Personalized Web Search Using Clickthrough Data and Web Page Rating

XuePing Peng
Beijing Institute of Technology, Beijing, 100081, China
Email: pengxp@bit.edu.cn

ZhenDong Niu*, Sheng Huang and Yumin Zhao
Beijing Institute of Technology, Beijing, 100081, China
Email: {zniu, 20812036, oblivion}@bit.edu.cn

***Abstract*—Personalization of Web search is to carry out retrieval for each user incorporating his/her interests. We propose a novel technique to construct personalized information retrieval model from the users' clickthrough data and Web page ratings. This model builds on the user-based collaborative filtering technology and the top-N resource recommending algorithm, which consists of three parts: user profile, user-based collaborative filtering, and the personalized search model. Firstly, we conduct user's preference score to construct the user profile from clicked sequence score and Web page rating. Then it attains similar users with a given user by user-based collaborative filtering algorithm and calculates the recommendable Web page scoring value. Finally, personalized informaion retrieval be modeled by three case applies (rating information for the user himself; at least rating information by similar users; not make use of any rating information). Experimental results indicate that our technique significantly improves the search performance.**

***Index Terms*—Personalization, Web page rating, information retrieval, clickthrough data**

I. INTRODUCTION

As the amount of information on the Web rapidly increases, it creates many new challenges for Web search. Millions of searches are conducted every day on search engines such as Yahoo!, Google and Bing, etc. Despite the popularity, search engines have their deficiencies: given a query, they usually return a huge list of results, the pages ranked at top may not meet users' needs and the same result regardless of who submitted the query [1]. One reason for this problem is the keyword-based query interface, which is difficult for users to describe exactly what they need. Besides, typical search engines often do not exploit user information. Even two users submit the same query, their information need may be different [21-22].

Personalized Web search is to carry out retrieval for each user incorporating his/her own information need. To solve this problem, researchers have developed systems that adapt their behavior to the goals, tasks, interests, and other characteristics of their users. Based on models that capture important characteristics of users, these personalized systems maintain their users' preferences and take them into account to customize the content generated or its presentation to the different individuals [2]. Some Web search systems use relevance feedback to refine user needs or ask users to register their demographic information beforehand in order to provide better service[23]. Since these systems require users to engage in additional activities beyond search to specify/modify their preferences manually, approaches that are able to implicitly capture users' information needs should be developed.

This paper focuses on utilizing clickthrough data and Web page ratings to improve Web search. Clickthrough data can be extracted from a large amount of search logs accumulated by web search engines. These logs typically contain user-submitted search queries, the URL of Web pages which are clicked by users in the corresponding search result page[24]. The data objects contained in the clickthrough data are of different types: user, query and Web page. By performing analysis on the clickthrough data, we attempt to discover the latent factors among these multi-type objects[1]. However, most of these references extract only clickthrough data for analysis, and ignore the specific characteristics of Web pages. Page rating is one important characteristic， which can be calculated from explicit relevance rates of users who browsed the Web page. By analyzing associations among clickthrough data multi-type objects and computing Web page rating, we construct a personalized search model, and then re-rank search results by the model.

In this paper, by analyzing the clickthrough data and calculating Web page rating, we propose a novel, effective and efficient personalized Web search model. In this model, we give solutions to the following two problems: (1) How to create user profiles, and (2) How to return the different results when the same query is submitted by different users?

Manuscript received Sep. 8, 2011; revised Oct. 20, 2011; accepted Oct. 28, 2011.
Project number: 1110012040112, 91101,3070012231001
*Corresponding author's email: zniu@bit.edu.cn

doi:10.4304/jcp.7.10.2578-2584

The remainder of this paper is organized as follows. Section 2 provides related work. Section 3 gives a brief introduction to personalized Web search model. Section 4 presents the experimental results and Section 5 offers some concluding remarks and directions for future research.

II. Related Work

A. Personalized Web Search

Different users may prefer different results for the same query. Personalized search [25-29] aims to provide the most relevant search results to individual users based on their interests. Personalized search comprises two major components: (1) User profiles, and (2) The actual search algorithm [19].

Approaches focused on the User Profile. Sugiyama et al. [31] analyzed surfing behavior and generated user profiles as features (terms) of the visited pages. Upon issuing a new query, the search results were ranked based on the similarity between each URL and the user profile. Machine Learning [32] was used on the past click history of the user in order to determine topic preference vectors and then apply Topic-Sensitive PageRank [33]. User profiling based on browsing history has the advantage of being rather easy to obtain and process. This is probably why it is also employed by several industrial search engines.

Approaches focused on the Personalization Algorithm. Effectively building the personalization aspect directly into PageRank [34] has received much attention recently. Haveliwala [33] computed a topic-oriented PageRank, in which 16 PageRank vectors biased on each of the main topics of the Open Directory were initially calculated off-line, and then combined at run-time based on the similarity between the user query and each of the 16 topics. More recently, Nie et al. modified the idea by distributing the PageRank of a page across the topics it contains in order to generate topic oriented rankings. Jeh and Widom proposed an algorithm that avoids the massive resources needed for storing one Personalized PageRank Vector (PPV) per user by precomputing PPVs only for a small set of pages and then applying linear combination. As the computation of PPVs for larger sets of pages was still quite expensive, several solutions have been investigated, the most important ones being those of Fogaras and Racz, and Sarlos et al., the latter using rounding and count-min sketching in order to fastly obtain accurate enough approximations of the personalized scores.

Only by opening to the outside world, can it bring in adequate flow of negative entropy, make the dissipation occur between telecom industry and the environment, and ultimately evolve to the dissipative structure. The open feature of telecom industry system is the prerequisite and essential condition of self-organized industrial system. As a nation’s economical and foundational industrial, telecom industry is in a complex environment and is interdependent with the external environment. On the one hand, the environment provides a variety of factors required by the system to survive and develop, such as materials, technology, information, capital elements. On the other hand, telecom companies export to the environment products and services and dynamically improve the environment through efforts, to create a more favorable environment for the development.

B. Collaborative Filtering

We may distinguish two broad categories of collaborative recom-mendation systems, namely content-based and collaborative filtering. A content-based system selects items based on the correlation between the content of the items (e.g. keywords describing the items, such as album genre, artists, etc., for music tracks) and the users' preferences [35]. However, it is limited to dictionary-bound relations between the keywords used by users and the descriptions of items and therefore does not explore implicit associations between users.

Collaborative filtering systems are divided into two categories, i.e. memory-based and model-based. In the memory-based systems we calculate the similarity between all users, based on their ratings of items using some heuristic measure such as the cosine similarity or the Pearson correlation score. Then we predict a missing rate by aggregating the ratings of the k nearest neighbours of the user we want to recommend to. The problem with memory-based systems is that we have to decide on a rather arbitrary basis over parameters such as the number of neighbours. What is more, in the case of social networks there is no straightforward way to introduce similarities between users based on friendships and social tagging, other than some way of ad hoc interpolation of similarity weights from those different sources.

The model-based filtering systems assume that the users build up clusters based on their similar behaviour in rating of items. A model is learned based on patterns recognised in the rating behaviours of users using clustering, Bayesian networks and other machine learning techniques. The problem with model-based methods is that it is necessary to fine-tune several parameters of the model as well as the fact that the models produced might not generalise well in radically different context. What is more, as in the case of memory-based systems extra effort and training needs to be done in order to introduce knowledge from social networks [18].

C. Clickthrough Data

User click-through data can be extracted from a large amount of search logs accumulated by web search engines. These logs typically contain user-submitted search queries, followed by the URL of Web pages which are clicked by users in the corresponding search result page. Although these clicks don’t reflect the exact relevancy, they provide valuable indications to the users’ intention by associating a set of query terms with a set of web pages. If a user clicks on a web page, it is likely that the web page is relevant to the query, or at least related to some extent. Many valuable applications have been proposed along this direction, such as query suggestion [3][4][5], query expansion [6], query clustering [7-8][14-15], web page summarization [12], web search results

optimizing[9-10][11][13] and conducting other interesting work [16-17].

III. PROPOSED APPROACH

A. User Profile

1) Sequence Score

Definition: A retrieval transaction is user's browsed sequence for search results, and is noted as *tran*.

A *tran* can record the accessed information of the search results after the user put the query strings in search engine. For example, if a user query "user model", the search result is a list in some order (eg: "*page*1, *page*2, *page*3, *page*4"), And the user's accessed sequence for the results is "*page*3, *page*2, *page*4", our model will capture a tuple "<*SessionId*, "user model", (*page*3, *page*2, *page*4)>" by analyzing the user's clickthrough data. The score of each item of the user's accessed sequence give the following evaluation equation,

$$Score\ (sequence\ )_{page} = \frac{1}{N} \cdot \sum_{i=1}^{N} Score\ (page\ )_i \quad (1)$$

$$Score(page)_i = \frac{m - j + 1}{m} \quad (2)$$

Where $Score(page)_i$ is the web page score in the "*SessionId*" *i*, *N* is the number of the different sessions in which a user browsed the same web page. *m* is length of user's accessed sequence in the session, and *j* is page's position in the user's accessed sequence.

2) Web Page Rating

The score of Web page rating,

$$Score(rate)_{page} = \frac{1}{n} \cdot \sum_{i=1}^{n} rate_i \cdot f(n) \quad (3)$$

Where *n* is the number of users who give relevance evaluation to the Web page, and $rate_i$ is score value of relevance rate given by user *i*. $f(n)$ is an increasing function of parameter *n*; the greater is the value of *n*, the more popular is the Web page.

3) Preference Score

User k preferences score for Web page *p*,

$$pref(k, p) = \delta \cdot Score(rate)_{page} + (1 - \delta) Score(sequence)_{page} \quad (4)$$

Where $Score(rate)_{page}$ is the score of Web page rating and $Score(sequence)_{page}$ is the score of user's browsed sequence. And δ is is an impact factor, whose range is in [0, 1]. The user profile can be created as following Table I.

TABLE I

USER PROFILE

UserId	PageId	Rating	Time
58743	29086	0.7321	22:50:21 24-11-2010
89301	8329	0.6859	22:50:33 24-11-2010
6741	73429	1.2942	22:50:45 24-11-2010
…	…	…	…

B. User-based Collaborative Filtering

User-based collaborative filtering predicts a test user's interest in a test item based on rating information from similar user profiles [1][5][14]. Each user profile (row vector) is sorted by its dis-similarity towards the test user's profile. Ratings by more similar users contribute more to predicting the test item rating. The set of similar users can be identified by employing a threshold or selecting top-*N*. In the top-*N* case, a set of top-*N* similar users $S_u(u_k)$ towards user *k* can be generated according to,

$$S_u(u_k) = \{u_a \mid rank\ S_u(u_k, u_a) \le N, x_{a,m} \ne \Phi\} \quad (5)$$

Where $|S_u(u_k)|=N$. $s_u(u_k, u_a)$ is the similarity between users *k* and *a*. Cosine similarity and Pearson's correlation are popular similarity measures in collaborative filtering, see e.g. [1][5]. The similarity could also be learnt from training data [9]. This paper adopts the Pearson's correlation similarity measure, comparing two user profiles by the Pearson's correlation of the similarity between the corresponding row vectors [21].

Consequently, the predicted rating $pref_{rec}(k,p)$ of test item *p* by test user *k* is computed as following,

$$pref_{rec}(k, p) = \sum_{j=1}^{N} pref(j, p) \bullet (s_u(u_k, u_j) + 1.0) \quad (6)$$

Where $pref_{rec}(k,p)$ is the recommendable Web page scoring value, *N* is the number of the top-N most similar users, and $pref(j,p)$ is preference value of user *j* for resource *p*. And $s_u(u_k, u_a)$ represents the similarity between user *u* and user *j*.

C. Personalized Search Model

According to selecting top-scoring documents from (6) and documents of interest to users including users accessed to and system predicted, we proposed a personalized Web search retrieval model which different users entering the same query keywords, the search results list is different. The model is described as following,

$$Score(q, p) = \begin{cases} (1-\alpha) Sim(q, p) + \alpha \cdot pref(k, p), p \in \text{list}_{\text{pref}} \\ (1-\beta) Sim(q, p) + \beta \cdot pref_{rec}(k, p),\ p \in \text{list}_{\text{rec}} \\ Sim(q, p),\ \ p \notin \text{list}_{\text{pref}} \text{ and } p \notin \text{list}_{\text{rec}} \end{cases} \quad (7)$$

Where $Score(q,p)$ is the score of Web page *p* for query *q* , $Sim(q,p)$ is the is the similarity between Web page *p* and query *q*; $pref(k,p)$ is the directly preference of given user *k* for Web page *p*, $pref_{rec}(k,p)$ is the predicted rating of given user *k* for Web page *p*; α and β are impact factors of the $pref(k,p)$ and $pref_{rec}(k,p)$; $list_{pref}$ is the collection of user *k* explicitly interested documents and $list_{rec}$ are the collection of user *k* implicitly predicted documents , and $list_{pref} \cap list_{rec} = \Phi$.

If there is rating information for the user himself for the particular page, then the first case applies.

If there is at least rating information by similar users for the particular page, then the second case applies.

Otherwise, the third case applies, which does not make use of any rating information.

IV. EXPERIMENTAL SETUP

A. Data Sets

Clickthrough data can be recorded with little over head and without compromising the functionality and usefulness of the search engine. In particular, compared to explicit user feedback, it does not add any overhead for the user. The query q and the returned ranking r can easily be recorded whenever the resulting ranking is displayed to the user. For recording the clicks, a simple proxy system can keep a logfile [10]. In this paper, we collect clickthrough data by using a proxy server of Web server side. The data include user login information, query string, the Web page id, Session Id, clicked sequence of search results, and the visiting time. The Table II describes the data information.

The Web page rating can be explicitly recorded after user browsed the page, and meanwhile provided relevance score to it.The range of relevance score is from 0 to 1, and includes 0 and 1. If a user thought a page browsed was not relevant, he/she could give relevance score of the page to 0. On the contrary, he/she could give relevance score of the page to a number greater than 0.

Our experiments are performed on the China Education Television (CETV) Learning Mall Resource Set, which contains 312,477 pieces of resource, and uploaded, by 5,664 resource producers. We have 165,379 users for our system, and get 130,452 rating records.

In our experiments system, we trace user's searching and browsing activity, and to update user's interest, then we provide personalized Web search to users according to their preferences.

B. Evaluation Metrics

We evaluate the ranking algorithms over a range of accepted information retrieval metrics, namely *Precision at K* (*P*(*K*)) and *Mean Average Precision* (*MAP*). Each metric focuses on a defferent aspect of system performance, as we describe below [9].

Precision at K: As the most intuitive metric, *P*(*K*) reports the fraction of documents ranked in the top *K* results that are labeled as relevant. In our setting, we require a relevant document to be labeled "Good" or higher. The position of relevant documents within the top *K* is irrelevant, and hence this metric measure overall user satisfaction with the top *K* results.

MAP: Average precision for each query is defined as the mean of the precision at *K* values computed after each relevant document was retrieved. The final MAP value is defined as the mean of average precisions of all queries in the test set. This metric is the most commonly used single-value summary of a run over a set of queries.

C. Ranking Methods Compared

BM25F: As a strong web search baseline we used the BM25F scoring, which was used in one of the best performing systems in the TREC 2004 Web track [12, 13]. BM25F and its variants have been extensively described and evaluated in IR literature, and hence serve as a strong, reproducible baseline. The BM25F variant we used for our experiments computes separate match scores for each "field" for a result document (e.g., body text, title, and anchor text), and incorporates query-independent link based information (e.g., PageRank, ClickDistance, and URL depth). The scoring function and field-specific tuning is described in detail in [12]. Note that BM25F does not directly consider explicit or implicit feedback for tuning.

BM25FP: The ranking produced by incorporating clickthrough statistics and Web page rating to reorder web search results ranked by BM25F above.

Lucene: Apache Lucene [30] is a high-performance and full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search. Lucene is scalable and offers high-performance indexing, and has become one of the most used search engine libraries in both academia and industry. Lucene ranking function, the core of any search engine applied to determine how relevant a document is to a given query, is built on a combination of the Vector Space Model (VSM) and the Boolean model of Information Retrieval. The main idea behind Lucene approach is the more times a query term appears in a document relative to the number of times the term appears in the whole collection, the more relevant that document will be to the query. Lucene uses also the Boolean model to first narrow down the documents that need to be scored based on the use of Boolean logic in the query specification.

LuceneP: The ranking produced by reordering the Lucene results using clickthrough statistics and Web page rating.

D. Users Evaluation

We use user's browsing sequence and page turning activity to test the accuracy of the search results list in our model, which is also called users evaluation.

The higher is the similarity of the search results sequence and user's browsing sequence; the higher is the

TABLE II
INFORMATION FORMAT OF CLICKTHROUGH DATA

ID	Query	PageId	Rank	UserId	SessionId	Time
1	User model	47806	3	58743	8232328228986249	9:21:43 24-11-2010
2	User model	38570	4	58743	8232328228986249	9:22:15 24-11-2010
3	User model	29086	6	58743	8232328228986249	9:22:15 24-11-2010
4	Web search	8329	2	89301	1923744500763862	9:24:27 24-11-2010
5	Personalized search	73429	1	6741	2785098742726650	9:30:36 24-11-2010
…	…	…	…	…	…	…

retrieval precision.

The sequence of the search results list is *vector1*= <(1, $\frac{m-1+1}{m}$),(2, $\frac{m-2+1}{m}$),…, (i, $\frac{m-i+1}{m}$),…, (m, $\frac{m-m+1}{m}$)>, where *m* represents the resource number in the recommended results list, *i* represents the *i-st* resource in the list.

User's browsing sequence is *vector2*= <(1, 0),(2, $\frac{n-1+1}{n}$),(3, $\frac{n-2+1}{n}$),…, (i, $\frac{n-j+1}{n}$), … , (m, $\frac{n-k+1}{n}$)>, where *n* represents the user-browsed resource number, *j* represents the *j-st* resource in the browsing list, k represents the *k-st* resource in the browsing list.

For example, the search results list is *item1,item2,…, itemi,…, item9, item10*, and *m* is 10 here.

And the user-browsed resource list is *item2, item4, item6, item8, item7*, and *n* is 5. Then we get vector1= <(1,1), (2,0.9), (3,0.8), (4,0.7), (5,0.6), (6,0.5), (7,0.4), (8,0.3), (9,0.2), (10,0.1)>; vector2= <(1,0), (2,1), (3,0), (4,0.8),(5,0), (6,0.6), (7,0.2),(8,0.4),(9,0),(10,0)>. We now use the laws of cosines to calculate the similarity between the two vectors. The greater is the similarity value, the higher is the retrieval precision.

We compare the search results between our information retrieval model (BM25FP) and the base information retrieval model (BM25F).

Fig.1 shows the sequence similarity between BM25FP and BM25F.The sequence similarity of the BM25FP is much better than the BM25F from the Fig.2 because the BM25F is only related to the similarity of query and document without considering user's preference.

E. Impact of Parameters

Recall the two parameters in (7): α balance the scores between the query similarity and user's preference score, and β balances the scores between the query similarity and the predicted rating, we first test the sensitivity of α, setting β to zero. This scheme counts directly on the user preference score, but does not use user-based collaborative filtering prediction. Fig. 2 shows web search MAP against varying α from zero (a pure information retrieval model) to one (a user preference score approach). The value of the optimal α demonstrates that interpolation between pure information retrieval model and user preference score approaches improves the Web search performance. More specifically, the best results are obtained with α around 0.4.

Fig. 3 shows the sensitivity of β after fixing α to 0.4. The graph plots the MAP when parameter β *is* varied from zero (a pure information retrieval model approach) to one (the predicted rating approach). We observe that β reaches its optimal in 0.2.

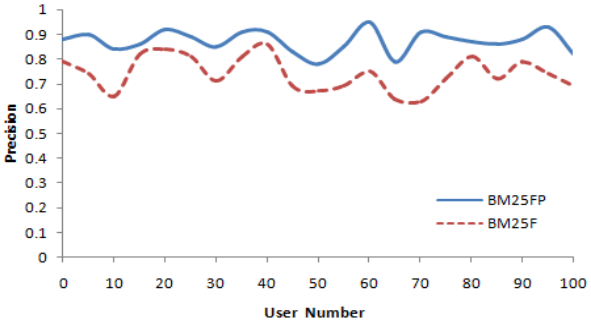

Figure 1. Sequence similarity between two models

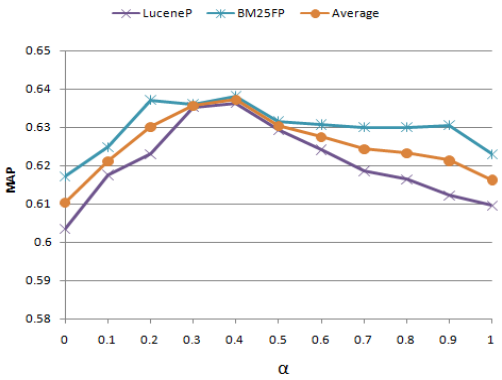

Figure 2. Impact of the parameter α

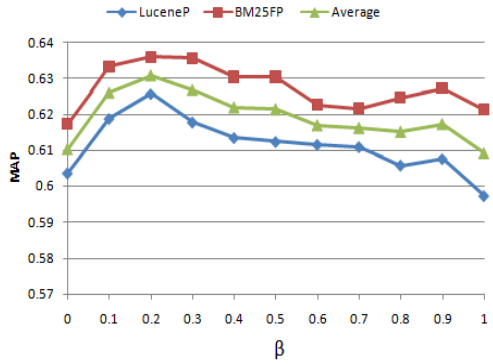

Figure 3. Impact of the parameter β (α=0.4)

Additional experiments (not reported here) verified that there is little dependency between the choice of α and the optimal value of β. The optimal parameters can be identified by using the cross validation from the training data.

F. Personalized Search Performance

We continue with a comparison to results obtained with other methods, setting α to 0.4 and β to 0.2. We first compare our results (BM25FP) to the standard BM25F. We report results for test the precision at 5, 10 and 20. The first two rows of Table III summarize the results, showing the performance of the BM25FP is better than the BM25F. Next, we first compare the LuceneP to the standard Lucene with the same condition. The last two rows of Tab.III summarize the results, showing the performance of the latter is better than the former too.

TABLE III

PRICISION COMPARISONS

	P@5	P@10	P@20	MAP
BM25F	0.80	0.76	0.72	0.6035
BM25FP	0.88	0.85	0.80	0.7207
Lucene	0.88	0.72	0.70	0.6172
LuceneP	0.88	0.82	0.78	0.7178

V. CONCLUSION AND FUTURE WORK

This article has proposed a personalized Web search model based on the method which calculates users' preferences according to the user's search behaviors and resource properties. This model has fully used the information in these two areas, does not need the user to make the appraisal when he or she glances over information, the system will analyze and quantize user's behaviors automatically. According to the user model which formerly established, this article simultaneously proposed the resources filtering and recommendation algorithm, which was based on Top-N resource recommending method.

ACKNOWLEDGMENT

This work was supported by Program for New Century Excellent Talents in University, China (NCET-06-0161, 1110012040112), Fok Ying Tong Education Foundation China(91101), BIT Major Foundational Research Project Supported (3070012231001), Beijing Institute of Technology graduate student scientific and technological innovation project.

REFERENCES

[1] J. T. Sun, H. J. Zeng, H. Liu, Y. Lu, and Z. Chen, "Cubesvd: a novel approach to personalized web search," Proceedings of the 14th international conference on World Wide Web, Chiba, Japan, May 10-14, 2005.

[2] Y. E. Ioannidis and G. Koutrika, "Personalized systems: models and methods from an IR and DB perspective.", Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005.

[3] C. Huang, L. Chien, and Y. Oyang, "Relevant term suggestion in interactive web search based on contextual information in query session logs," JASIST 54(7): 638-649,2003.

[4] N. J. Belkin, "Helping people find what they don't know," Communications of the ACM, No.8, 2000, pp58-61.

[5] H. Ma, H. Yang, I. King, and M. R. Lyu, "Learning latent semantic relations from clickthrough data for query suggestion," Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA, October 26-30, 2008.

[6] H. Cui, J.R. Wen, J.Y. Nie, and W.Y. Ma, "Query expansion by mining user logs," IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No. 4, July/August 2003.

[7] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, , Boston, MA, USA, August 20-23, 2000.

[8] J.R. Wen, J.Y. Nie, and H.J. Zhang, "Clustering user queries of a search engine," Proceedings of the Tenth International World Wide Web Conference, Hong Kong, China, May 1-5, 2001.

[9] E. Agichtein, E. Brill, and S. Dumai, "Improving web search ranking by incorporating user behavior information," Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006.

[10] T. Joachims, "Optimizing search engines using clickthrough data,". Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26, 2002.

[11] T. Joachims and F. Radlinski, "Search engines that learn from implicit feedback," *Computer*, No.8, 2007, pp34-40.

[12] J. T. Sun, D. Shen, H. J. Zeng, Q. Yang, Y. Lu, and Z. Chen, "Web-page summarization using clickthrough data," Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005.

[13] X. Wang and C. Zhai, "Learn from web search logs to organize search results," Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007.

[14] J. R. Wen, J. Y. Nie, and H. Zhang, "Query clustering using user logs," ACM Trans. Inf. Syst., No.1, 2002, pp59–81.

[15] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000.

[16] M. Pasca and B. V. Durme, "What you seek is what you get: Extraction of class attributes from query logs," Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007.

[17] D. Shen, M. Qin, W. Chen, Q. Yang, and Z. Chen, "Mining web query hierarchies from clickthrough data," Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 22-26, 2007.

[18] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, July 19-23, 2009.

[19] P.A. Chirita, C.S. Firan, and W. Nejdl,"Personalized query expansion for the web," Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007.

[20] J. Wang, A. P. de Vries and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006.

[21] R. B. Almeida and V. A. F. Almeida, "A community-aware search engine." Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004.

[22] F. Liu, C. Yu, and W. Meng, "Personalized web search by mapping user queries to categories," Proceedings of the International Conference on Information and Knowledge Management, McLean, VA, USA, November 4-9, 2002.

[23] L. Fitzpatrick and M. Dent, "Automatic feedback using past queries: social searching?," Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, USA, July 27-31, 1997.

[24] G.R. Xue, H.J. Zeng, Z. Chen, Y. Yu, W.Y. Ma, W.S. Xi, and W.G. Fan, "Optimizing web search using web clickthrough data," Proceedings of the International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004.

[25] Z. Dou, R. Song, J. Wen, "A large-scale evaluation and analysis of personalized search strategies," Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, May 8-12, 2007.

[26] F. Qiu, and J. Cho, "Automatic identification of user interest for personalized search," Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, UK, May 23-26, 2006.

[27] B. Tan, X. Shen, C. Zhai, "Mining long-term search history to improvesearch accuracy," Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006.

[28] J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing search via automated analysis of interests and activities," Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005.

[29] J. Teevan, E. Adar, R. Jones, and M. Potts. "Information re-retrieval: Repeat queries in yahoo's logs," Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007.

[30] O. Gospodnetic, and E. Hatcher, Lucene in action, Manning Publications Co.2005 .

[31] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," Proceedings of the 13th international conference on World Wide Web, New

[32] York, NY, USA, May 17-20, 2004.

[33] F. Qiu and J. Cho, "Automatic indentification of user interest for personalized search," Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006.

[34] T. Haveliwala, "Topic-sensitive pagerank," Proceedings of the Eleventh International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA, 7-11 May 2002.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Technical report, Stanford University, 1998.

Xueping Peng Anhui Province, China. Birthdate: October, 1980. is a Computer Software and Theory Ph.D. student in the School of Computer Science and Technology at Beijing Institute of Technology. And research interests on information retrieval, personalized recommendation, web data mining.

Zhendong Niu Anhui Province, China. Birthdate: October, 1968. is Computer Software and Theory Ph.D., graduated from Dept. Computer Science and Technology at Beijing Institute of Technology. And research interests on computer software architecture, knowledge management, intelligent education software system, digital library, neural information system.

He is a professor in the School of Computer Science and Technology at Beijing Institute of Technology, China.

Sheng Huang Henan Province, China. Birthdate: June, 1986. is a Computer Software and Theory Ph.D. student in the School of Computer Science and Technology at Beijing Institute of Technology. And research interests on text ming and opinion mining.

Yumin Zhao Henan Province, China. Birthdate: June, 1980. is a Computer Software and Theory Ph.D. student in the School of Computer Science and Technology at Beijing Institute of Technology. And research interests on digital library, data mining, computer network.