Toward a Web Search Personalization Approach Based on Temporal Context

Djalila Boughareb and Nadir Farah

Computer science department
Annaba University, Algeria
{boughareb,farah}@labged.net

Abstract. In this paper, we describe the work done in the Web search personalization field. The proposed approach purpose is the understanding and identifying the user search needs using some information sources such as the search history and the search context focusing on temporal factor. These informations consist mainly of the day and the time of day. Considering such data, how can it improve the relevance of search results? That's what we focus on it in this work; The experimental results are promising and suggest that taking into account the day, the time of the query submission in addition to the pages recently been examined can be a viable context data for identifying the user search needs and furthermore enhancing the relevance of the search results.

Keywords: Personalized Web search, Web Usage Mining, temporal context and query expansion.

1 Introduction

The main feature of the World Wide Web is not that it allowed making available billions byte of information, but mostly that it has brought millions of users to make of the information search a daily task. In that task, the information retrieval tools are generally the only mediators between a search need and its partial or total satisfaction.

A wide variety of researches have improved the relevance of the results provided by the information retrieval tools. However, the explosion in the volume of the information available on the Web, which is measured at least 2.73 billion pages according to a recent statistics[1] made in December 2010; the low expression of the user query reflected in the fact that the users usually employ a few numbers of keywords to describe their needs average 2.9 words [7], for example, a user who's looking to purchase a *bigfoot 4x4* vehicle submits the query "*bigfoot*" to *AltaVista*[2] search engine will obtain among the ten most relevant documents, one document on football, five about animals, one about a production company and three about the chief of the *Miniconjou Lakota Sioux* and zero document about *4x4 vehicle*, but if we add the keyword "*vehicle*", all first documents returned by the search engine will be about vehicles, and will satisfy the user information needs; moreover, the reduced understanding of the user needs engender the low relevance of the retrieval results and its bad ranking.

[1] http://www.worldwidewebsize.com/
[2] http://fr.altavista.com/

H. Cherifi, J.M. Zain, and E. El-Qawasmeh (Eds.): DICTAP 2011, Part I, CCIS 166, pp. 33–44, 2011.

In order to overcome these problems, the information personalization has emerged as a promising field of research which can be defined as the application of data mining and machine learning techniques to build models of user behavior that can be applied to the task of predicting user needs and adapting future interactions with the ultimate goal of improved user satisfaction [1].

The purpose of this work is to develop a system prototype, which is able to both automatically identify the user information needs and retrieve relevant contents without requiring any action by the user. To do this, we have proposed: A user profiling approach to build user profiles or user models through some of information sources which can be extracted from the search history of the users using Web usage mining techniques. We have mainly taken into consideration temporal context in order to investigate the effectiveness of the time factor in understanding and identifying the search needs of the user, based on the heuristic that user browsing behavior changes according to the day and the time of query submission.

Indeed, we have observed that the browsing behavior changes according to the day and the time of day, *i.e.* the user browsing behavior during workdays are not the same as weekends for example. Driven by the browsing behaviors observation of 30 users during one month from January 01, 2010 to January 30, 2010, we have found that their search behavior varies according to the day and the hour, for example 12 surfers on average conducted research about sport field on Wednesday evening from 6pm and 13 on Thursday morning, nevertheless 14 surfers on average conducted research on their study domain on Monday afternoon between 2 pm and 7 pm. Generally, the searches have been focused on leisure websites on Saturday. Moreover, we developed a query expansion approach to resolve the short query problem based on the building models.

The remainder of this paper is organized as follows. Before describing the proposed approach in section 3, we present a state of the art in section 2. Section 4 presents the experiments and we discuss obtained results in section 5. Section 6 concludes the paper and outlines areas for future research.

2 State of the Art

In the large domain of the personalization, user modeling represents the main task. Indeed, a personalization system creates user profiles *a priori* and employs them to improve the quality of search responses [8], of provided web services [11, 14] or of web site design [2]. User modeling process can be divided into two main steps, data collection and profiles construction. Data collection consists of collecting relevant information about the users necessary to build user profiles; the information collected (age, gender, marital status, job…etc) may be:

-Explicitly inputted by the user via HTML forms and explicit feedback [14, 15] but due to the extra time and effort required from users this approach is not always fitting;

-Implicitly, in this case the user information's may be inferred from his/her browsing activity [4], from browsing history [19] and more recently from his/her search history [17], that contains information about the queries submitted by a particular user and the dates and times of those queries.

In order to improve the quality of data collected and thereafter the building models, some of researches combine explicit and implicit modeling approach, Quiroga and Mostafa [12] researches show that profiles built using the combination of explicit and implicit feedback improve the relevance of the results returned by their search systems, in fact they obtained 63% precision using explicit feedback alone, and 58% of precision using implicit feedback alone. Nevertheless, by the combination of the two approaches an approximately of 68% of precision was achieved. However, white [21] proves that there are no significant differences between profiles constructed using implicit and explicit feedback.

The profiles construction consist the second step of the user profiling process, it has as purpose to build the profiles from the collected data set based on machine learning algorithms like genetic algorithms [22], neural networks [10, 11], Bayesian networks [5] … etc.

The employment of Web usage mining process (WUM) represents one of the main useful tools for user modeling in the field of Web search personalization, which has been used to analyze data collected about the search behavior of the users on the Web to extract useful knowledge. According to the final goal and the type of the application, researchers tempt to most exploit the search behavior such as a valuable source of knowledge.

Most existing web search personalization approaches are based mainly on search history and browsing history to build a user models or to expand the user queries. However, very little research effort has been focused on the temporal factor and its impact on the improvement of the web search results. In their work [9] Lingras and West proposed an adaptation of the K-means algorithm to develop interval clusters of web visitors using rough set theory. To identify the user behaviors, they were based on the number of web accesses, types of documents downloaded, and time of day (they divided the navigation time into two parts, day visit and night visit) but this presented a reduced accuracy of user's preferences over time.

Motivated by the idea that more accurate semantic similarity values between queries can be obtained by taking into account the timestamps in the log, Zhao et al. [23] proposed a time-dependent query similarity model by studying the temporal information associated with the query terms of the click-through data. The basic idea of this work is taking temporal information into consideration when modeling the query similarity for query expansion. They obtained more accurate results than the existing approaches which can be used for improving the personalized search experience.

3 Proposed Approach

The ideas presented in this paper are based on the observations cited above that the browsing behavior of the user changes according to the day and the hour. Indeed, it is obvious that the information needs of the user changes according to several factors known as the search context such as date, location, history of interaction and the current task. However, it may often maintain a pace well determined. For example, a majority of people visit the news each morning. In summary, the contribution of this work can be presented through the following points:

1. Exploiting temporal data (day and time of day) in addition to the pages recently been examined to identify the real search needs of the user motivated by the observed user browsing behavior and the following heuristics:

   - The user search behavior changes according to the day, i.e. during workdays the user browsing behavior is not the same as weekends for example surfers conducted research about leisure on Saturday;
   - The user search behavior changes according to the time of day and it may often maintain a well determined pace, for example a majority of people visit the news web sites each morning.
   - The information heavily searched in the last few instructions will probably be heavily searched again in the next few ones. Indeed, nearly 60% of users conducts more than one information retrieval search for the same information problem [20].

2. Exploiting temporal data (time spent in a web page) in addition to click through data to measure the relevance of web pages and to better rank the search results.

To do this, we have implemented a system prototype using a modular architecture. Each user access the search system home page is assigned a session ID, in which all the user navigation activities are recorded in a log file by the log-processing module. When the user submits an interrogation query to the system, the encoding module creates a vector of positive integers composed from the submitted query and information corresponding to the current research context (the day, the time of query submission and domain recently being examined). The created vector will be submitted to the class finder module. Based on the neural network models previously trained and embedded in a dynamically generated *Java* page the class finder module aims to catch the profile class of the current user. The results of this operation are supplied to the query expansion module for reformulating the original query based on the information included in the correspondent profile class. The research module's role is the execution of queries and results ranking based always on the information included in the profile class. In the following sections we describe in detail this approach, the experiments and the obtained results.

3.1 Building the User Profiles

A variety of artificial intelligence techniques have been used for user profiling, the most popular is Web Usage Mining which consists in applying data mining methods to access log files. These files which collect the information about the browsing history, including client IP address, query date/time, page requested, HTTP code, bytes served, user agent, and referrer, can be considered as the principal data sources in the WUM based personalization field.

To build the user profiles we have applied the mainly three steps in WUM process namely [3]: preprocessing, pattern discovery and pattern analysis to the access log files resulted from the Web server of the Computer Science department at *Annaba University* from January 01, 2009 to June 30, 2009, in the following sections we will focus on the first two steps.

3.1.1 Preprocessing

It involves two main steps are: first, the data cleaning which aims for filtering out irrelevant and noisy data from the log file, the removed data correspond to the records of graphics, videos and format information and the records with failed HTTP status codes;

Second, the data transformation which aims to transform the data set resulted from the previous step into an exploitable format for mining. In our case, after elimination the graphics and the multimedia file requests, the script requests and the crawler visits, we have reduced the number of requests from 26 084 to 17 040, i.e. 64% of the initial size and 10 323 user sessions of 30 minutes each one. We have been interested then in interrogation queries to retrieve keywords from the URL parameters (Fig. 1).

As the majority of users started their search queries from their own machines the problem of identifying users and sessions was not asked.

```
10.0.0.1 - - [16/Jan/2009:15:01:02 -0500] "GET /assignment-3.html HTTP/1.1" 200 8090
http://www.google.com/search?=course+of+data+mining&spell=1 Mozilla/4.0 (compatible; MSIE 6.0; NT 5.1;
SV1)"Windows
```

Fig. 1. An interrogation query resulting from the log file

3.1.2 Data Mining

In this stage, data mining techniques was applied to the data set resulted from the previous step. In order to build the user profiles we have brought the users who have conducted a search on a field *F*, in the Day *D* during the time interval *T* in the same profile class *C*, i.e., for this we have made a supervised learning based on artificial neural networks. Indeed, if we have proceeded to an unsupervised learning, we may be got a very disturbing number of classes, which do not allow us to achieve the desired goal of this approach, nor to test its effectiveness.

The edited network is an MLP (Multi Layer Perceptron) with a two hidden layers. The data encoding process was made as follows. An input vector $X\{[0,1]\}^p$with $p = 12$ is propagated from the input layer of four nodes to the output layer of eight nodes corresponding to the number of profile classes created, through two hidden layers (with 14, 12 nodes respectively). The input vector composed of four variables namely: the query, the day, the time of day and the domain recently being examined.

1. The query (q): we analyzed the submitted query based mainly on a keywords descriptor to find the domain targeted by the query; in our case we have created 4 vectors of terms for fields (computer science, sport, leisure and news). This analysis helps the system to estimate the domain targeted by the query. Other information can be useful to find the domain targeted by the query such as the type of the asked documents (e.g. if the user indicates that he is looking for pdf documents, this can promote computer science category. However, if the query contains the word video, it promotes the leisure category);
2. The day (d): The values that take the variable "day" correspond to the 7 days of the week.

3. The time of day (t): we divided the day into four browsing time: the morning (6:00 am to 11:59 am), the afternoon (noon to 3:59 pm), the evening (2:00 pm to 9:59 pm) and night (10:00 pm to 5:59 am).
4. The domain recently being examined (Rp): if that is the first user query this variable will take the same value of the variable query (q), otherwise the domain recently being examined will be determined by calculating similarity between the vector of the Web page and the 4 predefined descriptors of categories that contain the most common words in each domain, the vector page is obtained by *tf.idf* weighting scheme (the term frequency/inverse document frequency) described in the equation (1) [13].

$$\mathrm{tf.idf} = \frac{\mathrm{N}}{\mathrm{T}} * \log \frac{\mathrm{D}}{\mathrm{DF}} \quad (1)$$

Where N is the number of times a word appears in a document, T is the total number of words in the same document, D is the total number of documents in a corpus and DF is the number of document in which a particular word is found.

3.2 User Profiles Representation

The created user profiles are represented through a weighted keyword vector, a set of queries and the examined search results; a page relevance measure has been employed to calculate the relevance of each page to her correspondent query.

Each profile class (pc_i) is described through an n-dimensional weighted keyword vector $V_i = \langle (k_1, w_1), (k_2, w_2), \dots \dots (k_n, w_3) \rangle$ and a set of queries, each query q_j is represented as an ordered vector of relevant pages to it. $q_j = \langle p_1, p_2, \dots . p_n \rangle$, where the relevance of a page $p_i$to the query q_j can be obtained based on the click-through data analysis by the following measure described in the equation (2). Grouping the results of the previous queries and assign them a weighing aims to enhance the relevance of the top first retrieved pages and better rank the system results. Indeed, information such as time spent on a page and the number of clicks inside, can help to determine the relevance of a page to a query and to all similar queries to it, this in order to better rank the returned results.

$$R(p_i, q_j) = \frac{T_v(p_i, q_j).\ NC}{\sum_{k=1}^{n} T_v(p_k, q_j)} \quad (2)$$

Here $T_v(p_i, q_j)$ measure the time that page p_i has been visited by the user who issued the query q_j, NC measure the number of clicks inside page p_i by the user who issued the query q_j and $\sum_{k=1}^{n} T_v(p_k, q_j)$ refers to the total number of times that all pages have been visited by the user who issued the query q_j.

3.3 Profiles Detection

This module tries to infer the current user profile by analyzing keywords describing his information needs and taking into account information corresponding to the current research context particularly the day, the time of query submission and information recently been examined to assign the current user to the appropriate profile class. To do this, the profiles detection module create a vector of positive integers composed from the submitted query and information corresponding to the current research context (the day, the query submission hour and domain recently being examined), the basic idea is that information heavily searched in the last few instructions will probably be heavily searched again in the next few ones. Indeed, in theme researches Spink et al. [18] show that nearly 60% of users had conducted more than one information retrieval search for the same information problem.

The created vector will be submitted to the neural network previously trained and embedded in a dynamically generated Java page in order to assign the current user to the appropriate profile class.

3.4 Query Reformulation

In order to reformulate the submitted query, the query reformulation module makes an expansion of that one with keywords resulting from similar queries to it to obtain a new query closer to the real need of the user and to bring back larger and better targeted results. The keywords used for expansion are derived from past queries which have a significant similarity with the current query, the basic hypothesis is that the top documents retrieved by a query are themselves the top documents retrieved by the past similar queries [20].

3.4.1 Query Similarity

Exploiting the past similar queries to extend the user query consists one of the most known methods in automatic query expansion field [6, 16]. We have based on this method to extend the user query. To do this, we have represented each query as a weighted keywords vector using *tf.idf* weighting scheme. We have employed the cosine similarity described in the equation (3) to measure the similarity $Sim(q_i, q_j)$ between queries. If a significant similarity between the submitted query and a past query is found, this one will be assigned to the query set Q_s, the purpose is to gather from the current profile class all queries whose exceed a given similarity threshold £ and employing them to extend the current submitted query.

$$Sim(q_i, q_j) = \frac{q_i . q_j}{\|q_i\| \|q_j\|} \quad (3)$$

3.4.2 Query Expansion

As we have mentioned above, one of the most known problems in information retrieval is the low query expression reflected in the use of short queries. As a solution has been proposed to this problem, the query expansion which aims to support the user in his/her searches task through adding search keywords to a user query in order to disambiguate it and to increase the number of relevant documents retrieved. We have employed the first 10 keywords resulted from the most 5 similar queries to rewrite the original query q_0 ;

The weight of an added term P_t is obtained by averaging the weight of this term in Q_s queries where it appears.

$$P_t = \frac{\sum_i p_t}{k} \tag{4}$$

Where $\sum_i p_t$ is the sum of the weights of term t in k queries in Q_s where it appears k is the total number of queries containing the term t.

3.5 The Matching

In order to enhance the relevance of the top first retrieved pages and better rank results, we propose to include additional information like the page access frequency from previous queries results from similar queries. This can help to assign more accurate scores to the pages jugged relevance by the users having conducted a similar search queries. Based on the set of queries Q_s obtained in the previous step and contained all queries which have a significant similarity with the current one, we have defined a matching function described in the equation (5) as follow:

$$Match(q_i, p_j) = Sim(q_i, p_j) + nrank(p_j, Q_s) \tag{5}$$

$$nrank(p_j, Q_s) = \frac{\sum_k \sum_j R(p_j, q_k)}{\sum_i p_i \sum_q q_k} \tag{6}$$

Where $Sim(q_i, p_j)$ measure the cosine similarity between the page p_j vector and the query q_i vector, $nrank(p_j, Q_s)$ which is described in the equation (5) measures the average relevance of a page p_j in the query set Q_s based on the average time in which a page p_j has been accessed and the number of clicks inside compared with all others pages $(\sum_i p_i)$ resulted from all others similar queries $(\sum_q q_k)$. The $R(p_j, q_k)$ measure of the relevance of a page p_j to the query q_k have been defined above in the equation (2).

4 Experiments

We developed a Web-based Java prototype that provides an experimental validation of the neural network models. On the one hand, we mainly aimed to checking the ability of the produced models in catching the user profile according to: his/her query category, day, the query submission time and the domain recently being examined can be defined from pages recently visited, for this a vector of 4 values between] 0, 1] will be submitted to the neural network previously edited by *joone*[3] library, trained and embedded in a dynamically generated *Java* page.

The data set was divided into two separate sets including a training set and a test set. The training set consists of 745 vectors were used to build the user models while the test set which contains 250 vectors were used to evaluate the effectiveness of the user models. Results are presented in the following section.

[3] `http://sourceforge.net/projects/joone/`

The quality of an information search system may be measured by comparing the responses of the system with the ideal responses that the user expects to receive, based on two metrics commonly used in information retrieval are *recall* and *precision*. *Recall* measures the ability of a retrieval system to locate relevant documents in its index and *precision* measures its ability to not rank irrelevant documents.

In order to evaluate the user models and analyzing how the results quality can be influenced by the setting of the parameters involved in the user profiles. We have used a collection of 9 542 documents indexed by the *Lucene*[4] indexing API and we have been measuring the effectiveness of the implemented system in terms of *Top-n recall* and *Top-n precision* defined in the equations (7) and (8) respectively. For example, at n = 50, the top 50 search results are taken into consideration in measuring recall and precision. The obtained results are represented in the following section.

$$Top - n\ Recall = \frac{R_n}{M} \quad (7)$$

$$Top - n\ Precision = \frac{R_n}{N} \quad (8)$$

Where R_n represents the number of documents retrieved and relevant within n, M refers to the total number of relevant documents and N refer to the total number of documents retrieved.

5 Results and Discussion

Once the user models are generated, it is possible to carry out real tests as follows, we employed 15 users who build queries an average of 10 for each profile class. The experiments showed that over 80 submissions we obtain 6 errors of classification, i.e. 7,5%, we introduce the example of Profile class (pc_2) characterized by computer science students interested with leisure, (pc_7) characterized by users interested with leisure and (pc_8) characterized by users interested with music and videos, 1 vector from (pc_2) is classified in (pc_8) and 2 vectors are classified in (pc_7) that we don't consider this a classification error because profiles class can chair some characteristics and students browsing behavior will be similar than any other users browsing behavior over his scientific search.

Thereafter, in order to evaluate the expansion approach based on keywords involved from profile class caught, we tested the expansion of 54 queries and we obtain 48 good expansions, i.e. 88%. Taking the example of the query $q_0 = \langle programming, java \rangle$ submitted by a student who is recently examining a database course, in this period students in information and database system option were interested in a tutorial using Oracle framework. After reformulation step a new

[4] http://lucene.apache.org/java/docs/index.html

query $q' = \langle programming, java, Jbuilder, Oracle \rangle$ has been obtained. Another example the query $q_0 = \langle apple \rangle$after the expansion step, the system returns the query $q' = \langle apple, computer, Mac \rangle$ this because the recently examined pages were about computer science domain.

After analyzing user's judgments we observed that almost 76% of users were satisfied with the results provided by the system. The average Top-n recall and Top-n precision for 54 queries are represented in the following diagrams which show a comparison of the relevance of the Web Personalized Search System (*WePSSy*) results with *AltaVista, Excite* and *Google* search engine results.

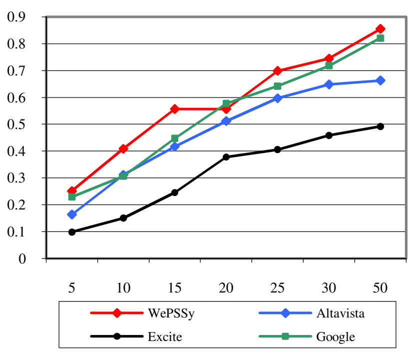

Fig. 2. Top-n recall (comparison of results obtained by the *WePSSy* system with *AltaVista, Excite* and *Google* search engine results)

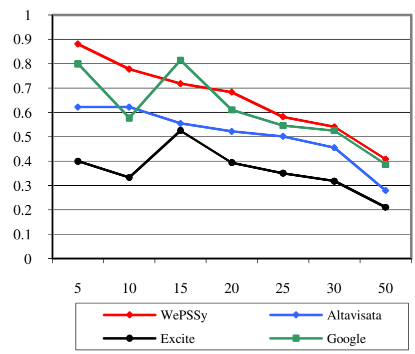

Fig. 3. Top-n precision (comparison of results obtained by the *WePSSy* system with *AltaVista, Excite* and *Google* search engine results)

6 Conclusion

In this paper, we have presented an information personalization approach for improving information retrieval effectiveness. Our study focused on temporal context information, mainly the day and time of day. We have attempted to investigate the impact of such data in the amelioration of the user models, the identification of the user needs and finally in the improvement of the relevance of search results. In fact, the built models prove its effectiveness and ability to assign the user to her/his profile class;

There are several issues for future work, for example, it would be interesting to support on an external semantic web resource (dictionary, thesaurus or ontology) for disambiguate query keywords and better identifying similar queries to the current one; also we attempt to enrich the data web house with other log files in order to test this approach in a wide area.

Moreover, we attempt to integrate this system as a mediator between surfers and search engines. To do this, surfers are called to submit their query to the system which detect their profile class and reformulate their queries before their submission to a search engine.

References

1. Anand, S.S., Mobasher, B.: Intelligent Techniques for Web Personalization. In: Carbonell, J.G., Siekmann, J. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 1–36. Springer, Heidelberg (2005)
2. Berendt, B., Hotho, A., Stumme, G.: Towards semantic web mining. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 264–278. Springer, Heidelberg (2002)
3. Cooley, R.: The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns. ACM Transactions on Internet Technology (TOIT) 3, 102–104 (2003)
4. Fischer, G., Ye, Y.: Exploiting Context to make Delivered Information Relevant to Tasks and Users. In: 8th International Conference on User Modeling, Workshop on User Modeling for Context-Aware Applications, Sonthofen (2001)
5. Garcia, P., Amandi, A., Schiaffino, S., Campo, M.: Evaluating Bayesian Networks' Precision for Detecting Students' Learning Styles. Computers and Education 49, 794–808 (2007)
6. Glance, N.-S.: Community Search Assistant. In: Proceedings of the 6th International Conference on Intelligent User Interfaces, pp. 91–96. ACM Press, New York (2001)
7. Jansen, B., Spink, A., Wolfram, D., Saracevic, T.: From E-Sex to E-Commerce: Web Search Changes. IEEE Computer 35, 107–109 (2002)
8. Joachims, T.: Optimizing search engines using click through data. In: Proceedings of SIGKDD, pp. 133–142 (2002)
9. Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems 23, 5–16 (2004)
10. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the effectiveness of collaborative filtering on anonymous web usage data. In: Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP 2001), Seattle, pp. 181–184 (2001)
11. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Communications of the ACM 43, 142–151 (2000)
12. Quiroga, L., Mostafa, J.: Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems. In: Proceedings of the 63rd Annual Meeting of the American Society for Information Science and Technology, Medford, vol. 37, pp. 4–13. Information Today, NJ (2000)
13. Salton, G., McGill, M.: Introduction to Modern Information Retrieval, New York (1983)
14. Shavlik, J., Eliassi-Rad, T.: Intelligent agents for web-based tasks: An advice taking approach. In: Working Notes of the AAAI/ICML 1998 Workshop on Learning for text categorization, Madison, pp. 63–70 (1998)
15. Shavlik, J., Calcari, S., Eliassi-Rad, T., Solock, J.: An instructable adaptive interface for discovering and monitoring information on the World Wide Web. In: Proceedings of the International Conference on Intelligent User Interfaces, California, pp. 157–160 (1999)
16. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine. Journal User Modeling and User-Adapted Interaction 14, 383–423 (2005)
17. Speretta, S., Gauch, S.: Personalizing search based user search histories. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2005, Washington, pp. 622–628 (2005)
18. Spink, A., Wilson, T., Ellis, D., Ford, N.: Modeling user's successive searches in digital environments, D-Lib Magazine (1998)

19. Trajkova, J., Gauch, S.: Improving Ontology-Based User Profiles. In: Proceedings of RIAO 2004, France, pp. 380–389 (2004)
20. Van-Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)
21. White, R.W., Jose, J.M., Ruthven, I.: Comparing explicit and implicit feedback techniques for web retrieval. In: Proceedings of the Tenth Text Retrieval Conference, Gaithersburg, pp. 534–538 (2001)
22. Yannibelli, V., Godoy, D., Amandi, A.: A Genetic Algorithm Approach to Recognize Students' Learning Styles. Interactive Learning Environments 14, 55–78 (2006)
23. Zhao, Q., Hoi, C.-H., Liu, T.-Y., Bhowmick, S., Lyu, M., Ma, W.-Y.: Time-Dependent Semantic Similarity Measure of Queries Using Historical Click-Through Data. In: Proceedings of 15th ACM International Conference on World Wide Web (WWW 2006). ACM Press, Edinburgh (2006)