PAPER

Personalized Web Page Recommendation Based on Preference Footprint to Browsed Pages

Kenta SERIZAWA[†], ***Nonmember***, **Sayaka KAMEI**[†a)], ***Member***, **Syuhei HAYASHI**[††], ***Nonmember***, ***and*** **Satoshi FUJITA**[†], ***Member***

SUMMARY In this paper, a new scheme for personalized web page recommendation using multi-user search engine query information is proposed. Our contribution is a scheme that improves the accuracy of personalization for various types of contents (e.g., documents, images and music) without increasing user burden. The proposed scheme combines "preference footprints" for browsed pages with collaborative filtering. We acquire user interest using words that are relevant to queries submitted by users, attach all user interests to a page as a footprint when it is browsed, and evaluate the relevance of web pages in relation to words in footprints. The performance of the scheme is evaluated experimentally. The results indicate that the proposed scheme improves the precision and recall of previous schemes by 1%–24% and 80%–107%, respectively.

key words: *personalization, information filtering, web search, relations among words, user profile*

1. Introduction

Because of the rapid increase in the amount of information on the web, huge amounts of resources are returned as search results by search engines such as Google and Yahoo!. It is often difficult for users to find the information they require from so many results. To reduce user workload, personalized web searches that customize search results based on user interest have been widely researched [1]–[3].

There are two approaches to personalized web search. One is a query extension that extends each query depending on user preference prior to sending queries to search engines [4]–[6]. The other approach is the modification of result lists by filtering or re-ranking depending on user preference after receiving result lists. In this study, we consider the latter approach.

The important parts of such schemes are creating user profiles that represent user interests and scoring web resources based on the user profiles [7]–[9]. It is most important to create accurate user profiles because personalization is performed on the basis of these profiles [8], [10]. Generally, we can create accurate user profiles if we force users to exert significant effort. However, many users do not want to invest significant effort for personalization [11]. Thus, it is very important to create accurate profiles without significant user effort. In addition, the accuracy of item scoring is important. If the scoring method is inaccurate, we cannot provide accurate personalization even though we can create accurate user profiles [11]. In some web search services, various types of web content, for example, documents, music, images, movies, among others, can be searched. In this study, we collectively refer to these different types of web content as "items." In addition, scoring methods should not depend on the types of items [10]. Moreover, for personalization schemes, privacy has also become increasingly important in recent years. Many personalization schemes assume that user histories and interests are stored on service providers' servers [1]. However, most users do not want their histories and interests to be identified: therefore, it is also important for personalization schemes to protect user privacy [12].

Our objective is to propose an information filtering approach to solve the above issues. We propose a Personalized recommendation scheme based on Relations among Search Queries (PRSQ), which is composed of a user profile method and a scoring method. In this paper, we make three main contributions:

- We propose a new scoring method which combines "preference footprints" for browsed pages and collaborative filtering.
- Our profile method creates accurate user profiles from user search terms (i.e., words in queries submitted by the users) and related terms.
- We also propose an approach to obtain related terms automatically without the use of dictionaries or user identification.

For our scoring method, we introduce the notion of "tags," which are similar to conventional social bookmark systems (SBSs); however, in contrast to SBSs, we associate such tags to each user. We use each user profile as a set of tags and attach all tags associated with a user to a browsed web page. Tags attached to pages, which are referred to as footprints, are shared by all users. As a result, we can acquire the information about the distribution of user interests that are relevant to a page without forcing each user to designate the type of page explicitly, which is required by SBSs. In other words, the footprints on the page indicate the type of users that have browsed that page. This significantly reduces user workload compared with conventional SBSs. In addition, it reduces the psychological resistance of users because it does not require registration, and determining user

Manuscript received March 29, 2016.
Manuscript revised July 5, 2016.
Manuscript publicized August 8, 2016.

[†]The authors are with the Graduate School of Engineering, Hiroshima University, Higashihiroshima-shi, 739–8527 Japan.
[††]The author is with NEC Corporation, Tokyo, 108–0014 Japan.
a) E-mail: s-kamei@se.hiroshima-u.ac.jp

DOI: 10.1587/transinf.2016EDP7134

preference is processed in a stochastic manner. Thus, we do not need to store user identification information, which allows us to protect user privacy. In addition, the footprints do not depend on the type of item; thus, we can score all items using footprints. We expect that the favorable properties of the proposed scheme will motivate many Internet users to use our scheme, which will increase opportunities to collect a significant number of tags compared with conventional tag-based page recommendation systems.

Our profile method creates accurate user profiles from user search histories without direct user involvement. Search histories are the easiest resources to obtain from search engines, and they can represent user preferences and interests. In the proposed profile method, user search terms and related terms are extracted from user histories. We can obtain terms searched by users automatically; therefore, we do not impose any user burden. The accuracy of user profiles is improved by employing both user search terms and related terms.

Our approach to obtain related terms automatically is based on the approach proposed by Eda et al. [13]; however we modify their approach to avoid user identification. The proposed strategy to detect related terms involves the relationships, which we refer to as relativity, between a pair of words. Relativity represents the co-occurrence of terms among users with similar interests.

The remainder of this paper is organized as follows. In Sect. 2, we define the requirements for personalized web search and introduce related work. In Sect. 3, we describe PRSQ. In Sect. 4, we discuss experimental evaluations using a dataset to show that PRSQ provides more accurate filtering results than previous schemes. We discuss an additional requirement for personalized web search and conclude the paper in Sects. 5 and 6, respectively.

2. Personalized Web Search

2.1 Requirements

Here we define the requirements for personalization schemes. When creating user profiles, accuracy and cost to the user are important factors.

- **Accuracy** of a user profile is measured by precision and recall. Precision is the ratio of the user profile that matches the actual interests of the user. Recall is the ratio of the user interests covered by the user profile.
- **User Cost** is the user effort required (other than search activities) to create a profile. The user cost is high if the user is forced to invest effort (e.g., selecting categories of interests; bookmarking favorite items). User cost should be low because high user costs can prevent systems from being developed as actual services.

In addition, for scoring items, accuracy, extensibility, and privacy are important.

- **Accuracy** of scoring also consists of precision and recall. Here let A be a set of items with high score, and let B be a set of items of user interest. Thus, precision and recall can be expressed as follows, respectively.

$$precision = \frac{|A \cap B|}{|A|}$$

$$recall = \frac{|A \cap B|}{|B|}$$

- **Extensibility** is the low dependency of scoring methods on item type. If we can score various types of items using the same method, the extensibility of that method is high.
- **Privacy** issue occurs if we can determine the interests of all users participating in the system at any time. For privacy issue, user histories or user profiles should be stored on a server without information that could be used for user identification.

2.2 Related Work

Search history has been researched extensively because search histories can be easily obtained from web search services. A search history is the history of the words or terms searched by users and the items selected by users from search results. It has been shown that selecting an item from a set of search results indicates the user's affirmation of the item [14]. Using search history facilitates extensibility and low user cost. However, search histories contain significant noise, which is typically generated by searching beyond user interests, clicking incorrect items, among others.

Personalization schemes using search history can be grouped into three types: relation-based schemes, term-based schemes, and category-based schemes. The relation-based schemes statistically estimate the features of queries, users, and items, or the relations among them. Sun et al. [1] proposed a scheme that estimates user interest in unknown terms and unknown items using singular value decomposition. However, privacy issues will occur because this scheme must identify all users of the system.

Term-based schemes create user profiles from the searched terms or words that exist in snippets[†] of items viewed by the user. Joachims et al. [15] determined that items ignored by a user are of no interest to the user and proposed a scheme using a Naive Bayes classifier [16]. Matthijs et al. [17] improved the effectiveness of personalization by changing the weights of words depending on the location (title, URL, etc.) they appeared. However, in term-based schemes, synonyms and polysemous words are problematic.

Several category-based schemes have been proposed to solve the issue of synonymous and polysemous words [11], [12], [18]–[20]. These category-based schemes can absorb the ambiguity of words by mapping user interest in words or terms as category interests. Category-based schemes can be classified according to the manner by which categories are created. One scheme uses categories that are created

[†]URL, title and description of item.

by humans [12], [18]. For example, categories by Open Directory Project (ODP) [21] and Wikipedia [22] are typically used in those schemes. Categories created by humans have an advantage; they are similar to categories that are assumed by the users. The other scheme uses system-generated categories that are created automatically [19]. This scheme can categorize huge amounts of words or items. However, the number of items or words in a given category is an issue for both types of category-based schemes. If a user is interested in a part of a category, but the category includes a lot of words or items in which the user is not interested, then items that the user is not interested in may be displayed as search results. If any category includes just one part of user interest, then items of interest may be excluded from the search results. Thus, setting appropriate constraints for the number of items or words in each category is a significant challenge.

3. Proposed PRSQ Scheme

In this section, we propose a new PRSQ page recommendation scheme that can effectively reflect individual user preference in recommendation results.

Figure 1 shows the PRSQ process flow. When a user u_i queries a search engine through the proposed system, the system requests a list of URLs from the search engine. At the same time, the system renews the user profile UP_i based on the query and a database of related words (DBRW). Once the list is received from the search engine, the proposed scheme filters the list according to the similarity between the requesting user profile and the footprints FP_j that are associated with each URL r_j on the list. The filtered list is then forwarded to the requesting user. The user then clicks a link on the list if the user is interested in the content of the web page. Upon receiving a user-selected URL, the scheme add that user profile to the set of footprints that are associated with the selected URL and renews the DBRW.

3.1 Details of the Scheme

Consider a case in which a user u_i searches a word q_1. Let T_i be the set of words that have been previously searched by u_i. We assume that the search history of u_i is stored as vector $UH_i = ((t_1, unum_i(t_1)), \cdots, (t_{|T_i|}, unum_i(t_{|T_i|})))$, where

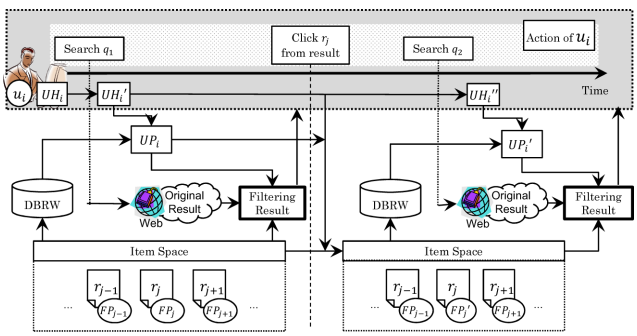

Fig. 1 Processing flow of PRSQ

$t \in T_i$ and $unum_i(t)$ is the number of times u_i searched t, in the space of u_i (e.g., cookie and web storage). We also assume that footprint FP_j is stored on our server for each item r_j. FP_j consists of three vectors, IH_j, P_j, and $Times_j$, which are defined as follows. Let n_j be the number of users who have viewed r_j. Let $U_j = \{u_1, u_2, \ldots, u_{n_j}\}$ be a set of users who have previously viewed r_j, and let $S_j = T_1 \cup T_2 \cup \cdots \cup T_{n_j}$ be a set of words that have been previously searched by users in U_j. Then, $IH_j = ((t_1, inum_j(t_1)), \cdots, (t_{|S_j|}, inum_j(t_{|S_j|})))$, where $t \in S_j$ and $inum_j(t) = unum_1(t) + unum_2(t) + \ldots + unum_{n_j}(t)$. This represents the set of words that are frequently used as queries by the users who have viewed r_j. $P_j = (p_1, p_2, \cdots, p_{k_j})$ is a vector in which each p is a merged user profile of users in U_j, and $Times_j = (times_j(p_1), \cdots, times_j(p_{k_j}))$, where $p \in P_j$ and $times_j(p)$ is the number of profiles used to create p. Then, k_j is the number of merged user profiles, i.e., p_1 is not a user profile of u_1 but is a profile created by merging one or more user profiles. User profile merging is described in Sect. 3.5. It should be noted that footprints do not include information that could be used for user identification.

First, PRSQ updates UH_i to UH_i' by incrementing $unum_i(q_1)$ when u_i searches q_1 as a query. Next, PRSQ creates the user profile UP_i from UH_i' and the DBRW as follows. Let RT_k be a set of words related to t_k in the DBRW, and let $V_i = T_i \cup RT_1 \cup RT_2 \cup \cdots \cup RT_{|T_i|}$ be the union of T_i and the set of related words for each word in T_i. For this case, $UP_i = ((t_1, w_i(t_1)), \cdots, (t_{|V_i|}, w_i(t_{|V_i|})))$, where $t \in V_i$ and $w_i(t)$ is the weight of t. The weight of a word represents the extent to which user u_i is interested in the word. To create UP_i, we assume the following hypotheses.

1. u_i is highly interested in words that u_i has frequently searched.
2. u_i frequently searches words that are related to the fields in which u_i is interested.

If u_i searches a word t_k frequently, it can be assumed that u_i requires information about t_k continuously. Therefore, we can assume the first hypothesis. However, in the real world, not all users search a single term many times. For example, we consider a user who is interested in "food." This user would search words that are highly related to food (e.g., pizza, pie, and meat) several times rather than searching the word "food" iteratively. In this case, we cannot find words that this user is interested in if we focus only on the frequency of searches for a given word. Thus, we assume the second hypothesis. If we know that "pizza," "pie," and "meat" are related in a "food" category, then we can determine that this user is interested in these words. In addition, we can also determine that this user is interested in the word "food" (i.e., category) even if this user does not search the word "food." The creation of UP_i from UH_i' and the DBRW are discussed in Sect. 3.2. To create the DBRW on our server, we use IH for every FP associated with each item. We calculate the strength of the relationship (relativity) between each pair of words from IH. It should be noted that the DBRW is shared by all users. We describe the cre-

ation of the DBRW in Sect. 3.3.

Next, PRSQ obtains a search result of q_1 from an existing search engine and calculates the score of each item in the results. To calculate the score, we assume that u_i is interested in an item viewed many times by users who have interests that are similar to those of u_i. The score of item r_j is calculated from the UP_i of u_i and P_j of FP_j. As a filtering result, PRSQ displays r_j only when the score of r_j is larger than $Threshold_{filter}$. Scoring and filtering is described in detail in Sect. 3.4.

Finally, when u_i clicks r_j from the filtering results, PRSQ updates FP_j to FP'_j by updating each composition. We will describe the detail of footprint updating in Sect. 3.5. After IH_j in FP_j is updated, PRSQ reconstructs the DBRW.

3.2 User Profile

Here we describe our method for creating user profiles. We create the user profile UP_i by performing the following process for each word t_k in UH'_i. Let $rel(t_k, t_{r_n})$ be the relativity between t_k and t_{r_n} where $t_{r_n} \in RT_k$. Note that the values of relativities rel between all pair of words are stored in the DBRW. First, we initialize the weight w_i of each word to 0. Next, we increase the weight of t_k depending on the number of times u_i searched t_k, i.e.,

$$w_i(t_k) = w_i(t_k) + unum_i(t_k).$$

Finally, we increase the weight of each related word t_{r_n} depending on its relativity to t_k as follows.

$$w_i(t_{r_n}) = w_i(t_{r_n}) + unum_i(t_k) \times rel(t_k, t_{r_n})$$

Thus, as users search t_k or its related words more frequently, the weight of t_k increases.

Here we discuss an example user profile. Assume that there is a user who searched "pizza" as its first search, and the DBRW is updated as shown in Table 1. When this user subsequently searches "pie," the user profile used at that time is updated as shown in Table 2. This user searched "pizza" and "pie" once; thus, the weights of these words are incremented by one. In addition, the weights of words related to "pizza" are incremented by their relativities to "pizza," and the weights of words that are related to "pie" are incremented by their relativities to "pie."

By using related words in this way, we can obtain words that a user is interested in even if such words are only searched a few times. Moreover, we can also determine the degree of user interests for words that have not been searched by that user.

3.3 Related Words

Many studies have focused on the determination of relationship among words [13], [23]. We use the approach proposed by Eda et al. [13] because it finds related words based on co-occurrence among users with similar interests, which matches our objective. However, they assume that users can be identified on their system. Thus, in PRSQ, we modified their approach to eliminate user identification.

Table 1 An example of the relativities among words in DBRW

	pizza	pie	meat	food	weather
pizza	-	0.5	0.5	0.7	0
pie	0.5	-	0.6	0.7	0
meat	0.5	0.6	-	0.7	0
food	0.7	0.7	0.7	-	0
weather	0	0	0	0	-

Table 2 User profile after searching "pizza" and "pie"

pizza	pie	meat	food	weather
1.5	1.5	1.1	1.4	0

Users with similar interests search many related words and view many related items. For example, users who are interested in cooking frequently search words related to cooking and view cooking-related items. Thus, we observe a domino effect; words are usually related if they have high $inum(\cdot)$ of IH among related items. In addition, items are usually related if they are associated with related words with high $inum(\cdot)$. We use EM algorithm [24] and JS divergence [25] to calculate the relativities among words. First, we assume that there is a fixed number of categories and optimize the probability that each word belongs to each category using the EM algorithm. Next, we treat the probabilities that a word belongs to each category as its probability distribution and calculate the distance between all pairs of probability distributions using JS divergence. Finally, we obtain the relativities between all pairs of words from their distances.

Here we explain the optimization of the probabilities of words belonging to each category. Let all words be $T = \{t_1, t_2, \cdots, t_Y\}$, all items be $R = \{r_1, r_2, \cdots, r_Z\}$, and let a set of categories be $D = \{d_1, d_2, \cdots, d_X\}$. In this study, we use $X = 80$, which is the same as [13]; however, this is an arbitrary value. Let $P(t_y|d_x)$ (resp. $P(r_z|d_x)$) be the probability that a word t_y (resp. an item r_z) belongs to a category d_x. Then, the probability that t_y is associated with the IH_z of r_z can be defined as follows.

$$P(t_y, r_z) = \sum_{x=1}^{X} P(t_y|d_x)P(r_z|d_x)P(d_x)$$

Thus, the log-likelihood LH of all $IH_1, IH_2, \cdots, IH_Z$ can be calculated as follows.

$$LH = \sum_{y=1}^{Y} \sum_{z=1}^{Z} inum_z(t_y) \times log\{P(t_y, r_z)\}$$

The EM algorithm optimizes $P(t_y|d_x)$ and $P(r_z|d_x)$ by maximizing the log-likelihood LH. Optimization is performed by repeating the following E-step and M-step alternately.

E-step:

$$P(d_x|t_y, r_z) := \frac{P(d_x|t_y)P(d_x|r_z)P(d_x)}{\sum_{x'=1}^{X} P(d_{x'}|t_y)P(d_{x'}|r_z)P(d_{x'})}$$

M-step:

$$P(d_x|t_y) := \frac{\sum_{z=1}^{Z} inum_z(t_y)P(d_x|t_y, r_z)}{\sum_{y'=1}^{Y}\sum_{z=1}^{Z} inum_z(t_{y'})P(d_x|t_{y'}, r_z)}$$

$$P(d_x|r_z) := \frac{\sum_{y=1}^{Y} inum_z(t_y)P(d_x|t_y, r_z)}{\sum_{y=1}^{Y}\sum_{z'=1}^{Z} inum_z(t_y)P(d_x|t_y, r_{z'})}$$

$$P(d_x) := \frac{\sum_{y=1}^{Y}\sum_{z=1}^{Z} inum_z(t_y)P(d_x|t_y, r_z)}{X}$$

In the EM algorithm, it is known that the likelihood will be larger than that of the previous step. However, the range of increase decreases depending on the number of steps. We terminate optimization when the range of increase becomes small and calculate $P(t_y|d_x)$ from $P(d_x|t_y)$ at that time using the Bayes theorem.

$$P(d_x|t_y) = \frac{P(t_y|d_x)P(d_x)}{P(t_y)}, \text{ where}$$

$$P(t_y) = \frac{\sum_{z=1}^{Z} inum_z(t_y)}{\sum_{y'=1}^{Y}\sum_{z=1}^{Z} inum_z(t_{y'})}.$$

Next, we calculate the distances of all pairs of words from their probability distributions. Related words will take similar probabilities for each category; thus, related words will demonstrate similar distributions. Therefore, we can obtain the relativity of each pair of words from their distances. Let PD_i and PD_j be the distribution of t_i and t_j, respectively. Then, the distance between PD_i and PD_j can be calculated as follows:

$$Dis(PD_i, PD_j) = H(\frac{PD_i + PD_j}{2}) - \frac{1}{2}H(PD_i) - \frac{1}{2}H(PD_j),$$

where $H(PD_i)$ is the entropy of the probability distribution PD_i.

Finally, we calculate the relativities between all pairs of words. The relativity between t_i and t_j can be calculated by the following equation using $0 < Threshold_{rel} < 1$:

$$rel(t_i, t_j) = \begin{cases} \frac{Threshold_{rel} - Dis(PD_i, PD_j)}{Threshold_{rel}} \\ \quad (Threshold_{rel} > Dis(PD_i, PD_j)) \\ 0 \quad \text{(otherwise)} \end{cases}$$

The relativity between t_i and t_j becomes 1 if PD_i and PD_j are the same, and the relativity is close to 0 if the distance between PD_i and PD_j is close to $Threshold_{rel}$.

3.4 Scoring and Filtering

Figure 2 shows the process for calculating the score of item r_j. First, we extract the set of user profiles P_j in FP_j that are associated with r_j. Next, we calculate the similarities between UP_i and each user profile in P_j. We then extract the subset of P_j as $RP_j = \{p_1, p_2, \cdots, p_M\}$ such that the similarity between UP_i and each $p_m \in RP_j$ is larger than

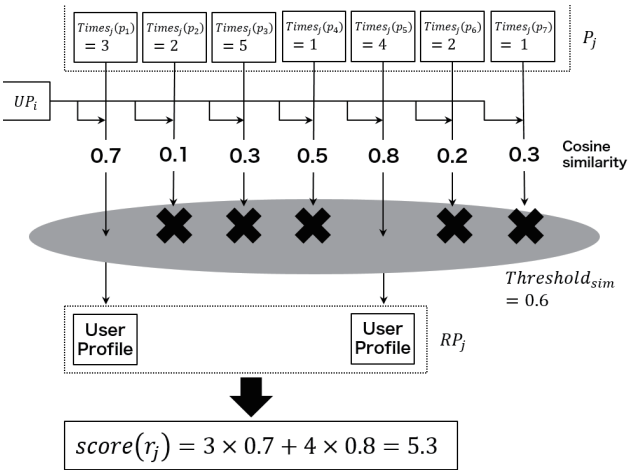

Fig. 2 Calculation of score of an item r_j

$Threshold_{sim}$. We treat the summation of these similarities as the score of r_j, i.e.,

$$score(r_j) = \sum_{p_m \in RP_j} \{times(p_m) \times sim(UP_i, p_m)\},$$

where $sim(UP_i, p_m)$ is the cosine similarity between UP_i and p_m as follows:

$$sim(UP_i, p_m) = \frac{UP_i \cdot p_m}{||UP_i||||p_m||}.$$

We use $Threshold_{sim}$ because the number of users who viewed r_j affects the score of r_j when we use all profiles in P_j. For example, we consider two items r_1 and r_2, where r_1 was viewed by 10,000 users whose similarities to user u_i are 0.1, and r_2 was viewed by 100 users whose similarities to u_i are 0.9. For this case, if we calculate the score from all profiles in P_1 and P_2, $score(r_1)$ will be larger than $score(r_2)$, even though more users with interests that are similar to u_i's interest view r_2 than r_1. Finally, we filter items using $Threshold_{filter}$, i.e., we recommend a set of URLs which scores are larger than or equal to $Threshold_{filter}$. To perform appropriate filtering, we must set $Threshold_{sim}$ and $Threshold_{filter}$ appropriately.

3.5 Footprint Updating

When u_i clicks r_j from the filtering results, IH_j is updated to $IH'_j = ((t_1, inum'_j(t_1)), \cdots, (t_{|T_i \cup S_j|}, inum'_j(t_{|T_i \cup S_j|})))$, where $t \in T_i \cup S_j$ and $inum'_j(t) = inum_j(t) + unum_i(t)$. P_j and $Times_j$ are updated as follows. If there exists a profile $p_k = ((t_1, w_k(t_1)), \cdots, (t_{|V_k|}, w_k(t_{|V_k|})))$, where $t \in V_k$ in P_j such that the similarity between p_k and UP_i is larger than $Threshold_{merge}$, PRSQ updates p_k to $p'_k = ((t_1, w'_k(t_1)), \cdots, (t_{|V_k \cup V_i|}, w'_k(t_{|V_k \cup V_i|})))$, where $t \in V_k \cup V_i$ and $w'_k(t) = w_k(t) + w_i(t)$. Note that if there are more than one such p_k, we choose the p_k with the maximum similarity. At this time, we increment $times_j(p_k)$ in $Times_j$. If there is no such $p_k \in P_j$, we update P_j(resp. $Times_j$)

to P'_j(resp. $Times'_j$) by adding UP_i(resp. $times_j(UP_i)$) to P_j(resp. $Times_j$) as a new component. Thus, we can reduce the amount of data on our server compared to a case in which we keep all profiles without merging. Such footprints are described in the form of XML.

4. Experimental Evaluations

4.1 Evaluation of User Profiles

4.1.1 Experiment Setup

We evaluated the accuracy of the proposed user profiles before performing an evaluation of the recommendation results because accurate user profiles are required for accurate personalization. However, because the contents of our user profiles are different from other existing profiles, we first evaluate our user profiles by the questionary investigation, and next by the recommendation results in the next subsection.

In this experiment, we contacted 25 students majoring in computer science and collected their three-month search histories to create their user profiles. Table 3 summarizes the participant statistics. The average number of words in user histories is the average number of words that are used as queries. Each user profile in the proposed scheme includes these words in the histories and their related words; thus, the average number of words in a user profile is far greater than in the user history.

However, the collected data were insufficient to create the DBRW; thus, we also used an AOL dataset [26] to create the DBRW. The AOL dataset is a collection of real query log data that is based on real users from March 1, 2006 to May 31, 2006. Each line of the dataset includes user ID, the query issued by the user, the time stamp, the rank of the item, and the URL. The rank of the item and the URL are included only if the user clicked on a search result. Table 4 shows a summary of the AOL dataset. Note that, in AOL dataset, there are many queries that were not followed by the user clicking on a result item. The number of query words is counted as follows. We count the number of discrete words submitted by each user. Then, we add up these totals. Of cource, we can use other dataset if it includes a lot of user search histories such that who sends what words as queries and views what.

Table 3 Statistics on the participants

Number of users	25
Avg # of searches	153
Avg # of words in histories (no duplicates)	53.9
Avg # of words in user profiles	12,902.7

Table 4 Statistics on the AOL dataset

Number of lines of data	36,389,567
Number of users	657,426
Number of user click-through events	19,442,629
Number of query words	27,165,898

For each participant, we randomly extracted 30 words from the user profile of the participant and administered questionnaires to determine the participant's interest for each extracted word on a scale of 1-5, where 5 indicates strong interest. It should be noted that each user profile has a high percentage of related words that are not in the user's history; thus, the extracted set of 30 words is more likely to include words that are not present in the history. To prevent blurring of answers, questionnaires were administered two times with the same set of words in different order. We treated the average of the two questionnaires as the participants' answers.

We used the influences of words in user profiles as evaluation metrics because the maximum weights of words in the user profiles vary among users. The influence of a word t_j for a user profile UP_i is calculated as follows.

$$Inf_i(t_j) = \frac{w_i(t_j)}{\max_{t'_j \in UP_i} w_i(t'_j)}.$$

In an ideal user profile, the influence of a word should be small if the user is disinterested, and the influence of a word should be large if the user is strongly interested.

4.1.2 Evaluation Results

Figure 3 shows the experimental results. The vertical axis represents the influences of words in the user profile, and the horizontal axis represents users' answers. If the user rated a word as 5, and its influence on the user profile is 0.8, we plot a point where answer = 5 and influence = 0.8.

According to the results, we find that most words with small rate have little influence. In Fig. 3, the domain where rate is $[1 \ldots 2.5]$ and influence is $[0.6 \ldots 1]$ is sparser than other domains. This indicates that, in PRSQ, erroneously assigning a high weight value to a word the user is not interested occurs infrequently. In addition, we find that some words with large rate have small influence and some have large influence. That is, we cannot assign high weight to all words of interest. For participants, the average correlation between rate and influence was 0.55, and the maximum (resp. minimum) correlation was 0.73 (resp. 0.41). This is

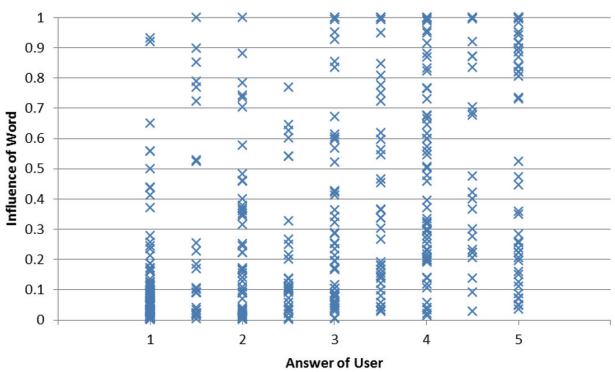

Fig. 3 The answers of users and the influences for each extracted word

because, users do not search all fields of interest sufficiently frequently. We expect that the raio of such words with large rate and small influence decreases after users search their interest fileds sufficiently.

Words with large influence primarily affect the scores of items. According to the results, we find that the proposed user profiles do not cover all user interests; however, most words with significant influence are words of interest. Thus, we expect that we can score items with relative equivalence to actual user interest.

4.2 Evaluation of Recommendation Results

4.2.1 Experiment Setup

Here we describe an experiment using the AOL dataset [26]. The experiment was conducted to evaluate the accuracy of recommendation results.

In this experiment, we split the dataset into two parts: the oldest 80% of the dataset was used as training data, and the remaining 20% was used as test data. The training data was used to create user profiles, the DBRW, and footprints. From each user profile and footprints, we deleted words with small influence to reduce computation time and space requirements. Thus, the size of each profile was less than 1 MB, and the average size of a footprint was 7.71 MB. It should be noted that the deletion of words with small influence has an insignificant effect on the score of each item. The test data was used to evaluate recommendation results.

First, we created user profiles, the DBRW, and footprints from all training data. Next, we extracted the set of users U who searched more than 50 times in the training data and viewed more than 10 items in the test data. We assumed that the items that $u_i \in U$ viewed in the test data were those in which u_i is interested. We refer to such items as positive items. Here let PI_i be the set of positive items. Table 5 shows a summary of the users in U. In this table, the number of positive items is the number of appearance patterns of triple (user ID, the query, the URL), where user ID is in U.

We conducted the following simulations for each user $u_i \in U$. We obtained the search results from Bing [27] for each word that u_i actually searched in the test data. Then, we scored each item in the result set using the schemes described in Sect. 4.2.2. For each scheme, we set the threshold value for filtering in such a way that all positive items were included in each recommendation list, i.e., we set $Threshold_{filter}$ for each user u_i as follows.

Table 5 Statistics on the users in U (unnormalized)

Number of users	66,874
Number of training data	14,475,061
Number of query words in training data	10,030,983
Number of test data	4,261,020
Number of query words in test data	3,158,373
Number of user click-through events in the test data	2,523,435
Number of positive items in the test data	1,963,612

$$u_i's\ Threshold_{filter} = \min_{r_j \in PI_i}(score(r_j)).$$

Then, we sorted the URLs in each recommendation list in descending order of scores. From the URL with the largest score, we calculate precision and recall values for each case in which the filtering threshold value is the score. Finally, we evaluated each scheme using the metrics described in Sect. 4.2.3.

4.2.2 Compared Schemes

In this experiment, we compared PRSQ to two baselines and three previously reported schemes. We set thresholds in PRSQ as $Threshold_{sim} = 0.6$, $Threshold_{merge} = 0.8$, and $Threshold_{rel} = 1$. These values were determined experimentally to yield the best results.

We compared PRSQ with two baselines to measure how much the accuracy of recommendation results improved using the related words. That is, the results also show how much the accuracy of our profiles. For the baselines, we created user profiles using term frequency-inverse user frequency (TF-IUF) and Okapi-BM25 [28] to weight words without related words. For both baselines, using such profiles, we create footprints and calculate the score for each item by the same way as PRSQ. That is, other settings such as scoring method, $Threshold_{sim}$ and $Threshold_{merge}$ were the same as PRSQ.

In addition, we compared PRSQ against three previous schemes using search histories to measure how much the accuracy of the recommendation results is improved by PRSQ. As mentioned in Sect. 2.2, personalization schemes using search histories can be grouped into three types: relation-based schemes, term-based schemes, and category-based schemes. We compared PRSQ with a representative scheme of each type.

- **Relation-based scheme**. We compared CubeSVD (CSVD) [1], which uses co-occurrence among users, items, and queries. We used CSLapack [29] for singular value decomposition.
- **Term-based scheme**. We compared SpyNB [14], which uses words in web snippets. We used Mecab [30] to extract words from Japanese snippets. For English snippets, we extracted words by detecting blanks.
- **Category-based scheme**. We compared the probabilistic model (PB) [18], which is a state-of-the-art category-based scheme. We obtained the list of categories from the ODP and optimized the parameters in PB using the EM algorithm.

These existing schemes contain methods for re-ranking result lists. These existing schemes also contain a scoring method, and the score obtained by each method represents the extent to which a URL satisfies a user preference. Therefore, we compared our algorithm against their scoring methods for recommendations, i.e., we did not use their re-ranking methods.

4.2.3 Metrics

We used the following metrics to evaluate the accuracy of PRSQ.

The first metric is *eleven-point interpolated average precision* (*11-points*) [31], which is used to measure the accuracy of scoring results by plotting the interpolated average precision for 11 standard recall points ($recall = 0.0, 0.1, 0.2, \cdots, 1.0$). For a recall point $recall = r_m$, the interpolated precision can be calculated as follows.

$$Prec(r_m) = \max_{r>=r_m} Prec(r),$$

where $Prec(r)$ is the precision at the point $recall = r$. An ideal scoring method assigns high scores to every positive item and low scores to other items. In that case, the interpolated average precision will be 1 for each recall point.

The other metric used was F_β [32], which is the harmonic mean of precision (P) and recall (R). This is used to measure the accuracy of the filtering results. F_β can be calculated by the following equation using a free parameter β.

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}.$$

In this study, we use $\beta = 1$, i.e., precision and recall were weighted equally. Unlike *11-points*, F_β is sensitive to thresholds.

4.2.4 Evaluation Results

Figure 4 shows a comparison of PRSQ and five other schemes using *11-points*. First, we focus on the accuracy of the items to which each scheme assigns high scores, i.e., the accuracy in the case that the filtering threshold value is large ($recall = 0$ to 0.2, Fig. 4). According to the results, the precision of PRSQ is better than that of state-of-the-art schemes, such as PB and SpyNB.

Next, we focus on accuracy where schemes need to provide a significant number of positive items ($recall = 0.2$ to 0.8, Fig. 4). From Fig. 4, it is evident that the precision of PRSQ is greater than that of other schemes. In the term-based schemes (SpyNB, TF-IUF, BM25), synonymous words and polysemous words were not considered. Thus, SpyNB cannot find items that have similar content using other words. TF-IUF and BM25 cannot find users who have similar interests but searched other words. Therefore, their precision is low. In addition, in the category-based scheme (PB), the precision is low if there are gaps between the categories assumed by the users and the categories recognized by the systems because the categories that users are interested in may contain non-positive items and may not contain all positive items related to the categories. In PRSQ, we can assign high weights to the words searched by the users and to the words strongly related to those words. PRSQ can find users who have similar interests but searched other words. Therefore, in PRSQ, we can find positive items with high precision.

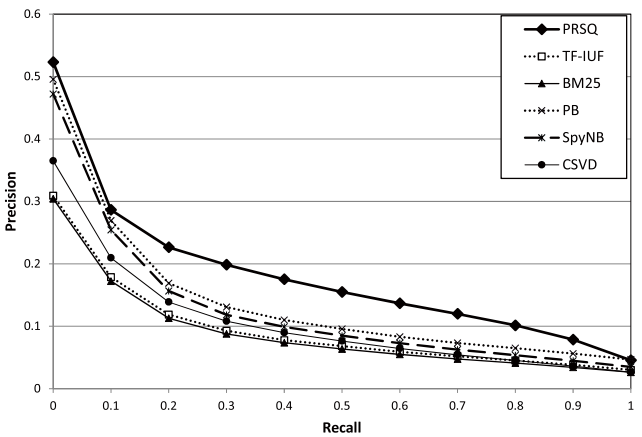

Fig. 4 11-points of each scheme

Finally, we focus on the accuracy of the case in which schemes need to provide almost all positive items, i.e., the accuracy in the case that the filtering threshold value is the original value $Threshold_{filter}$ ($recall = 0.8$ to 1.0, Fig. 4). According to the results, PRSQ's precision decreases sharply and becomes the same as that of the other schemes. This is because PRSQ scores the items based on the number of viewers who have interests similar to the users who performed the search. We cannot assign high scores to positive items viewed by only a few viewers; consequently, such positive items cannot be found and precision decreases sharply.

Table 6 shows a comparison of PRSQ with five other schemes for $F_{\beta=1}$. For each scheme, we calculate F_1 values at each point and present the highest value. From the result, it is evident that the F_1 value, precision, and recall of the proposed PRSQ scheme are higher than those for the other schemes. In the following, we will describe recall and precision in more detail.

The number of viewers affects item scores in PRSQ, TF-IUF, and BM25. If the user viewed an item viewed by only a few users and the filtering threshold value becomes small, a significant number of items are displayed as the filtering results. Thus, recalls tend to be high in these methods. Recall also tends to be high in PB and CSVD because these methods deal with synonymous and polysemous words. On the other hand, synonymous and polysemous words are not considered by SpyNB. Therefore, SpyNB can find only items that contain the same words as the items

Table 6 Recall, Precision and $F_{\beta=1}$ of each scheme

	Recall	Precision	F_1
PRSQ	0.444	0.172	0.248
TF-IUF	0.381	0.078	0.129
BM25	0.410	0.074	0.125
PB	0.247	0.170	0.201
SpyNB	0.215	0.156	0.181
CSVD	0.242	0.139	0.176

viewed by users in the past; therefore recall tends to be low.

In PRSQ, we can find many of users who have interests that are similar to each search user because we use related words. On the other hand, TF-IUF and BM25 can only find users who searched the same words. Thus, precision tends to be low in TF-IUF and BM25 because they cannot find the items viewed by users who searched the other words. In PB, the categories in ODP is updated by volunteers; therefore, they would be similar to the categories assumed by users. Consequently, PB precision is greater than that of term-based schemes, such as TF-IUF, BM25, and SpyNB. However, PB cannot deal with users who are only interested in a limited number of words or items in a given category. Therefore, the precision of PB is less than that of PRSQ. In SpyNB, synonymous and polysemous words are not considered, and the document frequencies of the words are not considered. Therefore, words used by many items (e.g., "image" and "video") are given the same weight as other specific words in SpyNB; thus, items in which the users are not interested are also displayed in many cases. As a result, precision tends to be low in SpyNB.

5. Discussion

In Sects. 3 and 4, we showed that PRSQ satisfies the requirements described in Sect. 2.1. In this section, we will describe *flexibility*, an additional requirement. User interest can be classified as long-term interests and short-term interests [33]. Long-term interests are relatively stable over time, and short-term interests may change for each search. Long-term and the short-term interests do not always correspond. In particular, if a user is interested in multiple fields, the field of interest should be changed for each query. Moreover, the part of the user profile related to the field would be different for each query (sometimes "gum" should be in the "food" field, but sometimes, it should be in a different field). Therefore, flexibility is important for personalization schemes. Flexibility facilitates changing filtering results for each search. However, it is impossible to predict the type of desired search filtering accurately. In this section, we consider the implementation of flexibility by providing an option to allow users to change their user profile filtering settings for each search. We can modify PRSQ as follows; prior to filtering, the user can view their user profile UP_i and the original results from the web. Then, the user can select applicable parts of the profile, and PRSQ can filter results using the selected filter settings.

To make it possible for users to easily make filter selections, we must visualize profiles in an approachable and usable manner. Generally, term-based user profiles contain a very large number of words. If we simply display all of the words, selecting words for filtering would be a heavy burden. However, if we display only the words with large weight or their categories, a user cannot select all words that cover their current interest.

In contrast, by using the relativities among the words, we can modify PRSQ to display each user profile in the following manner. We extract the representative word t_{max} with the highest weight from the user profile and group t_{max} and its related words. We display t_{max} as the center of the group and allocate other words in the group using the relativity between t_{max} and its related words (Fig. 5). By repeating this grouping of residual words until all words in the profile are displayed, we can divide the words into groups and display the words depending on their relativities. If a word is related to multiple groups, then it should be displayed in an area that is shared by both groups. In Fig. 5, we display words that are close to "pie" if they are strongly related to "pie" and display words at a distance from "pie" if they are weakly related to "pie." In addition, we display words that are related to both "pie" and "cake" in the area that is shared by both groups. By selecting words from such weakly grouped words, the user can easily select words that they want to use for filtering. It should be noted that the user should be able to select all or part of groups to perform filtering that best reflects their interests. Thus, we can flexibly provide effective user-based filtering by grouping according to weak relativities.

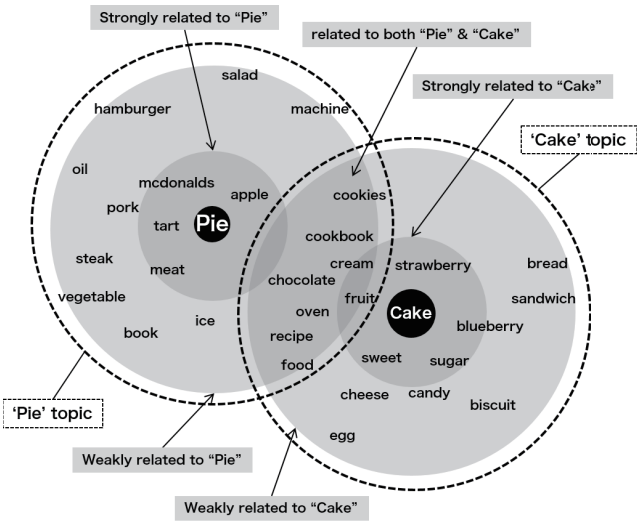

Fig. 5 An example of user interface to select a part of a user profile

6. Conclusion

In this study, we considered issues of personalized web search and have proposed PRSQ to solve those issues. First, we improved the accuracy of the user profile with low user cost using words searched by users and related words. We also improved the accuracy of scoring and resolved both privacy and extensibility issues. In addition, we evaluated the proposed user profiles and filtering results experimentally. According to the results of the experiments, the proposed profile method can yield high weights with high precision to words of interest. In addition, the proposed scoring method improves the precision and recall of previous schemes by 1%–24% and 80%–107%, respectively. This indicates that PRSQ can provide more positive items with higher precision

than previous schemes in the same recall points.

In future, we plan to improve the proposed method in several ways.

- The reduction of the processing time should be addressed. In PRSQ, after a user enters a query, many processes are performed to safeguard user identity. Thus, to reduce response time, we must reduce the number of processes required to provide filtering results. Especially, the construction and maintenance of DBRW take a long time. To reduce these time, it may be necessary to consider ways for parallelization.
- We intend to investigate ways to reduce the amount of data stored on our server: In this study, we merged similar user profiles to reduce the amount of data stored on our server. However, PRSQ is not scalable because the number of profiles associated with footprints becomes large when many users use the system. The large number of profiles in footprints causes significant response time. Thus, we must improve PRSQ by implementing scalability by, for example, reducing dynamically the number of words with small influence in each profile and each footprint. While typographical error words which user use are preserved in their profiles with small weight in PRSQ, these words are deleted after such improvement.
- PRSQ also has a cold start problem. Because DBRW and footprints are needed for recommendations, we need a lot of user histories before starting services. (Note that, users in the user histories are not necessary to be PRSQ user.)
- We also intend to implement frexibility (Sect. 5). This will involve calculating relativities among an extremely large number of words. Consequently, we will need to reduce the time it takes to calculate relativities.

References

[1] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, "Cubesvd: a novel approach to personalized web search," Proceedings of the 14th international conference on World Wide Web, pp.382–390, ACM, 2005.

[2] A. Hannak, P. Sapiezynski, A.M. Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson, "Measuring personalization of web search," Proceedings of the 22nd international conference on World Wide Web, pp.527–538, 2013.

[3] Y. Ustinovskiy, G. Gusev, and P. Serdyukov, "An optimization framework for weighting implicit relevance labels for personalized web search," Proceedings of the 24th International Conference on World Wide Web, pp.1144–1154, 2015.

[4] Z. Zhu, J. Xu, X. Ren, Y. Tian, and L. Li, "Query expansion based on a personalized web search model," Proceedings of the 3rd International Conference on Semantics, Knowledge and Grid, pp.128–133, 2007.

[5] J. Jayanthi, K.S. Jayakumar, and B. Akalya, "Personalized query expansion based on phrases semantic similarity," Proceedings of the 3rd International Conference on Electronics Computer Technology, pp.273–277, 2011.

[6] N. Limsopatham, C. Macdonald, and I. Ounis, "Modelling the usefulness of document collections for query expansion in patient search," Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp.1431–1440, 2015.

[7] D. Vallet and P. Castells, "On diversifying and personalizing web search," Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp.1157–1158, ACM, 2011.

[8] A. Sieg, B. Mobasher, and R. Burke, "Learning ontology-based user profiles: A semantic approach to personalized web search," IEEE Intelligent Informatics Bulletin, vol.8, no.1, pp.7–18, 2007.

[9] Z. Zhao, Z. Cheng, L. Hong, and E.H. Chi, "Improving user topic interest profiles by behavior factorization," Proceedings of the 24th International Conference on World Wide Web, pp.1406–1416, 2015.

[10] J.-W. Lee, S.-G. Lee, and H.-J. Kim, "A probabilistic approach to semantic collaborative filtering using world knowledge," Journal of Information Science, vol.37, no.1, pp.49–66, 2011.

[11] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," Proceedings of the 15th international conference on World Wide Web, pp.727–736, ACM, 2006.

[12] M. Speretta and S. Gauch, "Personalized search based on user search histories," Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp.622–628, IEEE, 2005.

[13] T. Eda, M. Yoshikawa, and M. Yamamuro, "Locally expandable allocation of folksonomy tags in a directed acyclic graph," Proceedings of the 9th International Conference on Web Information Systems Engineering, pp.151–162, 2008.

[14] W. Ng, L. Deng, and D. Lee, "Mining user preference using spy voting for search engine personalization," ACM Transactions on Internet Technology (TOIT), vol.7, no.4, p.19, 2007.

[15] T. Joachims, "Optimizing search engines using clickthrough data," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp.133–142, ACM, 2002.

[16] T. Mitchell, Machine learning, McGraw Hill, Burr Ridge, IL, 1997.

[17] N. Matthijs and F. Radlinski, "Personalizing web search using long term browsing history," Proceedings of the fourth ACM international conference on Web search and data mining, pp.25–34, ACM, 2011.

[18] D. Sontag, K. Collins-Thompson, P.N. Bennett, R.W. White, S. Dumais, and B. Billerbeck, "Probabilistic models for personalizing web search," Proceedings of the fifth ACM international conference on Web search and data mining, pp.433–442, ACM, 2012.

[19] K.W.-T. Leung, H.Y. Fung, and D.L. Lee, "Constructing concept relation network and its application to personalized web search," Proceedings of the 14th International Conference on Extending Database Technology, pp.413–424, ACM, 2011.

[20] E.F. Churchill and A.D. Sarma, "Data design for personalization: current challenges and emerging opportunities," Proceedings of the 7th ACM international conference on Web search and data mining, pp.693–694, 2014.

[21] "Open directory project." http://www.dmoz.org/

[22] "Wikipedia." http://en.wikipedia.org/wiki/Portal:Contents/Categories.

[23] K. Shinzato and K. Torisawa, "A simple www-based method for semantic word class acquisition," Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005, vol.292, p.207, 2007.

[24] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," Journal of the Royal Statistical Society. Series B (Methodological), pp.1–38, 1977.

[25] S. Kullback and R.A. Leibler, "On information and sufficiency," The Annals of Mathematical Statistics, vol.22, no.1, pp.79–86, 1951.

[26] G. Pass, A. Chowdhury, and C. Torgeson, "A picture of search," Proceedings of the 1st international conference on Scalable information systems, p.1, Citeseer, 2006.

[27] "Bing." https://datamarket.azure.com/dataset/bing/search.

[28] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," NIST SPECIAL PUBLICATION SP, pp.109–109, 1995.

[29] "Cslapack." http://www.dotnumerics.com/NumericalLibraries/LinearAlgebra/CSLapack/

[30] "Mecab." http://mecab.googlecode.com/svn/trunk/mecab/doc/
[31] C. Manning, P. Raghavan, and H. Schütze, Introduction to information retrieval, Cambridge University Press, Cambridge, 2008.
[32] C. Rijsbergen, "Information retrieval: theory and practice," Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems, pp.1–14, 1979.
[33] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," The Adaptive Web, pp.54–89, 2007.

Kenta Serizawa received the B.E., M.E. degrees in computer science from Hiroshima University in 2011 and 2013, respectively.

Sayaka Kamei received the B.E., M.E., and D.E. degrees in computer science from Hiroshima University in 2001, 2003, and 2006, respectively. She worked at the Tottori University of Environmental Studies and Hiroshima University as an assistant professor from 2006–2008 and 2008-2012, respectively. Now, she is an associate professor of the Graduate School of Engineering, Hiroshima University. Her research interests include distributed algorithms. She is a member of the IEEE Computer Society, ACM, IEICE, and IPSJ.

Syuhei Hayashi received the B.E., M.E. degrees in computer science from Hiroshima University in 2009 and 2011, respectively. He worked at the NEC Corporation from 2011.

Satoshi Fujita received the B.E. degree in electrical engineering, M.E. degree in systems engineering, and Dr.E. degree in information engineering from Hiroshima University in 1985, 1987, and 1990, respectively. After that, he worked at Hiroshima University as a research associate, and was promoted to an associate professor in 1995. Now, he is a professor of Hiroshima University. His research interests include communication algorithms on interconnection networks, parallel algorithms, graph algorithms, and parallel computer systems. He is a member of IEEE Computer Society, SIAM, IEICE, IPS, and SIAM Japan.