User Language Model for Collaborative Personalized Search

GUI-RONG XUE, JIE HAN, and YONG YU
Shanghai Jiao-Tong University
and
QIANG YANG
Hong Kong University of Science and Technology

Traditional personalized search approaches rely solely on individual profiles to construct a user model. They are often confronted by two major problems: data sparseness and cold-start for new individuals. Data sparseness refers to the fact that most users only visit a small portion of Web pages and hence a very sparse user-term relationship matrix is generated, while cold-start for new individuals means that the system cannot conduct any personalization without previous browsing history. Recently, community-based approaches were proposed to use the group's social behaviors as a supplement to personalization. However, these approaches only consider the commonality of a group of users and still cannot satisfy the diverse information needs of different users. In this article, we present a new approach, called collaborative personalized search. It considers not only the commonality factor among users for defining group user profiles and global user profiles, but also the specialties of individuals. Then, a statistical user language model is proposed to integrate the individual model, group user model and global user model together. In this way, the probability that a user will like a Web page is calculated through a two-step smoothing mechanism. First, a global user model is used to smooth the probability of unseen terms in the individual profiles and provide aggregated behavior of global users. Then, in order to precisely describe individual interests by looking at the behaviors of similar users, users are clustered into groups and group-user models are constructed. The group-user models are integrated into an overall model through a cluster-based language model. The behaviors of the group users can be utilized to enhance the performance of personalized search. This model can alleviate the two aforementioned problems and provide a more effective personalized search than previous approaches. Large-scale experimental

This research was supported by NSFC project 60873211 and Hong Kong RGC project 621307. This work was conducted while G.-R. Xue and J. Han were doing internship at Microsoft Research Asia. Q. Yang thanks Hong Kong ROC project 621307 for generous support. G.-R. Xue wishes to thank Microsoft Research Asia for their support to the MSRA-SFTU joint lab project.

Authors' addresses: G.-R. Xue, J. Han, and Y. Yu, Department of Computer Science and Engineering, Shanghai Jiao-Tong University, Minhang Distinct, Shanghai, China, 200240; email: {grxue,jhan, yyu@apex.sjtu.edu.cn}; Q. Yang, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong; email: qyang@cse.ust.hk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2009 ACM 1046-8188/2009/02-ART11 $5.00 DOI 10.1145/1462198.1462203 http://doi.acm.org/10.1145/1462198.1462203

evaluations are conducted to show that the proposed approach substantially improves the relevance of a search over several competitive methods.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.1.2 [**User/Machine Systems**]: Human factors

General Terms: Algorithms, Experimentation, Human Factors

Additional Key Words and Phrases: Collaborative personalized search, user language model, clustering, smoothing, data Sparseness, cold-start

ACM Reference Format:

Xue, G.-R., Han, J., Yu, Y., and Yang, Q. 2009. User language model for collaborative personalized search. ACM Trans. Inform. Syst. 27, 2, Article 11 (February 2009), 28 pages. DOI = 10.1145/1462198.1462203 http://doi.acm.org/10.1145/1462198.1462203

1. INTRODUCTION

Most existing Information Retrieval (IR) models, such as vector space models [Salton et al. 1975], probabilistic models [Robertson et al. 1980], and statistical language models [Ponte and Croft 1998], measure the relevance of documents against a given query without distinguishing between users. In fact, different users have different information needs that may not be accurately or completely represented by a single query. For example, the query "apple" submitted by two different users may represent the computer company or a kind of fruit accordingly. Personalized search has been proposed to satisfy the various information needs of different users, which attempts to adapt search results to enhance the relevance to users according to their past search behaviors.

Two general approaches of personalized search have been described in [Pitkow et al. 1996; Teevan et al. 2005]: query modification or query expansion based on user profiles [Liu et al. 2002], and reranking the search results using individual profile information. The latter is based on the use of the similarity of individual profiles and the Web page contents given in the search results [Haveliwala 2002; Teevan et al. 2005; Shen et al. 2005; Sugiyama et al. 2004; Chirita et al. 2005]. Most of the methods in these two approaches will need to have a user model, either manually or automatically constructed up front. They then adapt the search results to an individual user according to the user model. For example, Teevan et al. [2005] have proposed a rich profiling model that considers all search histories, documents, and emails. However, it is still difficult to implement a practical personalized search system because accurate personal information is hard to obtain. There are two major problems in personalization as described in Anand [2005], Xue et al. [2005], and Shahabi and Chen [2003]:

—*Data sparseness*. Since each user only submits a small number of queries and only a very small portion of Web pages are visited by a single user, historical search records for the user are sparse, making the mining task extremely difficult. This problem is also compounded by the fact that there are a lot of new pages emerging every day. On average, 5% of Web content is new information added into the Web weekly [Ntoulas et al. 2004], especially news-related content. Thus, a fixed individual user profile can hardly keep up with the changes.

—*Cold-start for new individuals*. Since new Web users use search engines without previous search histories, there is often no preference information available for personalization. As a result, it is often impossible to pursue any personalized search for these new or unregistered individuals. Currently, most of the solutions are based on explicitly constructing user profiles. Other applications [Pazzani 1999; Krulwich 1997] make personalization based on demographic information, such as gender, age, occupation and so on. Using the "wisdom of crowds" can also address this problem by considering the aggregated behaviors of a large number of users to build the profile for each individual, such as DirectHit (http://www.directHit.com) and Agichtein et al. [2006]. Mainly, these approaches do not consider the diversity among different users. As mentioned in Agichtein et al. [2006], these techniques perform well on average, but fail to work for the queries with divergent access patterns (e.g., for ambiguous queries with multiple meanings).

Although these problems pose serious difficulties in personalized search, there are still opportunities to find a solution. In order to deal with the data sparseness problem on constructing user profiles, collaborative filtering has been proposed in the product recommendation area to utilize the behaviors of like-minded users [Breese et al. 1998]. There are many successful commercial collaborative filtering applications, such as Amazon.com,[1] Hollywood Video,[2] Musicmatch.com[3] and the like. The basic assumption of collaborative filtering algorithms [Sarwar et al. 2001; Resnick et al. 1994; Konstan et al. 1997] is that a user prefers information that similar users prefer, or that dissimilar users do not prefer. Applying this assumption to personalized search, similar users can be defined as these who share search behaviors with that of a particular user. Then, the algorithms make a prediction on behalf of the user based on the behaviors of similar users. According to the basic algorithm of collaborative filtering, we can observe that the user's interest in missing items can be predicted based on the behaviors of similar users. The same idea has been proposed and implemented in the I-SPY project [Smyth et al. 2003] and in our previous work [Xue et al. 2005]. Thus, such methods can be utilized on these users' historical search data to solve the sparse data problem.

In this article, we take these ideas forward and propose a novel integrated method for personalized search: collaborative personalized search. By assuming that users with similar behaviors may have similar tastes in Web content, we propose to cluster *like-minded users* as user groups to help construct individual profile models. From the behaviors of a user group, the problem of sparseness for user data can be alleviated. Furthermore, to address the cold-start problem for a new user, this model can utilize the behaviors of global users that consist of all users' behaviors to help newcomers construct their initial profiles. In this general model, the behaviors of individual, group, and global users are considered as a whole.

[1]http://www.amazon.com.
[2]http://www.hollywoodvideo.com.
[3]http://www.musicmatch.com.

In our model, individual user profiles are constructed by the submitted queries and the content of visited Web pages. In order to describe the user's interests precisely, *short-term* and *long-term* profiles are applied to represent different interests of individuals in different periods. Then we can construct a group user profile and a global user profile based on the profiles of a group of individuals and all individuals respectively. Based on the profiles of individuals, group users and global users, we can construct three separate probabilistic models for the individual, the group and the global users to estimate the probability to a Web page. We denote them as the *individual model*, *the group user model,* and *the global user model*, respectively. In order to combine the three different models into a unified model, a cluster-based language model is applied to provide flexible interpolation. More specifically, the probability that a user likes a Web page is calculated through a two-step smoothing process. In this process, first, a global user model is used to smooth the probability of unseen terms in the individual profiles. In order to precisely describe the user's interests using similar users' behaviors, users are clustered into groups. Based on which, group-user models are constructed. The group models are then integrated into an overall model through a cluster-based language model. Thus, the behaviors of the group can be utilized for enhancing the performance of personalized search. The interpolating parameters are tuned by well-known methods used in the language model. As a whole, the three types of data sources are integrated seamlessly. We define the whole integrated user model as a user language model.

To validate our approach, large-scale experiments have been conducted in this paper. Our experimental results show that, by combining the profiles from other users, the performance of personalized Web search can be improved significantly. Furthermore, the sparseness and the cold-start problems can be alleviated by using the models from the group and global users' behaviors.

The remainder of the article is organized as follows. In Section 2, the literature focusing on personalized search systems and language modeling approaches for IR is reviewed. In Section 3, we introduce the preliminary language model for personalized search. In Section 4, the user language model is proposed to conduct the collaborative personalized search by flexibly integrating different user models. In Section 5, we present the hypotheses and experiments. We conclude the article with a discussion of future work in Section 6.

2. LITERATURE REVIEW

In this section, we review several major research papers on personalized search and language models.

2.1 Personalized Search

There have been several prior attempts to achieve personalized search. Generally, a personalized system tries to adapt the provided information to the needs of a user based on a user model. According to different information sources for constructing user profiles, the user model can be classified into three

categories: modeling based on individual behaviors, on group behaviors, and on global behaviors.

2.1.1 *Modeling Based on Individual Behaviors.* In this category, a user profile model is constructed based on individual behaviors only. Most of the models for personalization belong to this category. The models in this category can be further divided into two categories: explicitly constructed and implicitly constructed, by considering whether they need the user's input or not.

Some good examples of explicitly constructing user profiles include My Yahoo! [Manber et al. 2002] and Google Personal.[4] My Yahoo! allows users to explicitly build their own profiles by choosing a set of categories. The system removes any information that is irrelevant to the user profile, showing only the sections and details that the user may be interested in. In topic-sensitive PageRank [Haveliwala 2002], the system first precomputes Web pages based on categories in the Open Directory Project[5] and then ranks the search results according to the user's selection of favorite categories. Some other researchers have tried to predict the users' intentions by relevance feedback and query refinement [Koenmann et al. 1996; Anick et al. 2004]. These two methods ask for the user's additional interaction with the systems. As described in [Beaulieu et al. 1996; Belkin et al. 1996; Anick et al. 2004; Fox et al. 2005], most users are unwilling to use the function of query refinement on a real Web search. Even when people are motivated enough to expend additional efforts on specifying their search intent, they are not always successful in doing so.

Many existing approaches focus on mining the users' preference profiles by using automatic and implicit feedback methods [Kelly et al. 2003]. A wide range of implicit activities of individuals have been taken into consideration, including previous search queries, browsed Web pages, client data and so on. For example, Dumais et al. [2003] and Teevan et al. [2005] have proposed a rich model that takes search histories, documents, and e-mails into account, while Shen et al. [2005] use browsing histories and query sessions to construct short-term individual models for personalized search. In addition, a mixture model consisting of short-term and long-term user models is proposed by Billsus and Pazzani [1999]. There are also many systems based on implicit feedback, including Letizia [Lieberman 1995], WebWatcher [Mladenic 1996], Syskill & Webert [Pazzani et al. 1996], and Websifter[Kerschberg et al. 2001]. These studies have exploited the different behaviors of individuals to build a personal model. However, because they are constructed based purely on the behaviors of individuals, the personal data is still too sparse to result in a good model, and the ability to deal with the new emerging Web content is poor.

2.1.2 *Modeling Based on Group Behaviors.* Group behaviors are based on the activities from like-minded users, which is also called social search Sullivan [2004] by using the "wisdom of crowds" [Surowiecki 2004]. As described in Sullivan [2004], Eurekster realized a personalized social search by filtering the irrelevant search results based on the search behaviors of a user's friends.

[4]http://labs.google.com/personalized.
[5]ODP http://dmoz.org.

Kritikopoulos and Sideri [2003] Almeida et al. [2004] build Web communities based on users' interests and make recommendations based on the behaviors of the community. Sugiyama et al. [2004] also proposed a method to construct user profiles by utilizing the top K similar users' profiles. Since it uses a K-nearest-neighbor algorithm to calculate the similarity between users, the algorithm is difficult to scale due to its poor performance on sparse data. In addition, the study does not provide a solution for the cold-start problem. Smyth et al. [2003] propose a collaborative Web search approach by using a community of like-minded users, which is generated according to domain-specific search engines. However, the method only focuses on the behavior of a group of users, while the information from individuals is ignored. However, such information is very important for personalization.

Many learning based techniques in collaborative filtering can be applied to solve the sparseness problem. For example, some methods aim to directly reduce the dimensionality of the user-item matrix. A simple strategy is to form clusters of users or items and then use these clusters as basic units in making recommendations. Principle Component Analysis (PCA) Goldberg et al. [2001] and Latent Semantic Indexing (LSI) [Fisher et al. 2000; Sarwar et al. 2000] have also been proposed. Zeng et al. [2003] compute users' similarities removing unrepresentative or insignificant user-item pairs so as to condense the user-item matrix. However, some potentially useful information might also be lost during this reduction process [Xue et al. 2005]. The cluster-based smoothing method proposed by Xue et al. [2005] can achieve better performance than the above works. By using the rating information from a group of closely related users, unrated items of the individual users in a group can be predicted, which allows the missing values to be filled in. This article extends the basic idea of cluster-based smoothing and proposes a user language model for personalized search.

2.1.3 *Modeling Based on Global Behaviors.* A third category for user modeling is based on aggregated global behaviors of users to adapt search results to a user's needs. A well-known algorithm for this category is DirectHit, which utilizes the behaviors of global users to help make recommendations. Joachims [2002] introduced a technique based entirely on clickthrough data to learn a personalized ranking function. Agichtein et al. [2006] also proposed to predict the Web search result preferences for global users by learning from different kinds of implicit features of global users' activities, which include query-text features, browsing features, and clickthrough features. However, these algorithms do not consider the diversity among different users.

In contrast to the previous methods, in this work, we integrate the behaviors of individuals, group users, and global users together, where we provide a rich representation to alleviate the sparseness and cold-start problems of personalized search. Furthermore, a sound statistical user language model is applied to integrate these models into a unified framework by using smoothing techniques. As well as content, clickthrough data, which is collected by a Web search engine, is used to construct user profiles. With this approach, no additional effort from users is solicited to specify or maintain their profiles.

2.2 Language Model for Information Retrieval

Language modeling has been applied successfully in speech recognition. Ponte and Croft first applied the language modeling techniques in information retrieval (IR) [Ponte and Croft 1998]. Given the document d and the query q, the relevance-ranking problem is cast as the computation of the probability $P(d \mid q)$, which is calculated through:

$$P(d \mid q) \propto P(q \mid d)P(d), \tag{1}$$

where d represents a document and q represents the query.

In practice, $P(d)$, the prior probability that a document is relevant to any query, is assumed to be uniform.

The basic idea is to estimate the conditional probability $P(q \mid d)$, that is the probability of generating a query q by the given document d. Assuming that a query is generated under a multivariate Bernoulli model, several heuristic smoothing strategies have been proposed to estimate the Maximum Likelihood Estimate (MLE) of the document language model. It is common in most studies that the bag-of-words assumption is applied to estimate the probability of the sequence as the product of the probabilities of the individual terms.

$$P(q \mid d) = \prod_{w \in q} P(w \mid d). \tag{2}$$

Since a language model estimated from the term occurrences in a given document is very sparse, a necessary step for this estimation is to perform smoothing for the unseen query terms in the document. This is typically accomplished by incorporating the probability of the unseen term in the collection as a whole through one of the many smoothing methods available. For example, one smoothing function is as given as follows,

$$P(w \mid d) = \lambda P_{ML}(w \mid d) + (1 - \lambda)P_{ML}(w \mid Collection), \tag{3}$$

where λ is a general parameter for smoothing. For different smoothing methods, such as Jelinek-Mercer smoothing and Bayesian smoothing using Dirichlet priors, λ takes different forms. More information on smoothing methods can be found in Zhai and Lafferty [2004].

Lafferty and Zhai [2001] proposed a different language model for IR in which each document is estimated, as well as a language model for each query. Using these models, the document retrieval problem is cast in terms of a problem of risk minimization. Recently, many variations on these traditional language models have been developed to improve the performance of IR, such as relevance-based language model [Lavrenko and Croft 2001], time-based language model [Li and Croft 2003], title-based language model [Jin et al. 2002], and domain model [Bai et al. 2007]. Furthermore, Wang et al. [2006], extended the basic language model for log-based collaborative filtering by a user-item relevance model and smoothing is also applied to solve the data sparsity.

Another related work is the cluster-based language model for IR, which is used to filter any relevant documents. This type of model has been applied in the fields of Topic Detection and Tracking (TDT) [Allan at al. 1998; Yamron et al. 1999; Spitters and Kraaij 2001] and distributed information retrieval [Xu and

Croft 1999] to help find the relevant topic. The model can be described as follows: first, document clustering is used to organize collections around topics. Each cluster is assumed to be representative of a topic, and only contains documents related to that topic. Then, for a given query, language models are estimated for the clusters and are used to properly represent topics. Finally, the related topics are selected according to the cluster-based language model.

Furthermore, some researchers [Brown et al. 1992; Chen and Goodman 1998] have suggested that a cluster-based language model is effective in dealing with the data sparseness problem and has been successfully applied to machine translation [Brown et al. 1992] and information retrieval [Liu and Croft 2004]. For example, Liu and Croft [2004] proposed that cluster information can improve the performance by using topic-based information to smooth the language model instead of whole collection based smoothing, which is calculated by:

$$\begin{aligned} P(w \mid d) \;=\; & \lambda P_{ML}(w \mid d) \\ & + (1-\lambda)(\sigma P_{ML}(w \mid \mathit{Cluster}) \\ & + (1-\sigma) P_{ML}(w \mid \mathit{Collection})), \end{aligned} \tag{4}$$

where *Cluster* is the topic that the document belongs to and σ is a smoothing parameter.

In this article, the traditional language model for IR is extended for user modeling. In addition, a cluster-based language model is adapted to provide a flexible representation for user modeling.

3. PRELIMINARY

Generally speaking, personalized search aims to provide relevant information to individuals based on their profiles. In this paper, we focus on the issue of reranking the top N search results according to a user's interests. We explore how to learn the user model from the Web search history data (e.g., clickthrough data), which contain submitted queries and the Web pages visited by a number of users. In this work, the similarity between document model and query model can be estimated by Kullback-Leibler (KL) divergence [Zhai and Lafferty 2001] between their models. In the following, the traditional retrieval model based on KL divergence is introduced and a personalized retrieval model is given.

3.1 Traditional Retrieval Model

According to Zhai and Lafferty [2001] and Shen et al. [2005], the advantage of KL divergence is that it can naturally incorporate user behaviors as additional evidence to improve the estimation of the query language model. The parameters of the query model are represented as θ_q while the parameters of the document model are represented as θ_d. The retrieval task involves estimating a query language model $P(w \mid \theta_q)$ for a given query q and a document language model $P(w \mid \theta_d)$ for a document d, and then computing their KL divergence $D = (\theta_q \| \theta_d)$, which is calculated as:

$$D = (\theta_q \| \theta_d) = \sum P(w \mid \theta_q) \log \frac{P(w \mid \theta_q)}{P(w \mid \theta_d)}, \tag{5}$$

where $P(w \mid \theta_q)$ is the probability of the word w in the observation of query q and $P(w \mid \theta_d)$ is the probability of the word w in the observation of document d.

According to [Lavrenko and Croft 2001], the probability $P(w \mid \theta_q)$ can be estimated on unigram language models by maximum likelihood estimation (MLE):

$$P(w \mid \theta_q) = P(w \mid q) = \frac{tf(w,q)}{\sum_{w' \in q} tf(w',q)}, \tag{6}$$

where $tf(w,q)$ is used to denote the term frequency of a word w in the query q.

The KL divergence between the query model and the document model serves as the ranking score.

3.2 Personalized Retrieval Model

Let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of m users and q be the query. The user $u \in U$ is represented by a vector of terms, which are constructed from the submitted queries and the clicked Web pages by u. Similar to the work presented by Shen et al. [2005], the task for personalized search is to estimate the parameter θ for personalized query model $P(w \mid \theta)$, based on the current query model $P(w \mid \theta_q)$ as well as the user model $P(w \mid \theta_u)$.

Two models $P(w \mid \theta_q)$ and $P(w \mid \theta_u)$ are interpolated by:

$$\begin{aligned} P(w \mid \theta) &= \lambda P(w \mid \theta_q) + (1-\lambda) P(w \mid \theta_u) \\ &= \lambda P(w \mid q) + (1-\lambda) P(w \mid u), \end{aligned} \tag{7}$$

where $P(w \mid u)$ is the probabilistic estimation that how is the word w interesting to the user u, λ is a general parameter for smoothing. For different smoothing methods, λ takes different forms. For example, λ can simply be an arbitrary fixed weight between 0 and 1, which is Jelinek-Mercer smoothing [Zhai and Lafferty 2004].

The Jelinek-Mercer smoothing approach fixes the parameter λ for all queries. One observation is that, if a query is very long, which means that the user has provided enough information for the search engine, the query should be trusted more. On the contrary, when the query has just one or two words, more weight should be put on the user's personal profile. Hence, we treat λ as the result of assuming a Dirichlet prior for the multinomial $P(w \mid \theta_u)$ [Zhai and Lafferty 2004] and query q as observed data to estimate a personalized query model by the Bayesian estimator. We denote the method as Bayesian smoothing with Dirichlet prior, and λ takes the following form:

$$\lambda = \frac{\sum_{w' \in q} tf(w',q)}{\sum_{w' \in q} tf(w',q) + \mu}, \tag{8}$$

where w' is any word in query q and μ is the Dirichlet smoothing parameter.

In the next section we propose our methods for estimating the probability $P(w \mid \theta_u)$.

4. COLLABORATIVE PERSONALIZED SEARCH BY USER LANGUAGE MODEL

As mentioned in the introduction section, most of the existing user-modeling methods for personalized search estimate the probability $P(w \mid \theta_u)$ only based

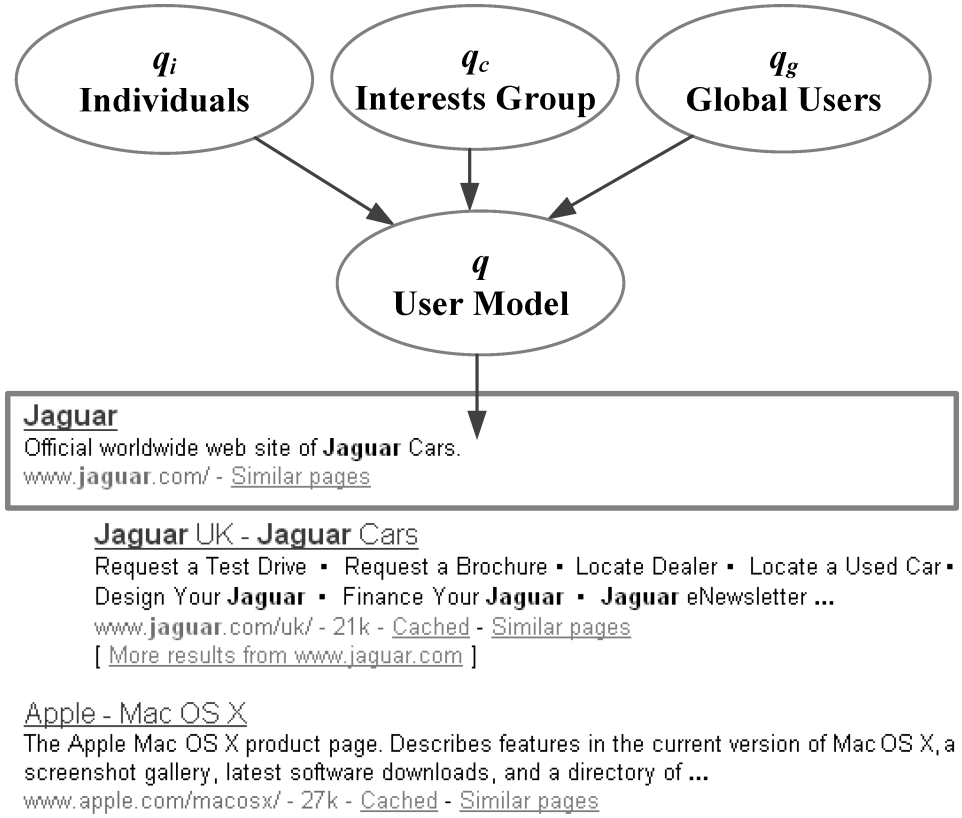

Fig. 1. Collaborative personalized search.

on an individual's data, which leads to the problems of data sparseness and cold-start for new individuals. In order to give a rich representation of a user model and to alleviate both personalization problems, we focus on exploiting different models for user u to estimate the probability $P(w \mid \theta_u)$. As described in Section 1, there exists some commonality among like-minded users. Intuitively, an individual user model θ_i, a group user model θ_c and a global user model θ_g are each valuable for improving search accuracy. We propose to integrate these three models to provide more precise user models, as shown in Figure 1. We show that statistical smoothing is effective as a solution for the sparseness and cold-start problems.

In the following, we first describe how to build different kinds of user models. Then, based on the smoothing methods in the language modeling approach for information retrieval, we propose our user language model for integrating these three models together under a unified framework.

4.1 Purely Individual Model Construction

We consider the profiles of each user as consisting of two aspects: long-term profiles and short-term profiles [Billsus and Pazzani 1999]. The long-term profiles

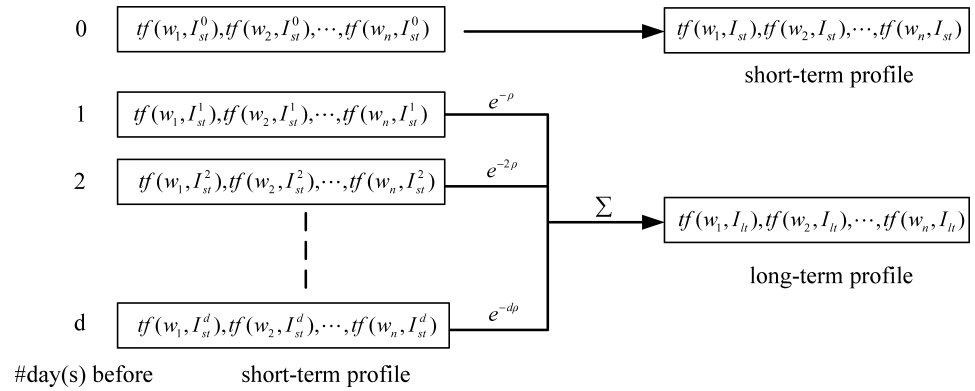

Fig. 2. Profile calculation for long-term and short-term profiles.

refer to information such as a user's long-term interests, which is reflected by the accumulated history of the user's search and browsing activities including submitted queries and browsed Web pages. The long-term profiles are generally stable for a long time and are accumulated over time. Meanwhile, users usually perform different tasks in one day and they probably perform several searches and browsing activities during that period. Such information can be categorized into short-term profiles. The individual's general profiles, the short-term profiles and the long-term profiles are denoted as I, I_{st}, and I_{lt}, respectively. We denote the user model based on the individual's general profiles I, the short-term profiles I_{st} and the long-term profiles I_{lt} are denoted as θ_i, $\theta_{i\text{-}st}$ and $\theta_{i\text{-}lt}$, respectively.

A short-term profile I_{st} consists of queries issued and Web pages browsed by the users over a short period, such as a one-day period, as shown in Figure 2. The probability $P(w \mid \theta_{i\text{-}st})$ in a short-term profile is defined as:

$$P(w \mid \theta_{i-st}) = \frac{tf(w, I_{st})}{\sum_{w' \in I_{st}} tf(w', I_{st})}, \tag{9}$$

where w' is any word in the short-term profile I_{st}.

As shown in Figure 2, a long-term profile is the accumulation of the previous d days' short-term profiles, and is calculated by:

$$tf(w, I_{lt}) = \sum_{k=1}^{d} tf(w, I^k_{st}) \cdot e^{-\rho \cdot k}, \tag{10}$$

where $e^{-\rho \cdot k}$ is a forgetting factor supporting the assumption that the user's profiles gradually decay as time passes and ρ is the forgetting coefficient. d is the number of days accumulated in the long-term profiles and I^k_{st} represents the kth short-term profile.

Then, the probability $P(w \mid \theta_{i-lt})$ for the long-term profiles is defined as follows:

$$P(w \mid \theta_{i-lt}) = \frac{tf(w, I_{lt})}{\sum_{w' \in I_{lt}} tf(w', I_{lt})}$$

$$= \frac{\sum_{k=1}^{d} tf(w, I_{st}^k) \cdot e^{-\rho \cdot k}}{\sum_{w' \epsilon I_{it}} \sum_{k=1}^{d} tf(w', I_{st}^k) \cdot e^{-\rho \cdot k}}, \tag{11}$$

where w' is any word in the long-term profile I_{lt}, which is accumulated from d days' short-term profiles.

Finally, the parameters θ_i of an individual model are calculated by interpolating the two models $P(w \mid \theta_{i-st})$ and $P(w \mid \theta_{i-lt})$:

$$P(w \mid \theta_i) = \beta P(w \mid \theta_{i-st}) + (1 - \beta) P(w \mid \theta_{i-lt}), \tag{12}$$

where β is a parameter used for smoothing.

Similarly, for different smoothing methods [Zhai and Lafferty 2004], β takes different forms. For example, β takes fixed value for Jelinek-Mercer smoothing. For Bayesian smoothing with the Dirichlet prior, β takes the form:

$$\beta = \frac{|I_{st}|}{|I_{st}| + \nu}, \tag{13}$$

where the free parameter ν is the Dirichlet smoothing parameter to control the degree to which the user's statistics are altered by the overall training database statistics. $|I_{st}|$ is the number of the terms in the short-term profiles.

We denote the parameters of the global model as θ_g. The global model $P(w \mid \theta_g)$ can be calculated based on the average of all users' information. That is, the average probability of a term w is the sum of all the probabilities across all users in which the term occurs, divided by the number of users in which the term occurs, which is represented as:

$$P(w \mid \theta_g) = \frac{\sum_{i' \in g} P(w \mid \theta_{i'})}{|g|}, \tag{14}$$

where $|g|$ represents the number of the users in g and i' represents any user in g.

Analogously, we denote the parameters of the group model as θ_c. The group model $P(w \mid \theta_c)$ can be calculated based on the average of all the users' information in group c. That is, the average probability of a term w is the sum of all the probabilities across all users belonging to group c in which the term occurs, divided by the number of users in which the term occurs, which is represented as:

$$P(w \mid \theta_c) = \frac{\sum_{i' \in c} P(w \mid \theta_{i'})}{|c|}, \tag{15}$$

where $|c|$ represents the number of users in c and i' represents any user in c.

4.2 Smoothing by Global User Model

An important issue is to obtain a stable model for user modeling. In the past, Ponte and Croft [1998] applied corpus data to stabilize a document model. In

this method, when a term does not appear in a document, the probability of the missing term can be smoothed by the probability of the term in the corpus. According to Zhai and Lafferty [2004], smoothing refers to the adjustment of the maximum likelihood estimator of a language model so that it can be more accurate. At the very least, it prevents the assignment of zero probability to unseen words. Additionally, as mentioned in Zhai and Lafferty [2004], when estimating a language model based on a limited amount of text, such as a single document, smoothing of the maximum likelihood model is considered very important.

Analogously, the information on individual users is also very sparse. If an individual's information is solely relied upon to construct personal preferences, the two major problems, including the data sparseness and cold-start for new users, will degrade the personalization performance. To solve these two problems, statistical information gathered from the global users can be considered as the global corpus and used for smoothing.

More specifically, we apply a global model to smooth the individual's model. The parameters of the global-based smoothing model is denoted as θ_{i+g}. Then the model $P(w \mid \theta_{i+g})$ can be calculated as:

$$\begin{aligned} P(w \mid \theta_{i+g}) &= \gamma P(w \mid \theta_i) + (1-\gamma) P(w \mid \theta_g) \\ &= \gamma P(w \mid \theta_i) + (1-\gamma) \frac{\sum_{i' \in g} P(w \mid i')}{|g|}, \end{aligned} \tag{16}$$

where $P(w \mid \theta_i)$ is the estimate of maximum likelihood that user i likes word w, and $P(w \mid \theta_g)$ is the estimate of maximum likelihood for that word rating among the set of all users g based on Equation (14).

Similarly, for different smoothing methods, γ (a general symbol for smoothing) takes different forms. For Jelinek-Mercer smoothing, γ takes a fixed value. For Bayesian smoothing with the Dirichlet prior, γ takes the form:

$$\gamma = \frac{|I|}{|I| + o}, \tag{17}$$

where the free parameter o is the Dirichlet smoothing parameter and $|I|$ is the number of the terms in the user's preferences.

According to the model, if a user is a newcomer to the system, $P(w \mid \theta_i)$ will be zero. Equation (16) is reformulated as:

$$\begin{aligned} P(w \mid \theta_{i+g}) &= \gamma P(w \mid \theta_i) + (1-\gamma) P(w \mid \theta_g) \\ &= (1-\gamma) P(w \mid \theta_g). \end{aligned} \tag{18}$$

As shown in Equation (18), a personalized system can rely on the behaviors of global users, which can be described as the DirectHit (http://www.directHit.com) algorithm in the content version instead of the voting version.

4.3 Smoothing by Group User Model

Using the behaviors of global users as the smoothing data, we can alleviate the cold-start problem to some extent. In addition, the sparseness problem is

also somewhat smoothed. However, the above model may be too general in that all users' common preferences are reflected heavily in the model. To address the problem, we propose to use a cluster-based language model for smoothing through similar users' preferences.

A cluster-based model requires that users are first grouped into clusters. Assume that users are clustered into K groups $C = \{c_1, c_2, \ldots, c_K\}$. Each user $u \in U$ belongs to one cluster only. In the following, we first present a cluster-based language model. An approach similar to the individual smoothing model is taken to build the language model for clusters. The cluster-based language model θ_{c+g} smoothes the representations of clusters θ_c using models of the global users model θ_g. Generally, based on Equations (14) and (15), the model $P(w \mid \theta_{c+g})$ can be calculated as:

$$\begin{aligned} P(w \mid \theta_{c+g}) &= \eta P(w \mid \theta_c) + (1-\eta) P(w \mid \theta_g) \\ &= \eta \frac{\sum_{i' \in c} P(w \mid i')}{|c|} + (1-\eta) \frac{\sum_{i' \in g} P(w \mid i')}{|g|}, \end{aligned} \tag{19}$$

where η is a general symbol for the smoothing methods used. Similarly, for Bayesian smoothing with the Dirichlet prior, η takes the form:

$$\eta = \frac{|c|}{|c| + \tau}, \tag{20}$$

where the free parameter τ is the Dirichlet smoothing parameter and $|c|$ is the number of the terms in the group's preferences.

Thus, each cluster is estimated by the language model and the cluster is then taken as a group of like-minded users for personalization.

Finally, the cluster-based language model is used to smooth the individual's model according to the cluster they belong to. In particular, our model θ_{i+c+g} is formulated as

$$\begin{aligned} P(w \mid \theta_{i+c+g}) &= \gamma P(w \mid \theta_i) + (1-\gamma) P(w \mid \theta_{c+g}) \\ &= \gamma P(w \mid \theta_i) + (1-\gamma)[\eta P(w \mid \theta_c) + (1-\eta) P(w \mid \theta_g)], \end{aligned} \tag{21}$$

where c represents the cluster i belongs to, while γ and η are general symbols for smoothing. The user's cluster model is first smoothed with the global user model, and then the individual user model is smoothed using the smoothed cluster model.

Here we propose two different criteria to select a cluster c according to the sensitivity of the cluster to the query: a query-dependent cluster selection method and a query-independent cluster selection method. Given a user u and a query q, query-dependent cluster selection is based on the similarity of cluster c to the user and to the query, which is calculated by:

$$\begin{aligned} c &= \arg\min_{c' \in C} (\zeta D(\theta_i \| \theta_{c'}) + (1-\zeta) D(\theta_q \| \theta_{c'})) \\ &= \arg\min_{c' \in C} \left(\zeta \sum_{w \in i} P(w \mid \theta_i) \log \frac{P(w \mid \theta_i)}{P(w \mid \theta_{c'})} + (1-\zeta) \sum_{w \in q} P(w \mid \theta_q) \log \frac{P(w \mid \theta_q)}{P(w \mid \theta_{c'})} \right), \end{aligned} \tag{22}$$

Table I. Different Models under the Unified Framework

Interpolating	Model	Parameters	Description
Query only	q	$\lambda = 1$	General Web search
Query + Individual	$q+i$	$1 > \lambda > 0, \gamma = 1$	Personalized Web search
Query + Group	$q+c$	$1 > \lambda > 0, \gamma = 0, \eta = 1$	Collaborative Web search
Query + Global	$q+g$	$1 > \lambda > 0, \gamma = 0, \eta = 0$	Content based DirectHiT algorithm
Query + Individual + Global	$q+l+g$	$1 > \lambda > 0, 1 > \gamma > 0, \eta = 0$	Personalized Web search using global smoothing
Query + Individual + Group	$q+i+c$	$1 > \lambda > 0, 1 > \gamma > 0, \eta = 1$	Personalized Web search using group smoothing
Query + Individual + Group + Global	$q+i+c+g$	$1 > \lambda > 0, 1 > \gamma > 0, 1 > \eta > 0$	Personalized Web search using group and global smoothing

where ζ is the interpolating parameter to leverage the weight between user and query.

4.4 Clustering Algorithms for Users

There are many algorithms that can be used to generate user clusters. In our work, the K-means algorithm [Hartigan and Wong 1979] is selected as the clustering algorithm, which is by far one of the most popular clustering tools used in scientific and industrial applications. The cluster number K is an input of the algorithm that specifies the desired cluster number. The cosine measure is taken as the similarity measure function. The clustering is constructed in several phases. In the first phase, the algorithm takes the first K users as the centroids of K nonoverlapping clusters, which represent each of the K clusters by the mean (or weighted average) of its data points. Each of the remaining users is then compared with the closest centroid. In the next phase, the cluster centroids are re-computed based on cluster centroids that are formed in the previous phase. The cluster-membership is then reevaluated.

4.5 A Unified Framework for User Language Model

The modeling method in Equation (21) can also be described as a two-stage smoothing method for a user language model in which first a cluster model is smoothed with a global model and then, the smoothed cluster model further be used to smooth an individual model. The final model can be viewed as a mixture of three separate models: the individual model, the group model, and the global model.

By considering each model separately, these models can be combined to make seven different models as shown in Table I. Different models use different interpolating parameters under this framework. Many existing personalization methods, such as personalized search and collaborative search, can fall under this framework.

For example, the model $q+i$ is the model for traditional personalized search by only utilizing the personal search behaviors, which is relevant to Teevan et al. [2005]. While the model $q+c$ is the model to personalization by using

the group user's behavior, which is the core idea of collaborative Web search [Smyth et al. 2003]. The model $q + g$ can be viewed as the reformulation of the DirectHit approach by using the global user's behaviors for personalization. The model $q + i + c$ is the model for smoothing the individual model with the cluster model, which is similar to the method described in Sugiyama et al. [2004], but with an explanation through the language modeling approach to information retrieval. The model $q + i + c + g$ uses like-minded users and global users to smooth the behavior of an individual, which is defined as collaborative personalized search. We compare the performance of these different models in the experiment section.

Finally, the user language model $P(w|\theta_u)$ can be combined with the query model $P(w|\theta_q)$ to enable a personalized search. Here θ_u can be θ_i, θ_c, θ_g, θ_{i+c}, θ_{i+g}, or θ_{i+c+g}.

5. EXPERIMENTS

In this section, we first present the experimental hypotheses and the dataset used in this paper. We conduct data analysis on these data. Then we describe the evaluation metric. Finally, we present the performances of different models.

5.1 Experimental Hypothesis

Our collaborative personalized search algorithm integrates three models: an individual user model, a group user model and a global user model. We denote this integrated model as $i + c + g$, which denotes individual (i) + user group (c) + global (g) models. In this section, we empirically evaluate our proposed algorithm in terms of the following hypothesis:

—Hypothesis H1: The collaborative personalized search algorithm that integrates all three models, that is, $i + c + g$, will perform better than personalized systems that do not use all three models. In our experiments, we will test the hypothesis on some concrete datasets.

At the beginning of the paper, we outlined two difficulties of personalized search: the data-sparseness problem and the cold-start problem. We further split the above hypothesis into two following subhypotheses to complete the evaluation:

—Hypothesis H2: When the personal data is few and sparse, the collaborative personalized search algorithm $i + c + g$ consistently performs better than other personalized algorithms that do not consider all three models.

—Hypothesis H3: The collaborative personalized search algorithm can still achieve better performance for the new users who suffer from the cold-start problem.

5.2 Dataset

The dataset used in this paper is collected from MSN Web search engine.[6] The full dataset comprises one month's accesses to the engine, collected from

[6]http://search.msn.com.

September 28 to October 27, 2005. There exist many different types of search results, including HTML, PDF, PPT, and other multimedia. In this research, we focus on the Web page search, so other types of search are filtered. Furthermore, the queries are in different languages, such as English, French, and Chinese. Here we only deal with English queries and remove the clickthrough data in other languages.

The search engine identifies the users by a unique id (GUID) for each user. We randomly selected 1,000 users from all these that made at least 100 queries within the four weeks as the whole user set. Ultimately, 263,665 requests have been recorded from these users. On average, there are 263 requests for each user. Each request consists of four parts: <GUID, Date, Query, URL>. Among these requests, there are 57,561 unique queries and 203,933 unique pages. On average, four pages are clicked by the users for each query. We will use the data to build user profile and to evaluate the performance of the personalized search.

In the clickthrough data, a Web page is associated with a query submitted by a particular user. This ground truth data can be taken as the correctness judgment for the personalized search in this experiment. Based on users' previous profiles, we can use the data to measure the ranking performance of the personalization system by resubmitting the query. Hence, this dataset can be used for validation.

5.2.1 *Data Analysis.* We performed a statistical analysis to show the similar behaviors among different users. Results show that only 11.3% of all the queries are submitted by different users while only 9.6% of all the pages are browsed by different users. Without considering content features of the queries and the pages, the users share very small fraction with each other. The data are too sparse to perform collaborative personalized search. For the content analysis, Porter stemming [Frakes 1992] is applied and the stop-words are removed for the content features of the queries and the Web pages. In the end, 1,693,596 terms are culled. Statistics show that the average percentage of similar terms among different users is about 44.3%. This demonstrates that applying a collaborative personalized search based on content analysis is feasible.

5.2.2 *Training and Testing Data.* In order to evaluate the performance of the proposed model, 100 users are randomly selected from the whole 1,000 users for training and testing. 328 queries submitted by the 100 users on the last day are collected to perform cross-validation experiments.

We split the 100 users into two groups: 70 in one group and 30 in the other. The queries submitted by 70 users are further split into two parts of equal size: tuning queries and testing queries. We tune the parameters on the tuning queries, which are defined as *Train70*, and take testing queries as *Test70*. The queries of the other 30 users are taken as the second testing data, named *Test30*. Details of tuning and testing data are shown in Table II. Hence, the user set of Test70 is the same as Train70 but with different queries, while the users in Test30 are different from those in Train70. We conduct the experiments on these two testing data sets to show the performance of the proposed algorithms.

Table II. Tuning Set and Testing Set

Label	Description	Number of Users	Number of Queries
Train70	Training Data	70	104
Test70	Testing Data I		105
Test30	Testing Data II	30	119
	Total	100	328

5.3 Evaluation Metric

To measure the ranking quality, the Discounted Cumulative Gain (DCG) [Järvelin and Kekäläinen 2000] is used. DCG is a measure that gives more weight to highly ranked documents and incorporates different relevance levels (highly relevant, relevant, and irrelevant) through different gain values.

$$DCG(i) = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + \dfrac{G(i)}{\log(i)} & \text{otherwise.} \end{cases} \tag{23}$$

In this work, the relevance level is acquired from the query log data. We use $G(i) = 1$ for clicked search results and $G(i) = 0$ for results that were not clicked. The average normalized DCG (NDCG) over all queries is selected to show the performance. In this work, NDCG N ($N = 1, 5, 10, 20, 30$) is used for evaluation, N is the number of top-N documents returned by the ranking algorithms.

5.4 Experiment Design

Since our work is focused on investigating the performance of personalized search for different individuals, this can be done by the reranking of search results from a general search engine. Hence, for each query, 200 search results are extracted from the MSN Web search engine[7] for personalized re-ranking. 65,600 Web pages are returned on 328 queries. Since our algorithm is based on the language model, these Web pages are indexed using a language model. A search engine is set up for these Web pages while the basic retrieval system uses a KL-Divergence model with Bayesian prior smoothing [Zhai and Lafferty 2001]. The Lemur retrieval system[8] is selected as the indexing and ranking system.

The major goal of this study is to examine how the behaviors of individuals, group users and global users can help improve the search performance in terms of NDCG. In particular, these behaviors can provide extra information to help estimate a better query model than using the query alone. Hence, our experiments involve comparing the retrieval performance through single queries as well as different user models in Table I.

Six different personalized models described in Table I are compared: individual model (i), group model (c), global model (g), individual plus group model ($i + c$), individual plus global model ($i + g$), individual plus group and global model ($i + c + g$). Since each model has different parameters, we manually tune

[7]http://search.msn.com.
[8]http://www.lemurproject.org.

Table III. Performance on Test70 Using Different Personalized Schemas for Web Pages Ranking

Model	NDCG1	NDCG5	NDCG10	NDCG20	NDCG30
q	0.422	0.434	0.441	0.416	0.384
$q+i$	0.664	0.655	0.613	0.535	0.467
$q+c$	0.724	0.674	0.635	0.515	0.438
$q+g$	0.672	0.667	0.626	0.546	0.497
$q+i+g$	0.707	0.674	0.641	0.556	0.474
$q+i+c$	0.712	0.675	0.64	**0.557**	0.474
$q+i+c+g$	**0.724**	**0.683**	**0.644**	0.555	**0.499**

Table IV. Performance on Test30 Using Different Personalized Schemas for Web Pages Ranking

Model	NDCG1	NDCG5	NDCG10	NDCG20	NDCG30
q	0.462	0.416	0.409	0.391	0.375
$q+i$	0.622	0.619	0.587	0.513	0.463
$q+c$	0.663	0.621	0.573	0.509	0.451
$q+g$	0.654	0.6	0.562	0.515	0.45
$q+i+g$	0.663	0.617	0.577	0.507	0.451
$q+i+c$	0.663	0.621	0.582	0.509	0.448
$q+i+c+g$	**0.673**	**0.625**	**0.592**	**0.52**	**0.472**

them for each model on the Train70 dataset as shown in Table II. The details for parameter selection are described in Section 5.6. These parameters are determined for the optimal performance level for each model. Generally, for the model $(i+c+g)$, Dirichlet interpolating is used to integrate the query model and the user model, while μ in Equation (8) is set to 5. Jelinek-Mercer method is taken as the interpolating method for combining different user models. β in Equation (12), γ, and η in Equation (21) are set to 0.7, 0.6, and 0.5, respectively. ρ is set to 0.4, 1,000 users are clustered into 20 clusters, and query-independent cluster selection is applied.

5.5 Experimental Results

5.5.1 *Overall Performance of Personalization.* The performance of different models is compared on two testing query datasets: Test70 and Test30. The results are shown in Table III and IV. The row labeled with q is the baseline performance and another row with i, c, or g is the performance of using the different user models. Several observations can be made from these two tables:

1. On average, personalization by different profiling schemas performs better than a query only on the two test sets. As an example of the measurement of NDCG5, personalization using the information from individuals can achieve 50.9% and 48.8% improvement over the baseline method on Test70 and Test30, respectively. The t-test is performed and the result shows that the improvement is significant (P-value <0.0001). The result supports that personalization can improve the search performance.
2. As shown in Tables III and IV, by interpolating the behaviors of individuals with that of group and global users $(q+i+c+g)$, the personalization

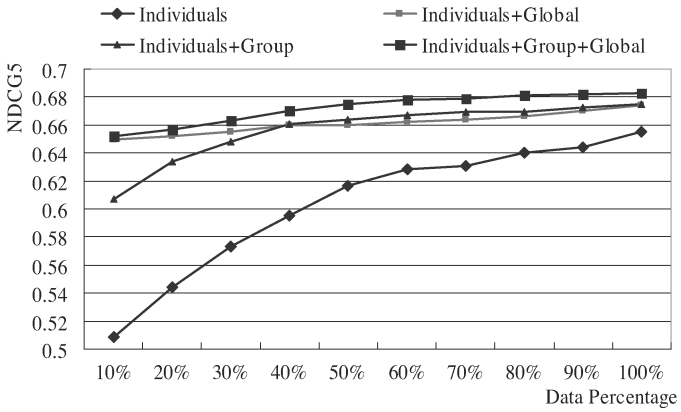

Fig. 3. Performance on using different sparse data on Test70.

can achieve the highest performance on the two test sets. In particular, it can achieve 57.4% and 50.2% improvement over a query only method, with the measure of NDCG5 on Test70 and Test30, respectively. The t-test is also performed and the result shows that the improvement is significant (P-value <0.0001). Furthermore, the $(q + i + c + g)$ model can achieve a higher performance than the other models on personalized search, which supports hypothesis **H1**. For example, a collaborative personalized model involving the global user model and the group user model can achieve 8.2% and 9.1% improvement over the personalized method based on the pure individual model, on the two different datasets by the measurement of NDCG1 (P-value=0.0351).

3. Experimental results also indicate that using behaviors of global users or group users for personalization can improve the search performance. As shown in Table III, using NDCG5 as a measure, the two methods in row $q+c$ and $q + g$ can achieve 55.3% and 53.7% improvement over query alone on Test70 queries, respectively. These two methods can also achieve 49.3% and 44.2% improvement over query alone on Test30 queries with NDCG5, which are shown in Table IV (P-value <0.0001). The results also show that using the information from group users can slightly achieve better performance than that of global users.

In the next two sections, the performance of using the global model with little personal data is shown to achieve better performance. This is designed to test the algorithm on its ability to handle the cold-start problem.

5.5.2 *Performance and Sparseness.* In order to demonstrate the performance of our proposed algorithm on sparse personal data, which corresponds to our hypothesis H2, experiments with different personalized methods are also conducted on sparse datasets of different degrees of sparseness.

In this experiment, we extract 10%, 20%, ..., 100% searching historical data for users in Test70 to represent the different degrees of data sparseness. Figure 3 empirically demonstrates how performance changes on the same test

Table V. Performance on the Cold-Start Problem

	Model	NDCG1	NDCG5	NDCG10	NDCG20	NDCG30
Test70	q	0.422	0.434	0.441	0.416	0.384
	$q+g$	0.629	0.65	0.58	0.519	0.474
Test30	q	0.462	0.416	0.409	0.391	0.375
	$q+g$	0.644	0.587	0.544	0.487	0.443

data by the measurement of NDCG5 when the personal data is changed from sparse to dense.

The results in Figure 3 show that different sparse levels of personal data can have a significant impact on the performance of most methods. The sparser the personal data the lower the achievable performance, as shown in Figure 3, the model based on individuals has the lowest performance on just 10% data. As the data increased, the performance for the individual model is obviously improved. The other three models produced consistently higher performances over the model based on only individual profiles.

When the training data only uses 10% of the search history data, using global behaviors such as smoothing is more accurate than the group model. This shows that global smoothing works better than the group models on very sparse data. As data increases, the performance of the group model increases quickly and achieves better performance than the global model after using 40% of data.

As shown in Figure 3, when the training data are increased from 10% to 60%, the performance of the model $(i + c + g)$ increased consistently. Then, the improvement of the model will saturate when the training data consists of more than 60% of the whole data as the performance is already close to 100%. Generally, the model $(i + c + g)$ can achieve a better performance than the other three models. From this evaluation with the increase in data, we can support **H2**. By comparison, the proposed smoothing models not only show greater precision, they perform more stably.

5.5.3 *Performance on Cold-Start.* As mentioned in Section 4.2, global users' behavior can help alleviate cold-start problems. Since a newcomer does not have any historical data, smoothing based on individual information and group users would be impossible, therefore the model $(i+c+g)$ will degrade to the model g. In our experimental setting for using only the behaviors of global users for personalization, the search histories of users in Test70 and Test30 are removed from the whole training data set. Thus, the search system cannot provide any personalization for the absence of personal information. In such a situation, our proposed model can still make a personalization by using the global information.

The experimental results are shown in Table V. The rows labeled $(q + g)$ are the results of using the behaviors of global users to solve the cold-start problem for new users. As shown in Table V, by directly applying the behaviors of global users, a performance higher than just using query only is achieved. On average, there is 49.7% and 41.1% improvement on NDCG5 over the query alone model on Test70 and Test30 datasets, which verifies that utilizing the global users' behaviors, can alleviate the cold-start problem. The t-test is performed and the

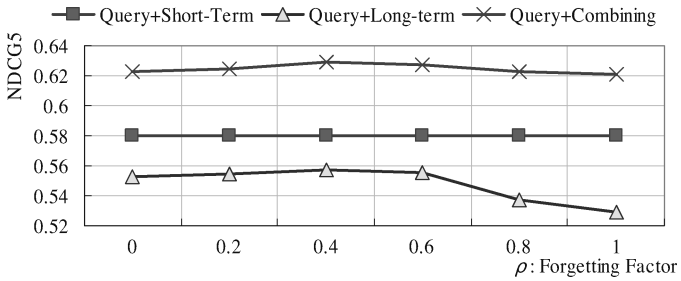

Fig. 4. Performance on different term profiles.

result shows that the improvement is significant (P-value <0.0001). Thus, we can support the hypothesis **H3**.

5.6 Parameter Tuning

A major advantage of our model is its flexibility in accommodating a wide range of parameter tuning in order to optimize the performance. In this section, experiments are conducted to tune the parameters for the different user profiles, the interpolating methods, and the clustering parameters. The experiments are performed by using the measurement of NDCG5. The query set used for this evaluation is Train70. In this paper, the other parameters are fixed when tuning one parameter.

5.6.1 *Individual Model Construction and Forgetting Factor Tuning.* As described in many personalized systems, an individual's model can be constructed through long-term and short-term preferences. Here the experiments are conducted to show the performance of personalization on the long-term, the short-term, and the combined models (interpolating between the short-term and the long-term preferences using Jelinek-Mercer smoothing by setting β with 0.7).

As shown in Figure 4, using only the short-term preferences can achieve better performance than using only the long-term preferences. This proves that current search behaviors of individuals are associated with their short-term behaviors. Furthermore, by interpolating the short-term and the long-term preferences, the best performance is achieved.

As shown in Equation (10), a forgetting coefficient ρ is assigned when the long-term preferences are constructed. The value of ρ varies from 0 to 1. When setting ρ to 0, the algorithm does not change the weight of the earlier data. When setting ρ to 1, the algorithm quickly decreases the weight of earlier data. The value ρ is tuned to obtain the best performance on personalization. As shown in Figure 4, the algorithm can achieve the best performance when ρ is set to 0.4. When ρ is higher than 0.4, which means that the earlier data will quickly be forgotten, the long-term preferences lose some valuable information for constructing the long-term preferences. When ρ is less than 0.4, which means the earlier data will dominate the long-term preferences, the performance for the current query is affected. In this paper, ρ is set to 0.4.

Table VI. Performance by the Two Interpolating Methods on Query Model and User Model

	Jeninek-Mercer	Dirichlet
q	0.373	0.373
$q+i$	0.603 ($\lambda = 0.8$)	0.627 ($\mu = 5$)

Table VII. Performance by the Two Interpolating Methods

	Jeninek-Mercer	Dirichlet
$i_{st} + i_{lt}$	0.627 ($\beta = 0.7$)	0.626 ($v = 200$)
$i+g$	0.648 ($\gamma = 0.6$)	0.646 ($o = 1700$)
$i+c$	0.650 ($\gamma = 0.5$)	0.648 ($\tau = 2100$)
$i+c+g$	0.656 ($\eta = 0.6$ $\gamma = 0.5$)	0.654 ($o = 1700$ $\tau = 2100$)

5.6.2 *Interpolating Parameters Selection.* As shown in Equations (7), (12), (16), (19), and (21), there are two interpolating methods: Jelinek-Mercer-based interpolating and Dirichlet-based interpolating. In order to measure which interpolating method gives better results, experiments are conducted to show the effects of different parameters.

The first experiment aims to show the performance on different interpolating methods between the query and user models. We tune λ in Equation (7) from 0 to 1 for Jelinek-Mercer smoothing and tune μ in Equation (8) for Dirichlet smoothing. The results on NDCG5 are shown in Table VI on tuning each parameter to get the best performance.

As shown in Table VI, the Dirichlet interpolating method between the query and user models returns better results. If the query is very long, which means that the user has provided enough information for the search engine, the query should be more trustworthy. When the query comprises only one word, it may be beneficial to put more weight on the user profile information.

The second experiment is conducted to show the performance on different interpolating methods between long-term and short-term, and among individuals, group users and global users. Each parameter is manually tuned for each of the two different interpolating methods. The parameters setting and the performance for two interpolating methods are shown in Table VII.

Table VII shows that the Jelinek-Mercer interpolating method achieves a slightly better performance. The t-test is performed on the two interpolating methods and the result shows that the improvement is significant (p-value = 0.0299). Different methods of interpolating for user profiles have little impact on the performance of personalization when user profiles are large enough. Moreover, according to the experimental results from Table VI and Table VII, we can find that Dirichlet interpolating method can perform better in case of varying length documents while Jelinek-Mercer interpolating method will achieve slightly better for the long profile. Additionally, since the Jelinek-Mercer interpolating method uses a fixed parameter for all users, it can easily be realized.

5.6.3 *Cluster Selection by Query-Independent or Query-Dependent.* As described in Section 4.3, two approaches exist for selecting the cluster to

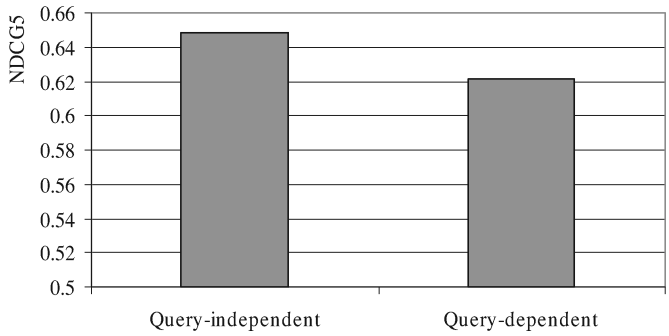

Fig. 5. Performance on query sensitive cluster selection.

Table VIII. Performance on Different Cluster Number

Cluster Number	1	20	40	60	80	100	200	400	1000
NDCG 5	0.646	**0.654**	0.652	0.644	0.637	0.64	0.635	0.623	0.623

interpolate with individuals during the search time: query-independent cluster selection and query-dependent cluster selection. For the query-dependent cluster selection method, which is shown in Equation (22), the cluster the user belongs to is selected according to two criteria. First, the cluster should be nearest to the individual user. Secondly, the cluster should be most relevant to the query. Accordingly, the parameter ζ is tuned to achieve the best performance. For the method of query-independent cluster selection, the cluster the user belongs to is selected only by the nearest cluster to the individual.

As shown in Figure 5, the query-independent selection method can achieve better performance than query-dependent selection method since query-dependent cluster selection may sometimes cause the user to be combined with a cluster that he is not similar to. The situation can be found that the cluster contains the query term while the user does not belong to the cluster. Additionally, query-independent cluster selection can be done offline to improve the efficiency of the personalization system. Therefore, query-independent cluster selection is adopted in this article.

5.6.4 *Selection of Clustering Number.* In order to select a suitable number of clusters, the Train70 dataset is used for tuning. The users are clustered into different K groups based on 9 values of K (1, 20, 40, 60, 80, 100, 200, 400, and 1000). Then personalization is performed on each group of clusters. As shown in Table VIII, the best performance on NDCG5 measurement is achieved for 20 clusters in Train70. Furthermore, the number of clusters should be selected properly although it has a little effect on the performance of our algorithms. The cluster number K is set to 20 in this work.

6. CONCLUSION AND FUTURE WORK

We observe that many commonalities can be found in the behaviors of an individual user and other users. As a result, in this paper, we proposed to use a statistical user language model to integrate the behaviors of individual users,

group users, and global uses. We have shown that our methods are effective in dealing with the data sparseness and the cold-start problems for personalized search. More specifically, the probability that a user likes a Web page is calculated through two-step smoothing mechanisms. In the first step, a global user model is used to smooth the probability of unseen terms in the individual profiles. In order to precisely describe the user's interests, the information derived from the similar user behaviors is utilized to enhance the individual model. Users are clustered into groups and group user models are constructed. In the second step, the group models are integrated into the whole model through a cluster-based language model. Based on the statistical user language model, we proposed a collaborative personalized search approach to improve the performance of general Web search and personalized search. Experimental results show that our proposed model can achieve significant improvements over the general search system with a personalized subsystem that is solely based on individual information. We have shown that the sparseness and cold-start problems can be alleviated by using data from the group and global users' behaviors.

Our work can be extended in several ways. First, only the query content and Web page content is used when we construct the user individual model. There is much information that can still be included, such as the category information of the Web pages and queries. Such information can also be integrated into our model for user modeling. Second, our work has focused on Web search only, but the user language model can be applied to many other areas, such as news filtering, email filtering, and online shopping. In this study, we are only concerned with a general user language model. The number of clusters and the interpolating parameters could be better measured by more experiments. Further research needs to be done on the general framework for studying optimization methods for parameters in personalized search models.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable and constructive comments.

REFERENCES

AGICHTEIN, E., BRILL, E., DUMAIS, S. T., AND RAGNO, R. 2006. Learning user interaction models for predicting Web search preferences. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, 3–10.

ALLAN, J., CARBONELL, J., DODDINGTON, G., YAMRON, J., AND YANG, Y. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the ARPA Broadcast News Transcription and Understanding Workshop*, 194–218.

ALMEIDA, R. B. AND ALMEIDA, V. A. 2004. A community-aware search engine. In *Proceedings of the 13th International Conference on World Wide Web*. ACM Press, New York, NY, 413–421.

ANAND, S. S. AND MOBASHER, B. 2005. Intelligent techniques for web personalization. In *Intelligent Techniques for Web Personalization*. Bamshad Mobasher and Sarabjot Singh Anand (Eds), Lecture Notes in Artificial Intelligence (3169), Springer, 1–37.

ANICK, P. 2004. Using terminological feedback for Web search refinement: A log-based study. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM Press, New York, NY, 88–95.

BAI, J., NEI, J. Y., CAO, G. H., AND BOUCHARD, H. 2007. Using query contexts in information retrieval. In *Proceedings of the 30th Annual international ACM SIGIR Conference on Research*

and Development in information Retrieval, Amsterdam, The Netherlands. ACM, New York, NY, 15–22.

BEAULIEU, M., THIEN, D., PAYNE, A., AND JONES, S. 1996. Enquire okapi project. *British Library Resear. Innov. Rep. 17*, 103.

BELKIN, N., COOL, C., KOENEMANN, J., NG, K. B., AND PARK, S. 1996. Using relevance feedback and ranking in interactive searching. In *Proceedings of 4th Text Retrieval Conference* (TREC-4). Harman, D.K., ed. NIST, 181–188.

BHARAT, K. AND KAMBA, T. 1995. An interactive personalized newspaper on the WWW. In *Proceedings of the 4th International Conference on World Wide Web*. ACM Press, New York, NY, 159–170.

BILLSUS, D., AND PAZZANI, M. 1999. A hybrid user model for news story classification. In *Proceedings of 7th International Conference on User Modeling*. 99–108.

BREESE, J. S., HECKERMAN, D., AND KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. 43–52.

BROWN, P. F., DELLAPIETRA, V. J., DESOUZA, P. V., LAI, J. C., AND MERCER, R. L. 1992. Class-based N-gram models of natural language. *Computat. Linguis. 18*, 4, 468–479.

BURGES, C., SHAKED, T., RENSHAW, E., LAZIER, A., DEEDS, M., HAMILTON, N., AND HULLENDER, G. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*. 89–96.

CHEN, S. AND GOODMAN, J. 1998. An empirical study of smoothing techniques for language modeling. Tech. Rep. *TR-10-98*, Computer Science Group, Harvard University.

CHIRITA, P., NEJDL, W., PAIU, R., AND KOHLSHUETTER, C. 2005. Using ODP metadata to personalize search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 178–185.

CLAYPOOL, M., LE, P., WASEDA, M., AND BROWN, D. 2001. Implicit interest indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces* (IUI2001), 33–34.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. 39*, 1, Series B, 1–38.

DUMAIS, S. T., CUTRELL, E., CADIZ, J. J., JANCKE, G., SARIN, R., AND ROBBINS, D. C. 2003. Stuff I've seen: A system for personal information retrieval and re-use. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, 72–79.

FISHER, D., HILDRUM, K., HONG, J., NEWMAN, M., THOMAS, M., AND VUDUC, R. 2000. SWAMI: A framework for collaborative filtering algorithm development and evaluation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 366–368.

FOX, S., KARNAWAT, K., MYDLAND, M., DUMAIS, S., AND WHITE, T. 2005. Evaluating implicit measures to improve web search. *ACM Trans. Inform. Syst. 23*, 2, 147–168.

FRAKES, W. B. 1992. Stemming Algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*, Chapter 8, Prentice-Hall, 131–160.

GOLDBERG, K., ROEDER, T., GUPTA, D., AND PERKINS, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Inform. Retriev. 4*, 2, 133–151.

HARTIGAN, J. A. AND WONG, M. A. 1979. A K-means clustering algorithm. *Appl. Statist. 28*, 100–108.

HAVELIWALA, T. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, ACM Press, 517–526.

HOFMANN T. AND PUZICHA J. 1998. Statistical models for co-occurrence data. Tech. Rep. AI Memo 1625, Artificial Intelligence Laboratory, MIT, February 1998.

JÄRVELIN, K. AND KEKÄLÄINEN, J. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, 41–48.

JIN, R., HAUPTMANN, A., AND ZHAI, C. 2002. Title language model for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, 42–48.

JOACHIMS, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press. 133–142.

KELLY, D. AND TEEVAN, J. 2003. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum 37*, 2, 18–28.

KERSCHBERG, L., KIM, W., AND SCIME, A. 2001. WebSifer: Personalizable meta-search agent based on semantic weighted taxonomy tree. In *Proceedings of the International Conference on Internet Computing*. 14–20.

KOENMANN, J. AND BELKIN, N. 1996. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground*. M. J. Tauber, Ed. ACM Press, 205–212.

KONSTAN, J. A., MILLER, B. N., MALTZ, D., HERLOCKER, J. L., GORDON, L. R., AND RIEDL, J. 1997. GroupLens: Applying collaborative filtering to usenet news. *Comm. ACM 40*, 3, 77–87.

KRITIKOPOULOS, A. AND SIDERI, M. 2003. The compass filter: Search engine results personalization using web communities. In *Proceedings of the Workshop on Intelligent Techniques for Web Personalization* (ITWP'03), 229–240.

KRULWICH, B. 1997. Lifestyle finder—Intelligent user profiling using large-scale demographic data. *AI Mag. 18*, 2, 37–45.

LAFFERTY, J. AND ZHAI, C. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 111–119.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 120–127.

LI, X. AND CROFT, W. B. 2003. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. ACM Press, New York, NY, 469–475.

LIEBERMAN, H. 1995. Letizia: An agent that assists Web browsing. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. 924–929.

LIU, F., YU, C., AND MENG, W. 2002. Personalized Web search by mapping user queries to categories. In *Proceedings of the 11th international Conference on information and Knowledge Management*, ACM Press, New York, NY, 558–565.

LIU, X. AND CROFT, W. B. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 186–193.

MANBER, U., PATEL, A., AND ROBISON, J. 2002. Experience with personalization on Yahoo! *Comm. ACM 43*, 8, 35–39.

MCLACHLAN, G. J. AND BASFORD, K. E. 1988. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

MITCHELL, T. M. 1997. *Machine Learning*, McGraw Hill, 1997.

MLADENIC, D. 1996. Personal Webwatcher: Design and implementation, Tech. Rep. ijs-dp-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh.

NTOULAS, A., CHO, J., AND OLSTON, C. 2004. What's new on the Web? The evolution of the Web from a search engine perspective. In *Proceedings of the 15th International Conference on World Wide Web*, ACM Press, New York, NY, 233–241.

PAZZANI, M. 1999. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Revi. 13*, 5, 393–408.

PAZZANI, M., MURAMATSU, J., AND BILLSUS, D. 1996. Syskill & Webert: Identifying interesting Web sites. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 54–61.

PITKOW, I., SCHUTZE, H., CASS, T., COOLEY, R., TURNBULL, D., EDMONDS, A., ADAR, E., AND BREUEL, T. 2001. Personalized search. *Comm. ACM 45*, 0, 50–55.

PONTE, J. M. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 275–281.

RESNICK, P., IACOVOU, N., SUCHAK, M., AND BERGSTORM, J. R. P. 1994. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 175–186.

ROBERTSON, S. E. AND WALKER, S. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 345–354.

ROBERTSON, S. E., VAN RIJSBERGEN, C. J., AND PORTER, M. F. 1980. Probabilistic models of indexing and searching. In *Proceedings of the 3rd Annual ACM Conference on Research and development in Information Retrieval*. 35–56.

SALTON, G., WONG, A., AND YANG, C. S. 1975. A vector space model for automatic indexing. *Comm. ACM 18*, 11, 613–620.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. T. 2000. Application of dimensionality reduction in recommender system—A case study. In *Proceedings of the ACM WebKDD Web Mining for E-Commerce Workshop*, 82–90.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. T. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.

SHAHABI, C. AND CHEN, Y. S. 2003. Web information personalization: Challenges and approaches. In *Proceedings of Databases in Networked Information Systems*. 5–15.

SHEN, X., TAN, B., AND ZHAI, C. 2005. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, 43–50.

SMYTH, B., BALFE, E., BRIGGS, P., COYLE, M., AND FREYNE, J. 2003. Collaborative Web search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (IJCAI'03).

SPITTERS, M. AND KRAAIJ, W. 2001. TNO at TDT2001: Language model-based topic detection. In *Topic Detection and Tracking Workshop Report*.

SUGIYAMA, K., HATANO, K., AND YOSHIKAWA, M. 2004. Adaptive Web search based on user profile constructed without any effort from users. In *Proceedings of the 13th International Conference on World Wide Web*. ACM Press, New York, NY, 675–684.

SULLIVAN, D. 2004. Eurekster launches personalized social search. Search engine watch. *http://searchenginewatch.com/searchday/article.php/3301481*.

SUROWIECKI, J. 2004. *The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*, Little Brown, 2004.

TEEVAN, J., DUMAIN, S. T., AND HORVITZ, E. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New York, NY, 449–456.

WANG, J., DE VRIES, A. P., AND REINDERS, M. J. 2006. A user-item relevance model for log-based collaborative filtering. In *Proceedings of 28th European Conference on Information Retrieval*. 37–48.

XU, J. AND CROFT, W. B. 1999. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, 254–261.

XUE, G. R., LIN, C, YANG, Q., XI, W., ZENG, H.-J., YU, Y., AND CHEN, Z. 2005. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 114–121.

YAMRON, J. P., CARP, I., GILLICK, L., LOWE, S. A., AND VAN MULBREGT, P. 1999. Topic tracking in a news stream. In *Proceedings of the DARPA Broadcast News Workshop*, San Francisco: Morgan Kaufmann.

ZENG, C., XING, C. X., AND ZHOU, L. Z. 2003. Similarity measure and instance selection for collaborative filtering. In *Proceedings of the 12th International Conference on World Wide Web*. 652–658.

ZHAI, C. AND LAFFERTY, J. 2001. Model-based feedback in KL divergence retrieval model. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, Georgia, USA, ACM Press, 403–410.

ZHAI, C. AND LAFFERTY, J. 2004. A study of smoothing methods for language models applied to information retrieval, *ACM Trans. Inform. Syst. 2*, 2, 179–214.

Received July 2006; revised April 2007, July 2008; accepted August 2008