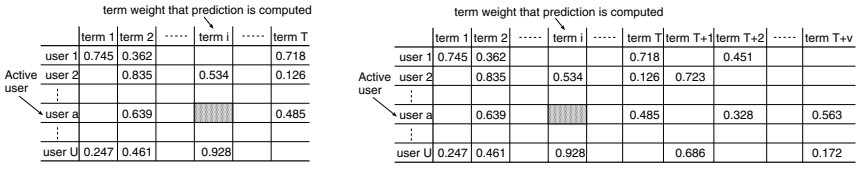

Fig. 3. User-term weights matrix for modified collaborative filtering [(a) when each user browsed k Web pages, (b) when each user browsed $k + 1$ Web pages].

(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,
(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, is computed using the Pearson correlation coefficient, defined below:

$$S_{a,u} = \frac{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a) \times (w_{u,i} - \bar{w}_u)}{\sqrt{\sum_{i=1}^{T}(w_{a,i} - \bar{w}_a)^2 \times \sum_{i=1}^{T}(w_{u,i} - \bar{w}_u)^2}}, \tag{9}$$

where $w_{a,i}$ is the weight of term i regarding user a computed based on term frequency in a browsed Web page, $\bar{w}_a$ 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 step (iii), predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{w}_a + \frac{\sum_{u=1}^{n}(w_{u,i} - \bar{w}_u) \times S_{a,u}}{\sum_{u=1}^{n} S_{a,u}}, \tag{10}$$

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 (9), and n is the number of users in the neighborhood.

3 Experiments

3.1 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 described in Section 2.1, and (3) user profiles based on the modified collaborative filtering algorithm described in Section 2.2. We used 50 query topics that were employed

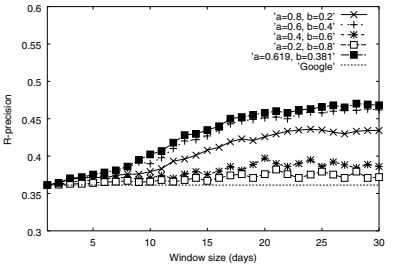

Fig. 4. R-precision obtained by relevance feedback-based user profile.

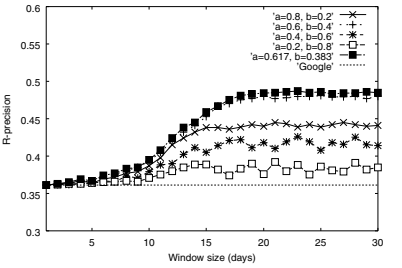

Fig. 5. R-precision obtained by pure browsing history-based user profile.

as test topics in the TREC WT10g test collection [4]. In the experiment, we observed the browsing history of 20 subjects for 30 days. In the following, let the i^{th} Web page in the search results and the user profile as defined by Equation (7) be rp_i and $\boldsymbol{P}$, respectively, Then, the feature vector of the i^{th} Web page rp_i in the search results, $\boldsymbol{w}^{rp_i}$, is defined as follows:

$$\boldsymbol{w}^{rp_i} = (w_{t_1}^{rp_i}, w_{t_2}^{rp_i}, \cdots, w_{t_m}^{rp_i}), \tag{11}$$

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

$$w_{t_k}^{rp_i} = \frac{tf(t_k, rp_i)}{\sum_{s=1}^{m} tf(t_s, rp_i)}, \tag{12}$$

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

$$sim(\boldsymbol{P}, \boldsymbol{w}^{rp_i}) = \frac{\boldsymbol{P} \cdot \boldsymbol{w}^{rp_i}}{|\boldsymbol{P}| \cdot |\boldsymbol{w}^{rp_i}|}. \tag{13}$$

Based on the value obtained in Equation (13), the search results are adapted to each user according to his/her profile. These results are compared with the search results of Google [3]. We then evaluate the retrieval accuracy using R-precision [1] ($R = 30$).

3.2 Experimental Results

3.2.1 User Profile Based on Relevance Feedback

Relevance feedback [11] is the most popular query reformulation strategy. In our experiment, we use the Rocchio formulation defined as follows:

$$\boldsymbol{Q}^{new} = \alpha \boldsymbol{Q}^{org} + \frac{\beta}{|D_r|} \sum_{\boldsymbol{d}_j \in D_r} \boldsymbol{d}_j - \frac{\gamma}{|D_n|} \sum_{\boldsymbol{d}_j \in D_n} \boldsymbol{d}_j, \tag{14}$$

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 (7), and employ $\boldsymbol{Q}^{new}$ as an initial preference of a user to construct a user profile. In this case, using Equation (7), user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + cb\boldsymbol{Q}^{new}. \tag{15}$$

We asked each subject to judge if the search result returned by the search engine is relevant, and constructed user profile $\boldsymbol{P}$ based on Equation (15). The number of feedback each user provided is just one time. Figure 4 shows the R-precision when the values of a and b are varied such that these values satisfy $a + b = 1$.

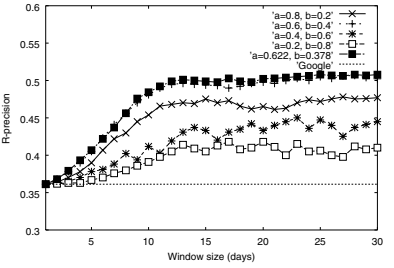

Fig. 6. R-precision obtained by modified collaborative filtering-based user profile (n=5).

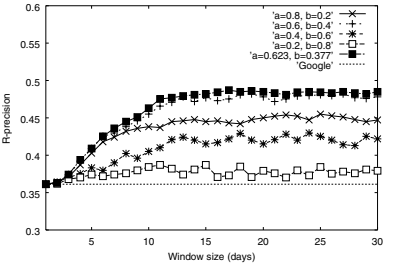

Fig. 7. R-precision obtained by modified collaborative filtering-based user profile (n=10).

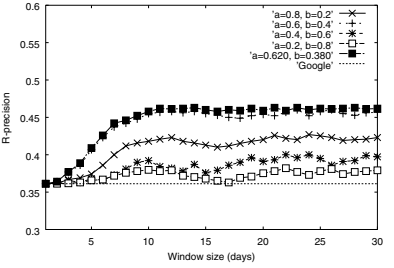

Fig. 8. R-precision obtained by modified collaborative filtering-based user profile (n=15).

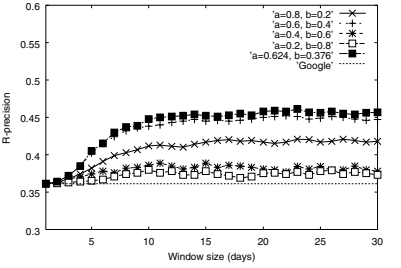

Fig. 9. R-precision obtained by modified collaborative filtering-based user profile (n=20).

3.2.2 User Profile Based on Pure Browsing History

In this approach, each user profile is constructed as mentioned in Section 2.1. As described in Section 2.1, user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + cb\boldsymbol{P}^{today}. \tag{16}$$

Figure 5 shows the R-precision when the values of a and b are varied such that these values satisfy $a + b = 1$.

3.2.3 User Profile Based on Modified Collaborative Filtering

In this approach, when the user browses a new Web page, a new term is added to his/her user profile. However, other users do not always browse the same page, so missing values occur in the user-term weights matrix as illustrated in Figure 3. These missing values are predicted using the algorithm described in Section 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 (7), and employ $\boldsymbol{V}^{pre}$ as an initial preference of a user to construct a user profile. In this case, using Equation (7), user profile $\boldsymbol{P}$ is defined as follows:

$$\boldsymbol{P} = a\boldsymbol{P}^{per} + cb\boldsymbol{V}^{pre}, \tag{17}$$

Figures 6 to 9 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 neighbors n are 5, 10, 15, and 20.

4 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 need for information. 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 conduct experiments with a greater number of subjects and attempt to improve our proposed approaches by using a longer term of the user's browsing history.

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*, 40(3):66–72, 1997.
3. 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)*, pages 107–117, 1998.
4. D. Hawking. Overview of the TREC-9 Web Track. *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*, pages 87–102, 2001.
5. 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)*, pages 230–237, 1999.
6. IBM Almaden Research Center. Clever Searching. http://www.almaden.ibm.com/cs/k53/clever.html.
7. 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*, 40(3):77–87, 1997.
8. U. Manber, A. Patel, and J. Robison. Experience with Personalization on Yahoo! *Communications of the ACM*, 43(8):35–39, 2000.
9. 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)*, pages 187–192, 2002.
10. 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)*, pages 175–186, 1994.
11. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
12. 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)*, pages 43–51, 2002.
13. 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)*, pages 891–901, 2002.
14. 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)*, pages 198–207, 2003.