Collaborative Web Search Utilizing Experts' Experiences

Jingyu Sun*,†, Xueli Yu* and Ning Zhong†,‡
*College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan, China
†International WIC Institute, Beijing University of Technology, Beijing, China
‡Dept. of Life Science and Informatics, Maebashi Institute of Technology, Maebashi, Japan
Email: {sunjingyu,yuxueli}@tyut.edu.cn, zhong@maebashi-it.ac.jp

***Abstract*—Collaborative Web search improves search quality by users' working in cooperation and is a subset of social search. Current Web browsers and search engines provide limited support for it. However, it is easier for experts, who are familiar with some topics, to fulfill they needs through search engines due to their backgrounds, domain knowledge and so on. A sharing experts' experiences approach should be struck based on today's Web browsers and major search engines. This paper presents a convenient way for users to share and utilize experts' experiences through a Web browser toolbar for collaborative Web search. The toolbar can catch search histories and favorites and display recommendations for every user in a popular Web browser through integrating with mainstream search engines like Google, Yahoo!, et al. These collected users' data are uploaded to a recommendation server, in which recommendations are built according to some rules based on an utilizing experts' experiences approach. The toolbar can download some valuable recommendations merging into default search list for prompting a searcher. The core of our proposed approach is a scalable method to measure "to what degree a user is an expert" for a given topic and to detect an expert's experiences based on a hierarchical user profile. Experiments showed that the novel collaborative Web search way is acceptant to users and experts' experiences improved search quality when compared to standard Google rankings. More importantly, results verified our hypothesis that a significant improvement on search quality can be achieved by utilizing experts' experiences.**

***Keywords*-collaborative Web search; experts' Web search experiences; user profile; personalization; social search.**

I. Introduction

As the amount of information on the Web continuously grows, Web search has become one of the prominent information behaviors. Generally, it is considered to be a solitary activity for satisfying users' individual needs. All mainstream search engines like Google, Yahoo!, et al. and Web browsers are designed for solo use. However, many tasks in both professional and casual settings can benefit from the ability to jointly search the Web with others [15]. For example, we consider a common user-case: data mining is an interesting topic to Yang and Sun. Yang is a PhD about data mining domain and has spent many years searching for relevant materials and papers on a range of research related topics; while Sun is a newcomer for this domain and successful searches can be elusive. Usually, Sun often struggles to find his wants in Google. But sometime he can't find them due to he has a few terms about data mining and wastes time re-searching for information he has found previously. Fortunately for Sun, Yang is one of his colleagues and he benefits from Yang and others when it comes to better understanding the things he should be searching for. Yang and others often email him some useful links about Web sites and papers that they have found. In this example, Sun benefits a lot from Yang and others. But they only depend on inefficient ways: to chat, to email and so on. In nature, this situation is an example about collaborative Web search without an effective tool to support.

In general, collaborative Web search often occurs in some search tasks, such as travel planning, literature search, technical information and so on. It is becoming a staple way to improve search quality by users' collaboration. Recently, some research efforts have focused on this area and some demo systems have developed, such as Heystaks [17], SearchTogether [13], S^3 [12], and CoSearch [2]. Primary results show that the process of collaborative Web search is more complex and it involves many questions [14], such as:

- Who are the people who engage in collaborative Web searches, and what is the nature of the relationships among these collaborators?
- What type of collaborative search tasks do people engage in, and what phases of these tasks provide opportunities for collaboration?

From prior example, we know that Sun and Yang engage in collaborative Web search. Obviously, Sun is a newcomer for data mining, but Yang can be taken as an expert for data mining. Sun benefits a lot from collaborative Web search. This example shows that collaborative Web search can improve search quality of common users (newcomers) through utilizing experts' experiences. However, today's collaborative search systems still don't provide a convenient way and an effective method to utilize experts' experiences based on current Web browser and major search engines. In addition, there are two core questions: how to find right experts? and how to detect experts' valuable experiences effectively?

In practice, there exist already many examples where someone considered to be an expert in a given search context may be a newcomer for another topic. One example is

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.265

that a "Database" teacher usually is easier to find some valuable information about "MySql" but is difficult to seek some literatures about "CPU" in Google. As a result, more information about "MySql" would be stored in her profile and she can be taken as an expert about "MySql" to some extent. That is to say, a user may be a partial expert and her user profile implies to what degree she is an expert for a given topic. On the other hand, her search experiences about "MySql" are suitable to help her students to know "MySql". It means search experiences yielded from a user's familiar topic usually are useful to other users who are unfamiliar to it. Therefore, a proper filtering of topics based on user profiles not only helps identify experts but also may help detect experts' valuable experiences.

However, users' experiences, i.e. search histories, favorites, etc., are mostly unstructured. It is hard to measure "to what degree a user is an expert" for a given topic directly. In addition, it is also difficult to incorporate them with search engines without right summarization. So, for the purpose of both collaborative Web search and Web personalization, it is necessary for a way to collect, summarize, and organize users' experiences into structured data and a method to measure "to what degree a user is an expert" for a given topic and to detect an expert's experiences.

This paper targets at exploring a collaborative Web search solution through utilizing experts' experiences and this is implemented in the following ways:

- Offers a convenient way where users depend on a Web browser toolbar for collecting and sharing their experiences. The toolbar is main interface of our proposed collaborative Web search engine and integrates with mainstream search engines like Google, Yahoo!, et al. for collaborative Web search. It usually is implemented as a plug-in of FireFox browser or Internat Explorer and can catch users' search histories and favorites. These users' data are uploaded to a recommendation server for building recommendations. In addition, recommendations are displayed through integrating with default result-list from Google or Yahoo based on the toolbar.
- Offers a method to measure "to what degree a user is an expert" for a given topic and to detect an expert's experiences based on a hierarchical user profile. The user profile is built hierarchically so that the higher-level interests with more supports are taken as user's familiar topics. Two parameters are used to measure "to what degree a user is an expert" and which experiences are valuable for a given topic. All of experiences related to user's familiar topics are taken as valuable experiences.
- Offers a clustered user profile to organize valuable experiences and some recommendation rules. The clustered user profile is built based on all experts' experiences related to one topic or search task. Recommendation rules are designed for prompting different type recommendations in a different place of search list.

The rest of the paper is organized as follows. The next section reviews related work focusing on social search, collaborative Web search, user profile and expertise-finding system. An overview of the problem is given in Section 3. Our approach is described in Section 4. Experiment results are presented in Section 5. Conclusions are presented in Section 6.

II. Related Work

This work is related to some research fields, i.e. social search, collaborative search, user profile and expertise-finding system. A brief summary is introduced as following.

With the growing of social data, Folksonomies like tags, et al. are emerging in social systems and there are around 115 million bookmarks on the del.icio.us social bookmarking site in 2008 [8]. Now more and more researchers begin to utilize them for social search. SBRank [22], a page popularity measure, is proportional to a number of existing social bookmarks. Zhou et al. [24] present a theoretically sound generative model for social annotations based on the language modeling approach. Hotho et al. [9] define a formal model for folksonomies and ranking algorithms called adapted PageRank and FolkRank. Bao et al. [5] propose two alternative algorithms, SocialSimRank and SocialPageRank. Abel et al. [1] evaluate these algorithms and propose a few new algorithms. In addition, some methods focus on personalized social search. Noll et al. [16] propose a reranking method based on users' tag profiles which are derived from their bookmarks in del.icio.us.

Collaborative Web search is a subset of social search. Recently, Morris [15], [13], [12], [14], Smyth [17], et al. and their teams focus on this new research field. Morris at el. [15] conduct a survey of 204 knowledge workers at a large technology company and present their findings that reveal that users often collaborate over both the process and products of search. Based on their findings, they create SearchTogether [13], S^3[12], and CoSearch [2] collaborative search systems. In addition, the recent work of Morris [14] investigates the who, what, where, when and why of collaborative search, and gives insight in how emerging solutions can address collaborators' needs. Smyth et al. [17] introduce a novel social search engine, named HeyStaks. It depends on a Web browser toolbar, which can integrate with Google, Yahoo and other search engines, to collect users' search histories, tags, votes and share them. This work has some similarity with our approach, however, our method explicitly utilizes experts' experiences for collaborative Web search based on a hierarchical user profile.

User profiles can be represented by a weighted term vector [19], weighted concept hierarchical structures [18] like ODP, other implicit user interest hierarchy [10], or Ontology [11]. Xu et al. [21] propose a hierarchical user profile used to control privacy. This work has some similarity with our approach, however, our method builds hierarchical user profiles used to choose experts' experiences through summarizing search cases including queries, tags, titles, snippets and so on.

In addition, expertise-finding systems [4] may be useful in helping people find trustworthy collaborators with knowledge of and interest in a particular topic. However, our work focuses on collaborative Web search in the online through utilizing experts' search histories directly, but expertise-finding systems is only used to find experts for a given topic or domain.

III. Problem Overview

There are two core issues for collaborative Web search utilizing experts' experiences. First one is to find a convenient way for users based on current Web browsers and major search engines. Second one is to choose right experts' experiences to recommend.

For first issue, with the development of Internet and software technology, mainstream Web browsers, i.e. Microsoft Internet Explorer (IE), Mozilla Firefox, et al., allow users extend their basic functions through developing a toolbar with plug-in technology or Firefox extension. For example, when someone browses a Web page in Firefox browser, a special toolbar can control the document model of a Web page and capture her click actions and extract the title, url, and others of the page. In addition, all data extracted can be uploaded to a recommendation engine server for processing with the help of some software development technologies like Ajax, Javascript, et al. and recommendations also be downloaded and merged with the return-list by a search engine. With the help of such a toolbar, current Web browsers and search engines are combined to support collaborative Web search in a convenient way for users.

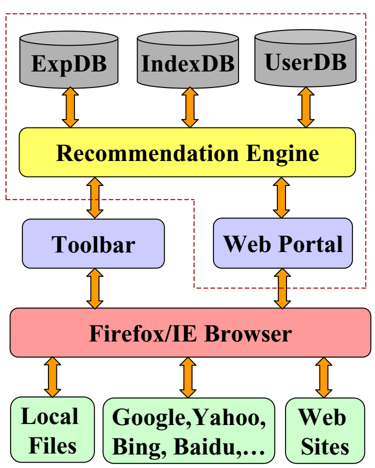

Figure 1. System overview

Based on above ideas, we design a social Web search demo system, named ExpertRec. As shown in Figure 1, it takes the form of two basic components: a client side browser toolbar and a back-end server. The ExpertRec toolbar is implemented as a Firefox extension, which can integrate with mainstream search engines like Google, Yahoo, Baidu, et al., for capturing users' search experiences (such as clicks of Google result-list, tags, votes and so on) and displaying recommendations. In addition, rankings from search engine results are revised by it based on the user profile and experts' experiences. The back-end server includes a recommendation engine, a Web portal and three databases which are used to store experiences (search histories, tags, votes), users and index them. The recommendation engine provides recommendations through combining collaborative filtering and case-based reasoning based on recommendation rules and analysis results of users' experiences by our proposed method. The Web portal provides several functions, such as to maintain search experiences, to index and so on, for users and administrators. One of three databases is an experience database, named ExpDB, to store users' experiences that are captured by the toolbar; the second one is an index database, named IndexDB, to store indexes of users' experiences; the last one is an user profile database, named UserDB, to store users' primary information and their interests (including familiar topics and supports), buddies and so on.

In brief, first issue is mainly a technical problem and we provide a toolbar with more functions to support collaborative Web search. It can integrate a Firefox Web browser and Google search engine and so on in current ExpertRec.

For second one, users' experiences, i.e. search histories, favorites and so on, are helpful to improve other users' search quality and experts' experiences are more valuable. However, only partial users' experiences are experts' experiences and a right analysis method is necessary. The question here is whether a solution can be found where users' experiences can be effectively filtered to improve the search quality. As a hierarchical user profile can summarize user's experiences into different levels with different supports, general topics with more supports can be taken as familiar topics and experiences under such topics can taken as experts' experiences. That is, we can measure "to what degree a user is an expert" for a given topic and detect an expert's experiences based on a hierarchical user profile.

But the solution of the second one influences directly search quality of users. The whole solution has three parts. First, a algorithm automatically builds a hierarchical user profile from search cases. Then, Two parameters for specifying recommendation requirements are offered to help the system to choose experts and their experiences for a given topic. Third, some recommendation rules are used to personalize the search results with the help of the hierarchical user profile. We discuss them in detail in following section.

IV. Collaborative Web Search Utilizing Experts' Experiences

A. Constructing a Hierarchical User Profile

Any users' experiences such as search histories, favorites, et al. could be the data source for user profiles and can be organized into search cases. A user's experience can be denoted by a search case including *title, queries, tags, votes, URLwords, snippet, selected-frequency*. Our hypothesis is that terms that frequently appear in *titles, queries, tags, snippets* of search cases can represent topics that interest users, and *votes, selected-frequency* of search cases influence degree of interests. Here we use an approach in [21] proposed by Xu et al. to build the hierarchical user profile based on frequent terms. In the hierarchy, general terms with higher frequency are placed at higher levels, and specific terms with lower frequency are placed at lower levels.

C represents the collection of all search cases and each case is a summary of a Web page including its *title, queries, tags, votes, URLwords, snippet, selected-frequency*. $C(t)$ denotes all cases covered by term t, i.e. all cases in which t appears, and $|C(t)|$ represents the number of cases covered by t. A term t is $frequent$ if $|C(t)| \geq minsup$, where $minsup$ is a system-specified threshold, which represents the minimum number of cases in which a frequent term is required to occur. Each frequent term indicates a possible user interest. In order to organize all the frequent terms into a hierarchical structure, relationships between the frequent terms are defined below.

Assuming two terms t_1 and t_2, the two heuristic rules used in this approach are summarized as follows:

1) ***Similar terms:*** Two terms that cover the document sets with heavy overlaps might indicate the same interest. Here we use the Jaccard function [7] to calculate the similarity between two terms: $Sim(t_1, t_2) = |C(t_1) \cap C(t_2)|/|C(t_1) \cup C(t_2)|$. If $Sim(t_1, t_2) > \delta$, where δ is another user-specified threshold, we take t_1 and t_2 as similar terms representing the same interest.
2) ***Parent-Child terms:*** Specific terms often appear together with general terms, but the reverse is not true. For example, "Windows" tends to occur together with "OS", but "OS" might occur with "Linux" or "Unix", not necessarily "Windows". Comparatively speaking, "OS" is a general term, but "Windows" is a specific term. Thus, t_2 is taken as a child term of t_1 if the condition probability $P(t_1|t_2) > \delta$, where δ is the same threshold in Rule 1.

Rule 1 combines similar terms on the same interest and Rule 2 describes the parent-child relationship between terms. Since $Sim(t_1, t_2) \leq P(t_1|t_2)$, Rule 1 has to be enforced earlier than Rule 2 to prevent similar terms to be misclassified as parent-child relationship. For a term t_1, any case covered by t_1 is viewed as a natural evidence of users' interests on t_1. In addition, cases covered by term t_2 that either represents the same interest as t_1 or a child interest of t_1 can also be regarded as supporting cases of t_1. Hence supporting cases on term t_1, denoted as $S(t_1)$, are defined as the union of $C(t_1)$ and all $C(t_2)$, where either $Sim(t_1, t_2) > \delta$ or $P(t_1|t_2) > \delta$ is satisfied.

Based on the above rules, a hierarchical user profile can be automatically built in a top-down fashion. The profile is represented by a tree structure, where each node is labelled a term t, and associated with a set of supporting documents $S(t)$, except that the root node is created without a label and attached with a user's name of C, which represent all personal cases. Starting from the root, nodes are recursively split until no frequent terms exist on any leave nodes. Two similar algorithms in [21] are used to build it.

B. Choosing Experts' Experiences

With the hierarchical user profile constructed above, every term with supporting search cases can be detected. In the following discussion, "topic" and "term" are indistinguishable in the context of the user profile. The support of an topic of a term t is $Sup(t)$, and $S(t)$ represents all the supporting cases for term t. $\sum Sup(t) = |C|$ is for all terms t on the leave node, where $|C|$ represents the total number of supports received from a user's search cases. In addition, our hypothesis is that a term t with lager $sup(t)$ represents a user's familiar topic and partial search cases in $S(t)$ are her valuable experiences.

The hierarchical user profile constructed is taken as an indicator of the user's possible familiar topics. According to probability theories, the possibility of a term can be calculated as $P(t) = Sup(t)/|C|$. With the context of information theory, the amount of information about a certain topic of the user is measured by its *self-information* [6]:

$$I(t) = log(1/P(t)) = log(|C|/Sup(t)). \qquad (1)$$

This measure has also been called *surprisal* by Myron Tribus [20], as it represents the degree to which people are surprised to see a result. More specifically, the smaller $Sup(t)$ is, the larger the self-information associated with the term t is, and the search case including term t is more valuable as it is a special search case for a user. This leads to two parameters for specifying the requirement of recommendation.

minFamiliar. The user profile above is organized from high-level to low-level. Terms associated with each node become increasingly specific as the list progresses, and same level terms are sorted from left to right in descending order of their supports. A threshold of $minFamiliar$ is defined to measure users familiar topics on both vertical and horizontal dimensions. With a specified $minFamiliar$, any term t in the user profile with $P(t) = Sup(t)/|C| \geq minFamiliar$ will be taken as a user's familiar topic.

Figure 2 is an example of the hierarchical user profile. Firstly, the possibility of every topic is calculated, for

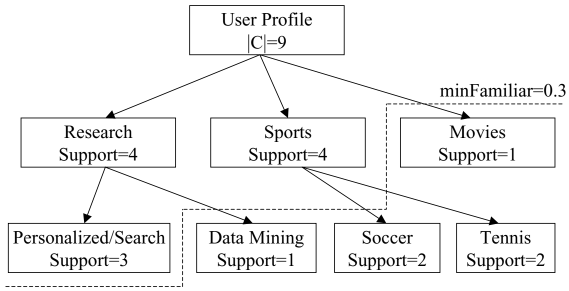

Figure 2. A hierarchical user profile

example, P("Sports")=Sup("Sports")$/|C| = 4/9 > 0.3$. If $minFamilar = 0.3$, topics above broken line are taken as familiar topics. The complete user profile is denoted as U, and $U[Fam]$ represents the familiar part of U, that is, the part above $minFamiliar$. Since the support for terms decreases monotonically travelling horizontally and vertically, the $U[Fam]$ will be a connected subtree of the complete user profile stemming from the user profile root. With the threshold $minFamiliar$, the system will know exactly which topics a user is familiar with.

For conventional, $U[Fam]$ is transformed into a list of weighted terms and the weight of each term in $U[Fam]$ is estimated by applying the concept of IDF (Inverse Document Frequency). Given a term t, the weight of t, denoted by w_t, is calculated as:

$$w_t = log(|C|/Sup(t)), \tag{2}$$

where $|C|$ represents the total number of search cases of $U[Fam]$, and $Sup(t)$ is the support of this term on the node in E. The user profile is expressed by a list $< t, w_t >$, where t is a term in $U[Fam]$ and w_t is the weight. For example. The list is $< research, 0.352 >$, $< sports, 0.352 >$, $< personalized/search, 0.477 >$.

In order to choose experts' experiences, we construct a clustered user profile with the same approach, named the expert profile E, through utilizing all cases of an ExpBase in ExpertRec, which is usually related to one search task or topic and includes several users' cases. However, since different users maybe visit same Web pages, *votes* and *selected-frequency* of a search case indicate degree of users' interest on it. So the number of supporting cases for a term t is adjusted as:

$$Sup(t) = \sum{}_{c \in S(t)} (\beta \cdot h(c) + \gamma \cdot v(c)), \tag{3}$$

where $h(t)$ represents *selected-frequency* of the search case c , $v(c)$ represents the score of the search case c and is defined as the minute of the number of supporters and non-supporters in ExpertRec, β and γ are two weight parameters and calculated as:

$$\beta = \eta \cdot \frac{|C|}{\sum{}_{c \in C} h(c)}, \gamma = (1 - \eta) \cdot \frac{|C|}{\sum{}_{c \in C} v(c)}, \tag{4}$$

where the parameter $\eta \in [0, 1]$ and usually is 0.8. In addition, for any case c in $S(t)$, if c appears in n nodes ($n \geq 1$), which was interpreted as c supporting all n terms, the support term from c in $S(t)$ is counted only as $1/n$.

expScore When a user inputs a query, a set of terms, in a search engines, the ExpertRec toolbar would capture and upload the terms to the recommendation server. The server would combine them and her profile to find valuable experts' experiences through travelling E. But which experiences are most valuable for her? In our opinions, search cases, which don't appear in her profile but include terms with larger self-information according to E, are more valuable. So we firstly choose such search cases according to E: their support topics include terms appeared in the query. Then a score, named $expScore$, for every search case, is computed by $\sum_t I(t)$, where t denotes a term which appears in the query and the search case at the same time. A search case with larger $expScore$ is recommended preferentially.

C. Recommendation Rules

In order to utilize experts' experiences for collaborative Web search, we propose three recommendation rules:

- First rule is that a recent search case associated with the query and appeared in her profile will be prompted in first place of search list and used to remind her in order to avoid to browse repeatedly.
- Second one is that a search case with largest $expScore$ is recommended in second place of search list and is used to recommend her a possible interesting new Web page.
- Third rule is that the search results returned by a search engine will be adjusted through incorporating the expert profile and the final ranking of the search results is decided by the search engine and E.

However, as first two rules provide two search results which can't be evaluated directly through comparing with standard results, we focus on the third rule in the following.

In ExpertRec, recommendations are built when a query is submitted to the recommendation server in five steps:

1) The expert profile of every ExpBase is built and represented by a set of $< t, w_t >$ pairs in the recommendation engine server.
2) The toolbar captures a query and the search results returned by a search engine and they are uploaded to the recommendation engine server. Each result comprises of a set of links related to the query, where each link is given a rank from the search engine, called DefaultRank.
3) For each of the returned link l, a score called $EScore$ is calculated by the expert profile as follows:

$$EScore(l) = \sum_t w_t \times f_t, \tag{5}$$

where t is any term in the expert profile, and f_t is the frequency of the term t in the snippet of the link l. An $ERank$ is assigned to each link according to its $EScore$, and the link with the highest $EScore$ will be ranked first.

4) Re-ranking results by combining ranks from both DefaultRank and ERank. The final rank, EERank (Experts' experiences Enhancing Rank), is calculated as:

$$EERank = \alpha * ERank + (1 - \alpha) * DefaultRank, \quad (6)$$

where the parameter $\alpha \in [0, 1]$ indicates the weight assigned to the rank from the expert profile. If $\alpha = 0$, the expert profile is ignored, and the final rank is decided by the expert profile instead of the search engine when $\alpha = 1$.

5) The toolbar downloads the final ranking of the search results and recommends them to the user.

V. Experiments

In this section all experiments are conducted with the following objectives: to verify the effectiveness of the clustered user profile to help improve search quality, and to explore the relationship between search quality and experts' experiences.

A. Experiment Setup

The approach is evaluated with 20 participants who are chosen from different research groups in our labs and with high levels of computer literacy and familiarity with Web search. They were divided into two groups of their free will: 10 for *A group* and 10 for *B group*. Participants in *A group* can run the toolbar on their own PC and issue their own queries for building expert profiles; participants in *B group* evaluate queries through a Google search wrapper. In addition, the hypothesis was that a user's search behavior would be allowed to capture in the real world.

There are two steps in our evaluation process. At first step, 4 ExpBases were created for 4 different topics: *data mining*, *Web search*, *OS* and *daily life*. Then participants in *A group*, who were familiar with at least 1 topic, were asked to select at least effective 100 queries according to their interests during 1 week. At second step, participants in *B group* were asked to select 25 queries for every topic from a list about 4 topics: *data mining*, *Web search*, *OS* and *daily life*. For each query, the top 50 links returned from our Google search wrapper and then displayed to the user. We believe these include the most meaningful results, and retrieving more links will not have a major impact on the experiment results due to their low Google search rankings. Given a set of links returned for a query, the participant was asked to determine which in their opinion were relevant. The links were presented in a random order so as not to bias the participants. The queries with no result or with no links marked as relevant by users were ignored.

To evaluate the search quality, we adopt a widely used measure, Average Precision [3], with a higher value indicating more relevant documents returned at an earlier time. Over a set of queries, search quality is represented by the mean of the average precisions, where *Average Precision* for a query is calculated as follows:

$$AveragePrecision = (\sum_{i=1}^{n} (i/(l_i.rank)))/n, \quad (7)$$

where l_i denotes the i^{th} relevant links identified for a query, and n is the number of relevant links. Each relevant link l_i identified by participants will be associated with two ranks: EERank which represents the final rank that combines both the expert profile and Google search rankings, and DefaultRank, which is the original Google ranking. Average precision is calculated for both two different rankings with 4 expert profiles. Intuitively, a higher average precision indicates a higher search quality.

All programs were implemented in Java. The two parameters mentioned in section IV.A are chosen empirically: $minsup = 5$ (through which most of the meaningless words are filtered); $\delta = 0.6$. And all participants are advised to use the same parameters for the purpose of comparability.

B. Effectiveness of the Expert Profile

First, it is a must to demonstrate the effectiveness of the expert profile in helping customizing search results. The users' experiences available in our program were search histories, Web pages from recommended emails, and favorites. The average number of the types of users' experiences stored in every ExpBase of recommendation server is that number of search histories is 842, number of Web pages from recommended emails is 28, number of favorites is 47. The experiences, which are not summarized into search cases and no frequent terms, were ignored.

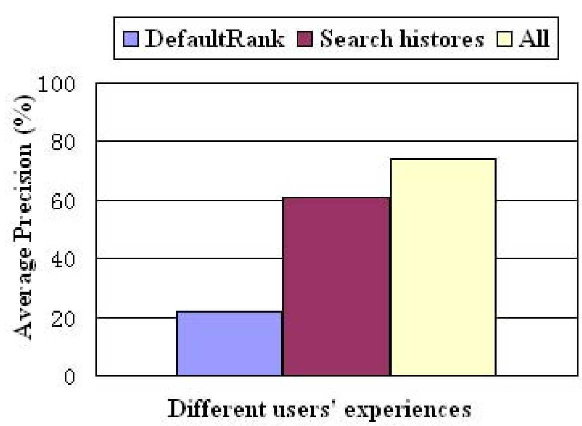

Figure 3. Effect of different users' experiences

In Figure 3, with all parameters fixed ($minFimiliar = 0.25$, $\alpha = 1.0$), the comparison of the average precisions for all queries of different topics, with different type experiences selected are shown. Compared to the original DefaultRank, the average precision that incorporates the expert profile is much higher, and the search quality improves. Experts' experiences always yield better results. The expert profile built from "all" search cases, including search histories, Web pages from recommended emails and favorites, has a better performance to using only search histories. Web pages from recommended emails and favorites seem have the positive effect on search quality because they only include a little noises.

Within all queries of different topics, the impact of the expert profile for EERank is studied by varying only parameter α. User's experiences are set to "all", $minFamiliar = 0.25$. Parameter α varies from 0 to 1.0, where $\alpha = 1$ indicates ranking search results by EScore only, and $\alpha = 0$ shows the results from the original Default search ranking.

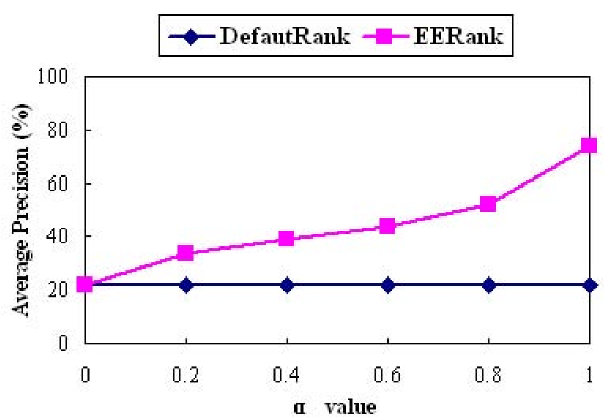

Figure 4. Impact of different α value

Figure 4 shows the average precisions of the *EERank*, which depend on the expert profile, and the original *DefaultRank* respectively. As α increases, the average precisions of the *EERank* increases almost linearly. The best result occurs when α is around 1.0. This indicates that the expert profile is important to get better results.

In Figure 5, the X-axis is changed to $minFamiliar(\alpha = 1.0)$. This shows that deleting greater amounts of unfamiliar topics ($minFamiliar$ from 0 to 0.4) can increase the search quality. The best result occurs when $minFamiliar$ is around 0.25. This indicates that our proposed method to measure to choose experts experiences is feasible.

The experiment results above illustrate two points: first, familiar topics (general terms) of users are much more useful than unfamiliar topics (specific terms) in helping to improve search quality. Second, only a few familiar topics utilized is not that useful. The experiments verify our hypothesis that utilizing experts' experiences could potentially return a

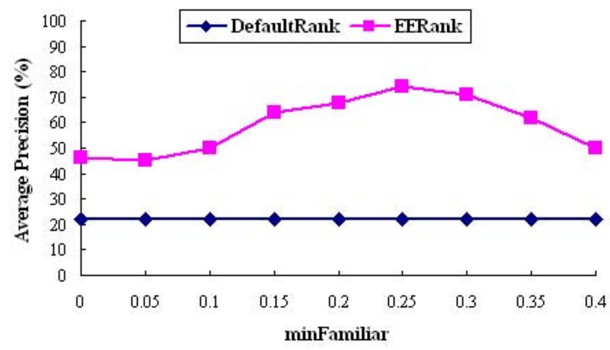

Figure 5. minFamiliar vs Search Quality

relatively high search quality.

C. Effectiveness of Collaborative Web Search

Since the toolbar provides many functions for a user to support collaborative Web search, i.e. the ability to tag or vote Web pages, to email findings to her buddies or chat with online friends, et al., users can conveniently share their experiences including all kinds of favorites or findings and so on in the online. For a search task, the toolbar allows users to create or join a ExpBase stored related search histories and share them with others. Usually, some new searched Web pages are recommended according to the second recommendation rule, and some most popular Web pages are prompted in order to be easily found for a newcomer in the first time or save time for an expert browsed them. All of above functions enhance collaborative Web search utilizing experts' experiences.

VI. Conclusions

Collaborative Web search is a promising way to improve search quality by users' working in cooperation. However, this approach requires a convenient way for users to work together. But current Web browsers and search engines provide limited support for this. In this paper, we introduced a feasible solution through a browser toolbar to combine a Web browser and major search engines like Google, Yahoo, and on on. On the other hand, high-quality users' experiences maybe improve the search quality. For a specific situation, it is easier for experts, who are familiar with some topics, to find they needs in a current search engine due to their backgrounds domain knowledge, et al. An approach utilizing experts' experiences was proposed for this goal based on a hierarchical user profile. First, a method was provided to the user for collecting, summarizing, and organizing her search cases into a hierarchical user profile, where general terms are placed to higher levels than specific terms. Through this profile, any user can be taken as an expert for a given topics and search cases under general terms are taken as experts' experiences. The degree an user is taken as an expert is adjusted the $minFamiliar$ threshold. An additional parameter to measure value of a search cases, $expScore$, was proposed to find most valuable experts' experience. In addition, all of experts' experiences were clustered into a hierarchical expert profile and three recommendation rules were proposed in order to utilize them for collaborative Web search with the higher search quality. Experiments showed that the hierarchical expert profile is helpful in improving search quality. The experimental results verified our hypothesis that experts' experiences can improve the search quality.

Yet, this paper is an exploratory work on the two aspects: First, we explorer a way to combine current Web browsers and search engines for collaborative Web search. Secondly, we try to define experts' experiences and utilize them to improve the search quality. There are a few of promising directions for future work. In particular, we are considering ways of finding right experts and their valuable experiences for a given query that we gain from expert-finding system. Also, we suspect that personalized collaborative Web search can be achieved if difference of the hierarchical expert profile and the user profile is measured for a specific query. In addition, some new outlier detection algorithms [23] may be used to mine a possible interesting search case for users.

ACKNOWLEDGMENT

The authors would like to acknowledge the following support for their research on ExpertRec: International collaboration Project of Shanxi (No. 2008081032); Youth Natural Science Foundation of Shanxi (No. 200821024); National Natural Science Foundation of China (No. 60873139). And special thanks for Xianhua Li and PhD Jian Yang.

REFERENCES

[1] F. Abel, N. Henze, and D. Krause. Ranking in folksonomy systems: can context help? *In Proceedings of CIKM*, pages 1429–1430, 2008.

[2] S. Amershi and M. Morris. Cosearch: A system for co-located collaborative web search. *CHI 2008*, pages 1647–1656, 2008.

[3] R. Baeza-Yates and B. Ribeiro-Neto. Modern information retrieval. *Addison Wesley Longman, MA*, 1999.

[4] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. *In Proceedings of SIGIR*, pages 43–55, 2006.

[5] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. *In Proceedings of WWW*, pages 501–510, 2007.

[6] T. M. Cover and J. A. Thomas. Elements of information theory, 1st edition. *Wiley-InterScience, New York, NY*, 1991.

[7] D. Eduardo. On clustering and evaluation of narrow domain short-text corpora. *PhD thesis*, 2008.

[8] P. Heymann, G. Koutrika, and H. Garcia-Molina. an social bookmarking improve web search? *In Proceedings of WSDM*, pages 195–206, 2008.

[9] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *n Y. Sure and J. Domingue, editors, The Semantic Web: Research and Applications*, 4011 of LNAI:411–426, 2006.

[10] H. Kim and P. K. Chan. Learning implicit user interest hierarchy for context in personalization. *In Proceedings of International Conference on Intelligent User Interface (IUI)*, 2003.

[11] Y. F. Li and N. Zhong. Mining ontology for automatically acquiring web user information needs. *IEEE Transactions on Knowledge and Data Engineering*, 18:554–568, 2006.

[12] M. Morris and E. Horvitz. S3: Storable, shareable search. *Interact 2007*, pages 120–123, 2007.

[13] M. Morris and E. Horvitz. Searchtogether: An interface for collaborative web search. *IUIST 2007*, pages 3–12, 2007.

[14] M. Morris and J. Teevan. Collaborative web search - who, what, where, when and why. *Claypool Publishers*, pages 1–100, 2009.

[15] M. R. Morris. A survey of collaborative web search practices. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1657–1660, 2008.

[16] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. *Ihe Semantic Web*, 4825:367–380, 2007.

[17] B. Smyth and P. Champin. The experience web: A case-based reasoning perspective. *In Grand Challenges for reasoning from experiences, Workshop at IJCAI'09*, pages 566–573, 2009.

[18] M. Speretta and S. Gauch. Personalized search based on user search histories. *In Proceedings of Web Intelligence*, pages 622–628, 2005.

[19] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. *In Proceedings of WWW*, 2004.

[20] M. Tribus. Thermostatics and thermodynamics. *D. Van Nostrand, New York, NY*, 1961.

[21] Y. Xu, B. Zhang, Z. Chen, and K. Wang. Privacy-enhancing personalized web search. *In Proceedings of WWW*, pages 591–600, 2007.

[22] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? *In Proceedings of JCDL*, pages 107–116, 2007.

[23] J. Yang, N. Zhong, Y. Y. Yao, and J. Wang. Local peculiarity factor and its application in outlier detection. *KDD 2008*, pages 776–784, 2008.

[24] D. Zhou, J. Bian, S. Zheng, H. Zha, and C. L. Giles. Exploring social annotations for information retrieval. *In Proceedings of WWW*, pages 715–724, 2008.