Project Number 001907

DELIS
Dynamically Evolving, Large-scale Information Systems

Integrated Project
Member of the FET Proactive Initiative **Complex Systems**

DELIS-TR-588

Adaptive Personalization of Web Search

Elbassuoni, Shady and Luxenburger,
Julia and Weikum, Gerhard

2007

Adaptive Personalization of Web Search

Shady Elbassuoni
Max-Planck Institute of Informatics
Saarbrücken, Germany
elbass@mpi-inf.mpg.de

Julia Luxenburger
Max-Planck Institute of Informatics
Saarbrücken, Germany
julialux@mpi-inf.mpg.de

Gerhard Weikum
Max-Planck Institute of Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

In this paper we present a client-side approach towards personalization of web search which adapts the means of personalization to the user need in place. We differentiate three different search goals: re-finding known information, finding out about topics of user interest, and satisfying an ad-hoc information need. Our approach carefully balances these search modes, which is endorsed by preliminary results of a small-scale user study.

1. INTRODUCTION

An often stated problem in state-of-the-art web search is its lack of user adaptation, as all users are presented with the same search results for a given query string. A user submitting an ambiguous query such as "java" with a strong interest in traveling might appreciate finding pages related to the Indonesian island Java. However, if the same user searched for programming tutorials a few minutes ago, the situation would be completely different, and call for programming-related results. Furthermore suppose our sample user searches for "java hashmap". Again imposing her interest into traveling might this time have the contrary effect and even harm the result quality. Thus the effectiveness of a personalization of web search shows high variance in performance depending on the query, the user and the search context. This coincides with the findings in [4] from a large-scale study on MSN query logs. To this end, carefully choosing the right personalization strategy in a context-sensitive manner is critical for an improvement of search results. In this paper, we present a general framework that dynamically adapts the query-result ranking to the different information needs in order to improve the search experience for the individual user. We distinguish three different search goals, namely whether the user re-searches known information, delves deeper into a topic she is generally interested in, or satisfies an ad-hoc information need. We take a relevance feedback approach in the spirit of Rocchio [11]; however, we vary what constitutes the examples of relevant and irrelevant information according to the user's search mode. This strategy yields an adaptive personalization that exploits context, yet avoids pitfalls of earlier approaches.

The remainder of the paper is structured as follows. Section 2 reviews related work, Section 3 introduces our personalization approach, and preliminary experimental results are shown in Section 4.

1st Workshop on Web Information-Seeking and Interaction '07 Amsterdam, The Netherlands

2. RELATED WORK

There are a number of attempts on personalizing Web search. Due to space limitations we give a high-level categorization of what has been done along with some exemplary references which are, however, not meant to be exhaustive. One way of personalizing search is by means of implicit user relevance feedback. Approaches along these lines include [17, 13] which inspired the work presented in this paper. They achieve personalization by a client-side re-ranking of Web search results based on the previous user search and browse behavior. However, each one tackles a single facet of personalization, either biasing search results to general user-interests [17] or respecting the current search session's context [13], while we unify both aspects and dynamically switch between these two search modes. Our approach towards handling the current session context builds upon ideas in [13], and extends them to the whole user clickstream during a search session.

Another path of addressing personalization is by the categorization of both user interests and search results and a biasing of search results according to some similarity measure on these categories. Approaches along these lines include [9, 3, 18]. Personal biases inside the state-of-the-art link analysis algorithms such as PageRank [2] and HITS [8] provide a further means to shift search results according to user interests. E.g., [6] has been the first to propose biasing the random jumps inside the PageRank algorithm towards pages of user interests. In [10] this idea is further extended by automatically learning topic preferences from the user search behavior. Some personalization techniques not only consider a single user, but also take the actions of a surrounding group of users into account, e.g., [14] follows a collaborative filtering approach.

3. OUR PERSONALIZATION APPROACH

We aim at a holistic approach towards search personalization that supports and balances the different search goals a user might pursue. To this end, we differentiate three major search modes as follows.

Re-finding known information. As motivated in [16], returning to information once successfully found is an important user need. Despite the existence of bookmarking tools that would allow the user to achieve this goal in a direct manner, users quite often prefer to re-search for information by re-submitting a previously issued query [16].

Finding out about topics of user interest. By considering the long-term search and browse history of a user, the main topics of user interest emerge. Whenever a user query is ambiguous or broad in nature, superposing the learnt user interests might serve the user search experience. However, as already found in [15] the benefit of such a approach might differ for *recurring* as compared to *fresh* queries which motivates a differentiated usage of long-term user information.

Serving an ad-hoc information need. Yet even though a user might have strong focus on several topics of interest, she still might switch interests or develop some short-term information needs outside the scope of her interests.

Before we dig into the details of the approaches to each of these personalization facets in Section 3.3, our system architecture, the general framework and underlying retrieval model are introduced.

3.1 Personalized search architecture

As especially the browsing activities beyond search are outside the reach of a search engine, client-side solutions are favorable. Moreover, as all user data is kept locally, user privacy is not violated. We therefore set up a client-side search personalization with the use of a proxy which is running locally. It intercepts all HTTP traffic, extracts queries, *query chains*, i.e., subsequently posed queries, result sets, clicked result pages, as well as the whole *clickstream* of subsequently visited web pages, and stores this information to a local database file which we refer to as the *local index* in the following. Accordingly, searches with Google (the same approach can be easily applied to any other search engine as well) are intercepted and search results are re-ranked according to personal preferences. We preferred a proxy over implementing a plugin for browser-independence. Moreover, the proxy is broader applicable as it may bundle several users and thus achieve biasing of search towards community interests, and at the same time when run locally serve as a pure personalization tool. The proxy we are using relies on the UsaProxy implementation [1] that enhances all html files passed through by some Javascript code that sends logging information on events such as the load of page, mouse movements, etc back to the proxy.

For the following discussion we define our notion of a *search session* which is based on heuristics about the user's timing as well as the relatedness of subsequent users' actions. User actions are (1) queries, (2) result clicks, and (3) other page visits. All successive actions are considered to be within the same search session as long as they are no more than 15 minutes apart from each other or their similarity exceeds a certain predefined threshold. When computing the similarity of subsequent actions, a query is represented by the centroid of the top-50 result snippets.

3.2 Retrieval model

The standard vector space model [12] serves as our retrieval model which represents both queries and documents as a vector of features $\vec{X} = (x_1, x_2, \ldots\ldots, x_n)$, where n is the number of unique terms in the corpus and x_i is the score of feature i. Terms are weighted according to tf-idf [12]. To overcome the lack of web-corpus statistics that is usually prominent in client-side approaches to personalization, we approximate the global document frequency (df) statistics needed from the documents viewed and queries submitted during the search session. That is, not only each page visited or result viewed will contribute to the statistics, but also each query string, as well as each result item (snippet and title), is considered as a document in the corpus. That way it is not only ensured that each term present in a result item has a non-zero document frequency (as the term might not be present in the local index yet), but also session-biased df statistics are created. These are better suited for measuring the discriminative power of terms in the session context than index-wide statistics would be. Similarly, the features and the document lengths of query results are derived from their snippets and titles, as retrieving their full text would be too time-consuming.

To facilitate personalization of search results we utilize the relevance feedback framework introduced by Rocchio [11]. Thus, we associate with each query a query vector which is initially constructed from the query terms. This query is later augmented with terms that best differentiate relevant documents from non relevant ones. That is,

$$\vec{q_1} = \alpha\vec{q_0} + \frac{\beta}{n_1}\sum_{i=1}^{n_1}\vec{R_i} - \frac{\gamma}{n_2}\sum_{i=1}^{n_2}\vec{S_i}$$

where $\vec{q_0}$ is the original query vector, $\vec{q_1}$ is the refined query vector, $\vec{R_i}$ is the i^{th} relevant document vector, $\vec{S_i}$ is the i^{th} non-relevant document vector, n_1 is the number of relevant documents in the corpus, and n_2 is the number of non-relevant documents in the corpus. The parameters α, β, and γ control the influence of relevant, and non-relevant documents on the refined query vector.

Once we have made the choice of using this relevance feedback model to improve the query representation, the problem dwells down to infering relevant and non-relevant documents with respect to the user need currently in place.

Result re-ranking. Our search agent retrieves more results than the typical user is likely to view (50 results). Whenever a user action allows to update the query representation, unseen results are re-ranked. E.g., this is the case when the user submits a query or when she presses the "Next" link to view more results. Yet we refrain from re-ranking when the user returns to a seen result list using the "Back" button, as we perceived this more as irritating than as advantageous.

Merging of personalized and original results. In order to incorporate the query-independent web page importance, personalized result ranks and original web ranks (as an approximation for the real page rank) are aggregated to form the final result ranking. Inspired by rank aggregation methods for the web presented in [5], we use Borda's method to combine the two result rankings. Thereby each result item is assigned a score corresponding to the number of results ranked below it. Then the total score of a result is a weighted sum of its scores with respect to each ranking, such that the combination weight w serves as a personalization control parameter. In our search agent, we provide the user with a sliding bar, with which she may control the value of w, thus enabling the user to cancel the personalization at any point of time.

3.3 Personalization Strategy

In the following we present for each search mode in detail how it affects the ranking of search results. The decision which search goal a user pursues, and thus which kind of personalization method applies, is currently based on heuristics. However, we plan to study more sophisticated adaptation factors next.

Re-finding known information. Whenever the first query in a session, has occurred in some previous session, we assume the user wants to re-find some information already searched before. We apply three strategies to satisfy this user need. First, we consult the local index for a suggestion how to re-write the query sent to the underlying search engine (in our case Google). This is the only case in which our search agent changes the query sent to Google. We thus implement a more conservative query expansion mode than [13]. Furthermore we give the user full control over this feature so that she may choose to cancel the automatic query rewriting at any time. Considering how the user modified her query in the previous session, gives valuable hints on an improved query formulation. We choose the last query in the previous query chain as the user most likely stopped refining her query when she was satisfied with the results. To ensure the robustness of our method, we additionally require the reformulated query to share at least one term with the original one.

Second, the query representation is updated interpreting all previously clicked result pages as relevant documents, whereas intentionally non-clicked documents ranked above clicked ones are treated as irrelevant to the query. However, non-clicked results that are ranked higher than a clicked one could be interpreted in two different ways: either the user has examined the result title and snippet and was not satisfied with the result, or the user has already seen or knows the result from a previous interaction. Thus in case the local index contains the result, we assume it is known to the user, and do not consider it as an irrelevant document, but ignore it during query refinement.

Third, documents visited in the previous session starting from result pages are returned as additional clickstream results associated with the result item from which they have been reached. We believe this to be useful in cases where the result page is a directory or a summary page with many links to more specific documents.

Finding out about topics of user interest. Whenever there were no interactions recorded for a recurring query or the first query in the session did not occur before, what the user is generally interested in, might be a good guess of her current interest. Thus, we perform personalized pseudo-relevance feedback by assuming that the top-10 documents retrieved from the user's local index are relevant. The terms used to construct the query vector are selected from the titles or summaries of the top-10 documents.

In addition, the returned result set is extended by the top-10 documents from the local index. By doing so, we enable the user to search her own history of viewed Web pages.

Serving an ad-hoc information need. For every query except the first in a session, we refrain to the context provided by the current search session for personalization. The query representation is updated whenever a result click, a page visit or a query refinement occurs. (1) In case of a result click, the user profile of the query to which the result belongs is updated to include terms that best differentiate the clicked and intentionally non-clicked results. For all other similar queries within the session, the query vectors are updated to incorporate terms from the clicked result. (2) In case a page visit has occurred, the query vectors of all queries that are similar to the visited page are updated to incorporate terms from the visited page. (3) Finally, in case the user has refined her query, the new query is augmented with terms from previous similar queries within the current session. Again, for computing query similarities queries are represented by the result sets' centroids. Updating the representation of earlier queries in the current session is typically useful in cases where the user returns back to a query and investigates its unseen results, or in the face of parallel query submissions through tabbed browsing.

4. EVALUATION RESULTS

4.1 Experimental Setup

In order to evaluate the effectiveness of our proposed approach, we asked 9 volunteers to evaluate 10 self-chosen queries. Before the evaluation took place, the participants used our proxy on their local machines to log their browsing activities for a period of 2 weeks. For each participant, 5 of the evaluated queries were about topics the participant had been inquiring during the logging period and were used to assess the effectiveness of our personalization approach in case of re-finding known information, and finding out about topics of user interest. For each query, the participant was presented with the top-50 Google results, respectively additional top-50 results for the re-written query, placed in random order in order to avoid result's position bias. Then the participant was asked to mark each result as highly relevant, relevant or completely irrelevant. The rest of the evaluated queries were used to assess the quality of result re-ranking based on the search session context. Thereby participants performed a normal search with our personalization in place. After finishing their search, they were asked to evaluate the top-50 Google results of the last posed query in that session.

To measure the ranking quality, we use the Discounted Cumulative gain (DCG) [7], which is a measure that takes into consideration the rank of relevant documents and allows the incorporation of different relevance levels. DCG is defined as follows

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

where i is the rank of the result within the result set, and $G(i)$ is the relevance level of the result. We used $G(i) = 2$ for highly relevant documents, $G(i) = 1$ for relevant ones, and $G(i) = 0$ for non-relevant ones.

4.2 Experimental Results

As shown in Table 1, automatical re-writing of recurring queries clearly improves search result quality. E.g., the query "eccentricity" is reformulated as "eccentricity graph

Result Set	NDCG	Standard deviation
Original Google	0.469	0.178
Automatically re-written	0.803	0.128

Table 1: Average NDCG for recurring queries.

"eccentricity" Google's NDCG: 0.138	"eccentricity graph theory" Personalized NDCG: 0.932
Eccentricity - Wikipedia, ...	Glossary of graph theory - Wikipedia, ...
Orbital eccentricity - Wikipedia, ...	Glossary of graph theory - Information from Answers.com
Eccentricity - from Wolfram MathWorld	Graph Clustering for Very Large Topic Maps
Eccentricity ONLINE	Graph Theory - from Wolfram MathWorld
Scropio	LINK: a combinatorics and graph theory work bench ...

Table 2: Top-5 results (query re-writing).

theory" by our system resulting in a more than 6 times better NDCG (see Table 2 for the top-5 results).

Ranking Method	NDCG	Standard deviation
Original Google	0.806	0.182
Local-index PRF	0.794	0.182
Local-index PRF($w = 0.5$)	0.824	0.182
Textual Similarity	0.681	0.203

Table 3: Average NDCG for pseudo-relevance feedback from the local index.

Table 3 gives NDCG values for pseudo-relevance feedback (PRF) based on the local index. The pure personalized results slightly overdue personalization, however, when combined with the original web ranks choosing $w = 0.5$, the original Google results are outperformed. As an additional competitor we consider the performance of re-ranking the top-50 Google results based on pure textual similarity to the original query, which is consistently outperformed by the personalized results. Results for the sample query "vilnius, lithuania" are presented in Table 4. We see the top-5 results from the user's local index being all about hotels in Vilnius, thus biasing the personalized results more towards travel guides, hotels and outings in Vilnius.

When investigating the effectiveness of the session context for personalization, we find slight but consistent improvements over the original Google results (see Table 5). Again, combining personalized results and original web ranks further improves the ranking. The ranking quality obtained by re-ranking the results based on the local index indicates the need for our approach of different personalization strategies based on the information need.

Top-5 local index results	Google NCDG: 0.731	Personalization NCDG: 0.841
Radisson Sas Astorija hotel Vilnius	Vilnius City Municipality	Lithuania travel guide
Ramada Vilnius hotel	The Web's No. 1 Lithuanian Tourist Guide	Vilnius, Lithuania Restaurants
Crowne plaza Vilnius hotel	U.S. Mission to Lithuania	Lithuania Hotels Booking ...
Novotel Vilnius hotel	Vilnius	Lithuania in your pocket city guide ...
Vilnius forum: Kaunas to Vilnius - trip advisor	Lithuania in your pocket city guide ...	Vilnius Lithuania (Google maps)

Table 4: Top-5 results for query "vilnius, lithuania" (pseudo-relevance feedback from the local index).

Ranking Method	NDCG	Standard deviation
Original Google	0.766	0.18
Session context	0.783	0.154
Session context ($w = 0.5$)	0.784	0.185
Local-index PRF	0.751	0.18
Local-index PRF ($w = 0.5$)	0.729	0.192
Textual Similarity	0.666	0.201

Table 5: Average NDCG for session-context personalization.

5. REFERENCES

[1] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the user's every move - user activity tracking for website usability evaluation and implicit interaction. In *WWW*, 2006.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[3] P.-A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschuetter. Using odp metadata to personalize search. In *SIGIR*, 2005.

[4] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW*, 2007.

[5] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, 2001.

[6] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, 2002.

[7] K. Jrvelin and J. Keklinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR*, 2000.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM symposium on Discrete Algorithms*, 1997.

[9] F. Liu, C. Yu, and W. Meng. Personalized web search for improving retrieval effectiveness. In *IEEE Trans. on Knowledge and Data Eng.*, 2004.

[10] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *WWW*, 2006.

[11] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, 1971.

[12] G. Salton and M. J. McGill. Introduction to modern information retrieval. In *McGraw-Hill*, 1983.

[13] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *CIKM*, 2005.

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

[15] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *KDD*, 2006.

[16] J. Teevan. The re:search engine - helping people return to information on the web. In *UIST*, 2005.

[17] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR*, 2005.

[18] Y. Xu, B. Zhang, Z. Chen, and K. Wang. Privacy-enhancing personalized web search. In *WWW*, 2007.