International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.10, December 2012 – www.ijais.org

Evaluating the Effectiveness of Web Search Metrics

J.Jayanthi,
Associate Professor,
Sona College of Technology,
Salem -636005.

M.Ezhilmathi
M.E–Software Engineering,
Sona College of Technology
Salem –636005.

S. Rathi, PhD.
Assistant Professor (SG),
Government College of
Technology,
Coimbatore.

ABSTRACT

Software metrics are the key performance indicators, using which the performance of a system can be assessed quantitatively. Metrics can also be applied for personalized web search which can be used to retrieve relevant results for each individual user depending on their unique profile. Although personalized search based on user profile has been under research for many years and various metrics have been proposed, it is still uncertain whether personalization is unswervingly effective on different queries for different user profiles. A framework for personalized search which retrieves result based on user profile has been presented in this paper. User profile is maintained in the form of preference network (PN). Further metrics for ranking the search results based on user profile is also proposed.

General terms
Web mining, Software metrics

Keywords
Personalization, user profile, preference network

1. INTRODUCTION
Many new disputes arise for web search with the increasing amount of information on the web. A conventional search engine returns same set of results when the same query is submitted by all users, irrespective of who submitted the query. For example, for the query “orange”, some users may be interested in documents dealing with “orange” as a “fruit”, some users may need document related to “orange software company”, and while some other may need information about “orange mobile phones”. As well, different users have utterly different information needs. Personalization is found to be a great solution to address all these problems since it can provide distinct search results depending on user profile and preference. Various personalization strategies, which include [4], [6], [7], [16], [17], [19], [20], [21] and [22] have been proposed. But they are far from optimal [1]. Main problem of current personalized search is that most proposed algorithms are applied homogeneously to all users and queries. The main stand is that all queries should not be handled in same way, because a single personalization algorithm might not be suitable for all queries and all users.

Each algorithm has its own pros and cons. For example, for the query “orange” topical-interest-based personalization may lead to better performance but may be ineffective for the query “free games online”. All relevant documents for query “free games online” are mostly classified into the same topic categories, and topical-interest-based personalization is futile in such cases. Also applying personalization techniques on certain queries may be totally ineffective. For example, on the query “orange” using personalization based on topical interests of users might give better performance for individual users than a regular web search. In contrast, for the query “Yahoo!”, which is a typical navigational query as defined by Broder [23] and Lee et al. [24], almost all users consistently select a link to Yahoo!’s homepage. Therefore, none of the personalization strategies can provide apparent benefits to the users as demonstrated by [1].
As a solution to these problems, an evaluation framework has been developed to predict the appropriate algorithm to be applied based on different criterion. In this paper strategies are provided to:

(1) gather and model user’s search history in the form of “preference network (PN)”,
(2) a rule engine deduce appropriate metrics and algorithms for each query and each user, and
(3) improve web search effectiveness by using these metrics and algorithms.

2. RELATED WORK

The content similarity between user profile and returned web pages can be used to re-rank search results. User profiles can be obtained explicitly [4], [5] or implicitly. Majority of user are reluctant to provide explicit feedback on search results and their interests, many works in the area of personalized search focus on how to automatically learn user preferences without direct participation of users [4], [6], [7], [8]. Dou et al. [1] developed an evaluation framework based on real query logs to enable large-scale evaluation of personalized search. They also evaluated five personalization algorithms and proposed new metric called click entropy [1]. WebMate [9] uses user profiles to refineuser queries, but no experimental results are given. Watson [9] refines queries using a local context, but does not learn the user profile. Inquirus 2 [11] uses users’ preferences to choose data sources and refine queries, but it does not have user profiles and requires the users to provide their preferences of categories. In addition, only four non-topical categories are included in Inquirus 2. The method in [4] learns users’ profiles from their surfing histories and re-ranks/filters documents returned by meta-search engine based on the profiles.
Several approaches represent user interests by using topical categories. In [4], [5], [12], [13], [14], and [15], a user profile is usually structured as a concept/topic hierarchy. User-issued queries and user-selected snippets/documents are categorized into concept hierarchies that are accumulated to generate a user profile. When the user issues a query, each of the returned snippets/documents is also classified. The documents

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.10, December 2012 – www.ijais.org

are re-ranked based upon how well the document categories match user interest profiles.

Some other personalized search approaches use lists of keywords to represent user interests. Sugiyama et al. [8] built user preferences as vectors of distinct terms and constructed them by aggregating past preferences, including both long-term and short-term preferences. Shen et al. [6] first used language modeling to mine contextual information from a short-term search history. Tan et al. [16] then used the method to mine context from a long-term search history. Teevan et al. [17] and Chirita et al. [18] exploit rich models of user interests, built from both search-related information and other information about the user, including documents and e-mails that the user has read and created. In the work of Liu et al. [2], [6], keywords are associated with categories, and thus, user profiles are represented by a hierarchical category tree based on keyword categories.

In this proposed approach, user profile is used to retrieve relevant results. User profile is maintained in the form of preference network (PN). Also, a rule engine that can automatically identify the type of metric and algorithm to be applied for a query and user is also developed.

3. PROPOSED SYSTEM

An evaluation framework which can automatically identify the type of metric and algorithm to be applied based on various criterions such as user profile and user search history is proposed. The architecture of proposed system is illustrated in Figure 1.

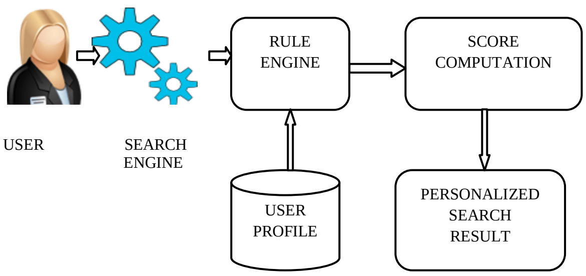

Figure 1 System Architecture

3.1 User Profile

User profile is maintained in the form of preference network (PN) [3]. Preference Network is constructed based on TF-IDF measure. TF-IDF measure is computed for each term in the top k documents retrieved by the web server for a query. The identical high scored terms are selected and the weights of each identical set of terms are summed up. From that list, again high weighted terms are selected to build the preference network.

The formula for Term Frequency is:

$$tf_i = \frac{n_i}{\sum_k n_k} \qquad (1)$$

n_i = Number of occurrences of a term *i*
n_k= Total number of terms in a document

The formula for Inverse Document Frequency is:

$$idf_i = log\frac{N}{df_i} \qquad (2)$$

N = Total number of relevant terms in the document
df_i= Number of documents that contain the term *i* at least once

Thus the TF-IDF weight is calculated using the formula:

$$TF - IDF\ weight = TF_i * IDF_i \qquad (3)$$

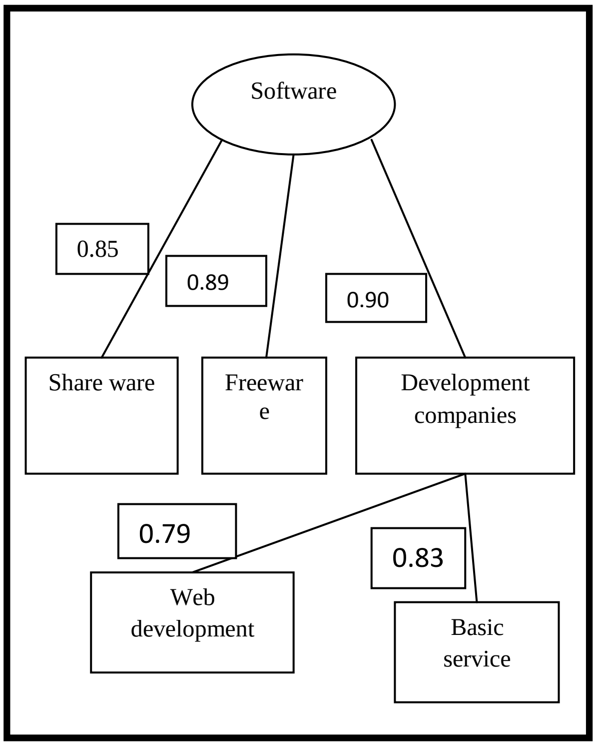

Figure 2Preference Network for a query

3.2 Rule Engine

A rule engine which identifies the convergence level of a user profile based on the structure and content of their preference network has been developed.

3.2.1. Profile Classification

It classifies user profile into three categories

(i) Converged profile
(ii) Semi-converged profile
(iii) Non-converged profile

3.2.1.1Converged Profile (CP)

A profile is said to be converged if same set of queries are repeated over a period of time. Say, a set of 5 queries are repeatedly given in 30 sessions observed over a period of 30 days. In such cases, user profile will contain very few preference networks (here, 5).

3.2.1.2 Semi-Converged Profile (SCP)

A profile is said to be semi-converged if it has equal number of repeated queries as well as new queries. Say, in 30 sessions, 10 queries are entirely new and a set of 4 queries are given repeatedly by the user in 5 sessions. The number of preference networks will be half the count of number of sessions considered (here, 14).

3.2.1.3 Non-Converged Profile (NCP)

A profile is not converged if the user gives entirely different query in each session. Say, 30 different queries in 30 sessions. So the number of preference networks for such users will be greater than or equal to number of sessions considered.

3.2.2 Query Classification

After the successful classification of user profile, the rule engine then classifies the given query into three types:

(i) Type-1: Self-Repeated Query (SRQ)
(ii) Type-2: Repeated Query (RQ)
(iii) Type-3: SRQ-RQ

3.2.2.1 Self-Repeated Query

When a user issues a query which is previously issued only by that user and which is not issued by any other user then it is a Self-Repeated query.

3.2.2.2 Repeated Query

If a query issued by a user is not that user's search history i.e. PN but in the PN of other users, then it is a repeated query.

3.2.2.3 SRQ-RQ

If a query issued by a user is in the PN of both the current user and other users, then it belongs to this type.

3.2.3. Ranking Search Results

For Type-1 queries, the documents are ranked in descending order of P-Click[1] scores of documentswhich were previously clicked by that user for the same query.

The formula for calculating P-Click score is:

$$P - Click_{Doc_n}(q,p,u) = \frac{|Clicks(q,p,u)|}{|Clicks(q,\blacksquare,u)| + \beta} \qquad (4)$$

|Clicks(q,p,u)| - number of clicks on web page*p* for the query *q* by the user *u*
|Clicks(q,■,u)| - total number of clicks for query *q* by*u*
β - smoothing factor.

For Type-2 and Type-3 queries, the documents are ranked in descending order of G-Score calculated using P-Click score of related documents from the profile of all the users who issued that query previously.

The formula for calculating G-Score is:

$$G - Score_{Doc_n} = \frac{\sum_{i=1}^{N}(P - Click_{Doc_n})_i}{N} \qquad (5)$$

$(P - Click_{Doc_n})_i$ – P-Click score of Doc_n of user *i*
N – Total number of user profiles which contains Doc_n

4. EXPERIMENTAL RESULTS

4.1 Sample Data Set

In this section, sample data set and score calculation has been presented. Here, the performance of the system for varying user profile sizes and different types of queries has been calculated. The performance of the system is also given in graphical representation.

Total number of queries	31
Number of users	6
Number of users with fully converged profile	3
Number of users with semi converged profile	2
Number of users with non-converged profile	1
Number of queries repeated by 6 users	None
Number of queries repeated by 5 users	1
Number of queries repeated by 4 users	2
Number of queries repeated by 3 users	11
Number of queries repeated by 2 users	10
Number of queries repeated by 1 user	7

Converged profile – Same set of queries are repeated over a period of time. Say, a set of 5 queries are repeatedly given in 30 sessions observed over a period of 30 days.

Semi-converged profile – Profile which contains a set of repeated queries and other set of queries being totally new. Say, in 30 sessions, 10 queries are entirely new and a set of 4 queries are given repeatedly by the user.

Non-converged profile – Profile in which each query is unique. None of the query is repeated.

Table 1 Profile of User 1 & 2 (Converged)

USER 1				USER 2			
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
Computer science	Doc 1,2,3	7,6,3	0.424242,0.36,0.18	Hardware	Doc16,17,19	11,9,8	0.39,0.31,0.28
Software	Doc6,4,5	9,7,5	0.4186,0.33,0.23	Software	Doc6,19,20	8,6,5	0.41,0.30,0.25
Operating System	Doc7,9,8	10,9,3	0.44444,0.4,0.13	Graphics	Doc21,22,23	7,4,2	0.52,0.30,0.14
Algorithms	Doc11,10,12	9,6,2	0.51429,0.34,0.11	Algorithms	Doc11,10,24	8,5,4	0.46,0.29,0.23
AI	Doc15,14,13	8,7,4	0.412026,0.36,0.21	Programming	Doc25,26,27	12,10,7	0.4,0.33,0.24

Table 2 Profile of User3(Converged) and User4(Semi converged)

USER 3				USER 4			
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
Algorithms	Doc 11,10,28	9,7,3	0.46154,0.35897, 0.15385	Security	Doc 39,40,41	11,7,9	0.4,0.25455, 0.32727
Software	Doc 6,4,29	9,8,5	0.4,0.35556, 0.22222	IDS	Doc 42,43,44	8,6,5	0.41026,0.30769, 0.25641
HCI	Doc 30,31,32	9,6,4	0.46154, 0.30769, 0.20513	Internet	Doc 45,46,47	11,10,8	0.37288,0.33898, 0.27119
Data communication	Doc 33,34,35	9,5,3	0.51429, 0.28571, 0.17143	Data communication	Doc 33,34,48	8,7,5	0.39024,0.34146, 0.24390
Mobile Computing	Doc 36,37,38	8,6,4	0.43243, 0.32432,0.21622	Mobile computing	Doc 36,38,49	4,3,2	0.42105,0.31579, 0.21053

Table 3 Profile of User 4 cont'd

USER 4							
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
HCI	Doc 30,31,50	3,2,1	0.46154,0.30769,0 .15385	Algorithm	Doc 11,10,63	2,2,1	0.36364,0.36364, 0.18182
Malicious software	Doc 51,52,53	2,1,1	0.44444,0.22222, 0.22222	AI	Doc 15,14,64	2,1,1	0.44444,0.22222, 0.22222
VPN	Doc 54,55,56	4,2,1	0.53333,0.26667, 0.13333	Hacking	Doc 65,66,67	2,1,1	0.44444,0.22222, 0.22222
Cryptography	Doc 57,58,59	3,1,1	0.54545,0.18182, 0.18182	Firewall	Doc 68,69,70	2,1,1	0.44444,0.22222, 0.22222
Biometrics	Doc 60,61,62	2,1,1	0.44444,0.22222, 0.22222				

Table 4 Profile of User 5 (semi converged)

USER 5							
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
VPN	Doc 54,55,71	9,8,5	0.4,0.35556, 0.22222	Malicious software	Doc 51,82,83	2,1,1	0.44444,0.22222, 0.22222
Authentication	Doc 72,73,74	10,8,7	0.39216,0.31373, 0.27451	Cryptography	Doc 57,58,84	3,1,1	0.54545,0.18182, 0.18182
Honeypots	Doc 75,76,77	8,7,5	0.39024,0.34146, 0.2439	Biometrics	Doc 60,61	3,1	0.66667,0.22222
Mac OS	Doc 78,79,80	10,9,8	0.36364,0.32727, 0.29091	Internet	Doc 45,46,85	3,1,1	0.54545,0.18182, 0.18182
Firewall	Doc 68,69,81	3,1,1	0.54545,0.18182, 0.18182	Graphics	Doc 21,23,86	3,2,1	0.46154,0.30769, 0.15385

Table 5 Profile of User 5 (semi converged) and User 6 (Non converged)

USER 5				USER 6			
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
Computer science	Doc 1,2,3	3,1,1	0.54545,0.18182, 0.18182	Cryptography	Doc 57,58,84	3,2,2	0.4,0.26667, 0.26667
Data communication	Doc 33,34,38	2,1,1	0.44444,0.22222, 0.22222	Firewall	Doc 68,69,81	1,2,2	0.18182, 0.36364, 0.36364
Java	Doc 87,88,89	2,1,2	0.36364,0.18182, 0.36364	Java	Doc 87,88,93	2,1,1	0.44444,0.22222, 0.22222
MIS	Doc 90,91,92	2,2,1	0.36364,0.36364, 0.18182	MIS	Doc 90,91,94	2,2,1	0.36364,0.36364, 0.18182

Table 6 Profile of User 6

USER 6							
Query	**Document**	**Number of clicks**	**P-Click**	**Query**	**Document**	**Number of clicks**	**P-Click**
Graphics	Doc 21,22,86	2,1,1	0.44444,0.22222, 0.22222	Mobile Computing	Doc 36,37,49	2,1,2	0.36364, 0.18182,0.36364
Computer Science	Doc 1,2,3	2,1,1	0.44444,0.22222, 0.22222	VPN	Doc 54,55,71	1,2,1	0.22222,0.44444, 0.22222
Algorithms	Doc 10,11,28	2,2,1	0.36364,0.36364, 0.18182	Malicious software	Doc 51,53,83	2,1,1	0.44444,0.22222, 0.22222
Hardware	Doc 16,18,95	2,1,1	0.44444,0.22222, 0.22222	Authentication	Doc 74,72,73	2,1,1	0.44444,0.22222, 0.22222
Software	Doc 6,4,19	3,1,1	0.54545,0.18182, 0.18182	Security	Doc 39,41,40	2,1,1	0.44444,0.22222, 0.22222
AI	Doc 14,15,13	1,1,1	0.28571,0.28571, 0.28571	Hacking	Doc 65,66,67	1,1,1	0.28571,0.28571, 0.28571
OS	Doc 7,8,96	2,1,1	0.44444,0.22222, 0.22222	Mac OS	Doc 78,79,80	2,1,2	0.36364, 0.18182,0.36364
Programming	Doc 25,26,97	2,1,1	0.44444,0.22222, 0.22222	Companies	Doc 98,99,100	1,1,1	0.28571,0.28571, 0.28571
IDS	Doc 42,44,43	2,2,1	0.36364,0.36364, 0.18182	Chats	Doc 101,102, 103	2,2,1	0.36364,0.36364, 0.36364
Internet	Doc 45,47,85	3,1,1	0.54545,0.18182, 0.18182	Forum	Doc 104,105, 106	3,1,2	0.46154,0.15385, 0.30769
HCI	Doc 30,31,32	3,1,2	0.46154, 0.15385, 0.30769	Super computing	Doc 107,108, 109	1,2,1	0.22222,0.44444, 0.22222
Biometrics	Doc 60,62,61	1,2,1	0.22222,0.44444, 0.22222	Open source	Doc 110,111, 112	2,1,1	0.44444,0.22222, 0.22222
Data communication	Doc 33,34,48	2,1,1	0.44444,0.44444, 0.22222	Soft computing	Doc 113,114, 115	3,1,1	0.54545,0.18182, 0.18182

4.2 Sample Calculation

Type-1: When the query “super computing” is issued by User 6 it belongs to type-1, since in training set the query is issued only by User 6 and not by any other users. So the documents in search result are ranked in the order of P-Click score calculated based on user profile.

Doc113

Doc114

Doc115

Type-2: When the query “software” is issued by User 4 it belongs to type, since the query is not issued by user 4 but issued by other users namely user1, user2, user3 and user6. So for such type of queries G-Score (Group score) must be calculated. The average score for each document is calculated based on P-Click score for each user document and the documents are ranked in descending order of scores. The relevant documents for this query are Doc4, Doc5, Doc6, Doc19, Doc20 and Doc29. The documents repeated by many users are Doc6, Doc4 and Doc19.The G-score of each page can be calculated using the formula:

$$GScore_{Doc_n} = \frac{\sum_{i=1}^{N} PClick_i}{N}$$

N – Number of users who clicked the document *n* for that query

P-Click$_i$– Personalized score of each user for that document.

$$GScore_{Doc_6} = \frac{0.4186+0{,}41026+0.4+0.54545}{4}$$

$$GScore_{Doc_6} = 0.44358$$

$$GScore_{Doc_4} = 0.28765$$

$$GScore_{Doc_{19}} = 0.24476$$

$$GScore_{Doc_{20}} = 0.25641$$

$$GScore_{Doc_{29}} = 0.22222$$

The search list is ranked in descending order of G-Score:

Doc6

Doc4

Doc20

Doc19

Doc29

Type-3: when the query "algorithm" is issued by user1 it belongs to type3, since it is issued by user1 as well as user2, user3, user4 and user6. The relevant documents are Doc 10, Doc11, Doc 12, Doc24, Doc28 and Doc68. Calculating G-Score for each document by using the formula above, the documents are ranked in descending order of scores.

4.3 Performance of the System

The following figures show the performance of the proposed system for various user profile size.X-axis represents the user profile size and y-axis represents the precision value of the result set retrieved by the system for the query issued by the user. The formula to calculate precision is:

$$Precision = \frac{number\ of\ relevant\ items}{number\ of\ retrieved\ items} \quad (6)$$

According to the data set considered, value 5 in x-axis represents converged profile, value 15 represents semi-converged profile and value 30 represents non-converged profile.

Fig3 shows the performance of the system for various user profile size for the queries that are self-repeated by the users. This graph shows that when the queries issued by the users belong to self-repeated category, then precision of the system is high proving that the results retrieved by the system based on the user profiles have satisfied the users.

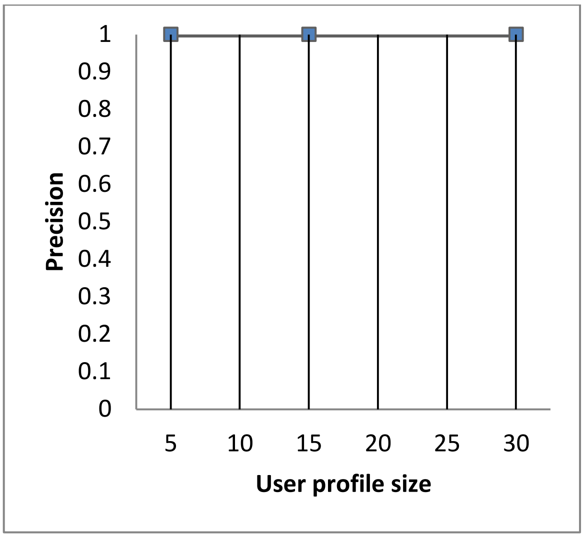

Figure 3 Performance of the system for SRQ

Fig4 shows the performance of the system for repeated queries. Performance of the system for RQ is not as high as the performance for SRQ but it exhibits a reasonable performance that satisfies the user to a certain extent. If the user profiles are fully converged then the system can give high performance for RQ also.

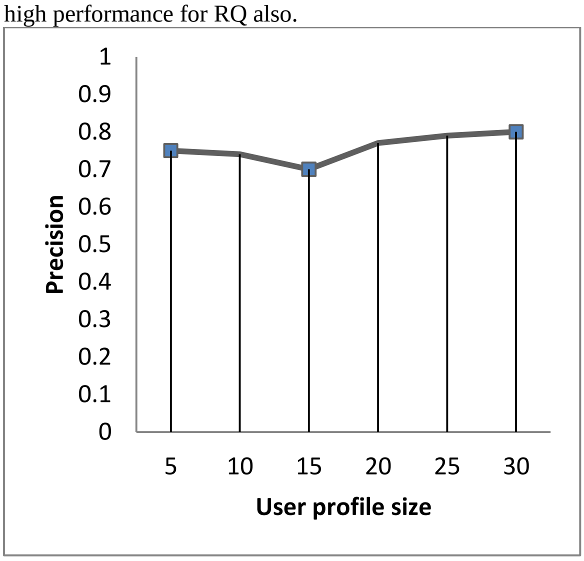

Figure 4 Performance of the system for RQ

Fig5 shows the performance of the system for SRQ-RQ i.e., type 3 queries. For such type of queries, the system exhibits high performance and satisfies the user completely. Hence using the user profile and search history of various users helps in retrieving more relevant results for various users with varying user profile sizes.

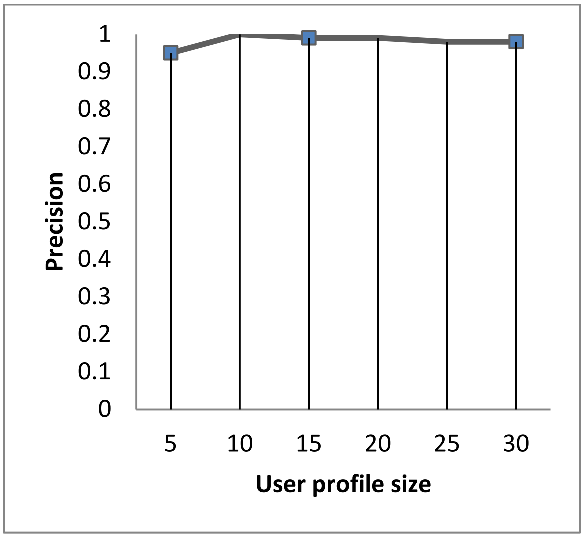

Figure 5 Performance of the system for SRQ-RQ

5. CONCLUSION

In this paper, an evaluation framework for automatic identification of metrics and algorithms to be applied for retrieving relevant web search results for individual users has been proposed. User profile is maintained in the form of Preference Networks (PN). Further techniques and strategies for classifying user profiles and queriesis also proposed. This approach would be useful to improve search accuracy and for retrieving relevant results for each individual user depending on their preference. Future work can be extended in proposing metrics for entirely new queries which is not issued by any of the users in the data set.

6. REFERENCE

[1] Z. Dou, R. Song, J.-R. Wen and X. Yuan, “Evaluating the Effectiveness of Personalized Web Search”, IEEE Trans. Knowledge and Data Eng., vol. 21, no. 8, pp. 1178-1190, August 2009.

[2] F. Liu, C. Yu, and W. Meng, “Personalized Web Search forImproving Retrieval Effectiveness,” IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 28-40, Jan. 2004.

[3] J. Jayanthi, B. Keerthana and K. S. Jayakumar, “Building User Profile Based on Preference Networks”, International Journal of Web Engineering, pp. 11-14, 2012.

[4] A. Pretschner and S. Gauch, “Ontology Based PersonalizedSearch,” Proc. 11th IEEE Int’l Conf. Tools with Artificial Intelligence (ICTAI ’99), pp. 391-398, 1999.

[5] P.-A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschutter, “UsingODP Metadata to Personalize Search,” Proc. 28th Ann. Int’lACM SIGIR Conf. Research and Development in InformationRetrieval (SIGIR ’05), pp. 178-185, 2005.

[6] X. Shen, B. Tan, and C. Zhai, “Implicit User Modeling forPersonalized Search,” Proc. ACM Int’l Conf. Information andKnowledge Management (CIKM ’05), pp. 824-831, 2005.

[7] F. Liu, C. Yu, and W. Meng, “Personalized Web Search byMapping User Queries to Categories,” Proc. ACM Int’l Conf.Information and Knowledge Management (CIKM ’02), pp. 558-565, 2002.

[8] K. Sugiyama, K. Hatano, and M. Yoshikawa, “AdaptiveWebSearch Based on User Profile Constructed without Any Effort from Users,” Proceedings of 13th Int’l World Wide Web Conference (WWW ’04), pp. 675-684, 2004.

[9] L. Chen and K. Sycara, “WebMate: A Personal Agent for Browsing and Searching,” Proc. Second Int’l Conf. Autonomous Agents and Multi Agent Systems, pp. 132-139, 1998.

[10] J. Budzik and K. Hammond, “Watson: Anticipating and Contextualizing Information Needs,” Proc. 62nd Ann. Meeting Am. Soc. Information Science, 1999.

[11] E.J. Glover, G.W. Flake, S. Lawrence, W.P. Birmingham, A. Kruger, C.L. Giles, and D.M. Pennock, “Improving Category Specific Web Search by Learning Query Modifications,” SAINT, pp. 23-34, 2001.

[12] J. Chaffee and S. Gauch, “Personal Ontologies for Web Navigation,” Proc. ACM Int’l Conf. Information and Knowledge Management (CIKM ’00), pp. 227-234, 2000.

[13] S. Gauch, J. Chaffee, and A. Pretschner, “Ontology-BasedPersonalized Search and Browsing,” Web Intelligence and Agent Systems, vol. 1, no. 3/4, pp. 219-234, 2003.

[14] J. Trajkova and S. Gauch, “Improving Ontology-Based User Profiles,” Proc. Recherched’InformationAssistee par Ordinateur (RIAO ’04), pp. 380-389, 2004.

[15] M. Speretta and S. Gauch, “Personalized Search Based on User Search Histories,” Proc. IEEE/WIC/ACM Int’l Conf. Web Intelligence (WI ’05), pp. 622-628, 2005.

[16] B. Tan, X. Shen, and C. Zhai, “Mining Long-Term Search History to Improve Search Accuracy,” Proc. 12fth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD ’06), pp. 718-723, 2006.

[17] J. Teevan, S.T. Dumais, and E. Horvitz, “Personalizing Search via Automated Analysis of Interests and Activities,” Proc. 28th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR ’05), pp. 449-456, 2005.

[18] P.A. Chirita, C. Firan, and W. Nejdl, “Summarizing Local Context to Personalize Global Web Search,” Proc. ACM Int’l Conf. Information and Knowledge Management (CIKM), 2006.

[19] J. Pitkow, H. Schutze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, “Personalized Search,” Comm. ACM, vol. 45, no. 9, pp. 50-55, 2002.

[20] G. Jeh and J. Widom, “Scaling Personalized Web Search,” Proc. 12th Int’l World Wide Web Conf. (WWW ’03), pp. 271-279, 2003.

[21] P. Ferragina and A. Gulli, “A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering,” Special Interest Tracks and Posters of the 14th Int’l Conf. World Wide Web (WWW ’05), pp. 801-810, 2005.

[22] J.-T. Sun, H-J.Zeng, H. Liu, Y. Lu, and Z. Chen, "CubeSVD: A Novel Approach to Personalized Web Search," Proc. 14th Int'l World Wide Web Conf. (WWW '05), pp. 382-390, 2005.

[23] A. Broder, "A Taxonomy of Web Search," ACM SIGIR Forum, vol. 36, no. 2, pp. 3-10, 2002.

[24] U. Lee, Z. Liu, and J. Cho, "Automatic Identification of User Goals in Web Search," Proc. 14th Int'l World Wide Web Conf. (WWW '05), pp. 391-400, 2005.