SEREBIF
Search Engine Result Enhancement by Implicit Feedback

Ralph Weires, Christoph Schommer and Sascha Kaufmann
University of Luxembourg, Faculty of Sciences, Technology and Communication
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg

Keywords: Information Retrieval, Relevance Feedback, Implicit Feedback, Collaborative Retrieval.

Abstract: Web search engines often include results for a query which are not really relevant for a user. SEREBIF is an approach for incorporating feeback from the users into the search engine results to increase the result quality. We especially focus on implicit feedback, which does not require the users to put any additional effort than usual into the search process. The captured information (e.g. entered queries, clicked results) is afterwards analyzed, and the results are then taken into account for further search sessions. SEREBIF can generally be used on top of an existing search engine to improve its results. In this paper, we explain the basic idea of SEREBIF, the current state of the prototype we realized and first results.

1 INTRODUCTION

Today web search engines use more and more complex and elaborated ranking functions to deliver the proper results for a given query. On the other hand, website owners are trying to exploit the ranking methods to get their pages as high as possible in the result lists. In the end, it often happens that items not relevant for a user show up in the result list, too.

SEREBIF is an approach which makes use of implicit feedback for a given search engine, and incorporates this information into the results to increase their quality. The overall goal is to analyze the preferences and in general the behavior of users utilizing a search engine to get information about the real relevance of the results. For example, if the users always tend to click on the second given result for a certain query and usually leave out the first one, the ranking does not seem to be appropriate and could be changed accordingly (Joachims, 2002). So it should be possible to increase the quality of the results over time, just by taking the implicitely collected information about the clicked results into account.

Section 2 describes some related work, whereafter the SEREBIF system is explained in more detail in section 3. Section 4 shows the current status of our prototype, followed by the concluding section 5.

2 RELATED WORK

Relevance feedback (Salton and Buckley, 1990) can be divided into explicit and implicit feedback. The former type requires the users to actively give feedback, e.g. by answering questions. In the context of search results, this could mean to ask users which of the given results have actually been relevant to them. This type of feedback requires user cooperation, so the users need to be willing to spend additional time into giving the feedback. However, the information gained by this way is rather reliable, since it was explicitely given by the users. For the results of web search engines, few users would be willing to actually rate the given results, so explicit feedback would most likely be of limited practical use. Implicit feedback on the other hand is automatically collected information about the user behaviour (Kelly and Teevan, 2003). By this way, it is possible to get some information about what was really interesting to the users, without the need of explicitly asking them about it. So the users do not need to invest any extra time to give (explicit) feedback; they just do what they need to do anyway in order to find the relevant information that they are looking for. This makes it rather easy to obtain large amounts of data, for we do not need to motivate the users to actively give feedback to the system.

Important users' interactions which can be taken into account in the context of search engines are entered queries, clicked results, and the time that users

Weires R., Schommer C. and Kaufmann S. (2008).
SEREBIF - Search Engine Result Enhancement by Implicit Feedback.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 263-266
DOI: 10.5220/0001513202630266

spend on the result sites. The mere click on an item in the result list for a query can already be seen as an indication of relevance for that item. Moreover, the actual time which a user spends watching the result site can give a closer indication of relevance for that result (Kelly and Belkin, 2004). The gained information by this way is not as reliable as with explicit feedback, though. Users can of course also click on results which they later see are not important to them. However, we take the general assumption that clicks on results and the visiting time are generally a relevance indicator. This means that if many users click on a certain result for a given query, there is a high probability for this result being indeed relevant.

Furthermore, we want to keep track of the connections between multiple queries in a search session. It often happens that users cannot find the information they are looking for with a certain query and therefore use another, reformulated query afterwards. Such so-called query chains (Radlinski and Joachims, 2005) can be used to propose alternative queries to later users, helping them to find the desired information quicker.

3 ARCHITECTURE

This section outlines the basic ideas and the architecture used for SEREBIF. We generally require a search engine to be given whose results we want to improve.

3.1 Feedback Capturing

We decided to build SEREBIF as a proxy-like system, looking like a search engine to the users. Figures 1 and 2 show how the communication works. The first picture shows a search query coming from a user, which is forwarded to the underlying search engine. The results are returned to the user in a slightly modified way. This modification mainly affects the HTML links in the result page so that they point back to SEREBIF instead of the actual site of the result. By this way, SEREBIF is able to get to know on which results the users click afterwards. To the users, this does not make much of a difference, because a result request is immediately redirected to the actual result site as shown in figure 2.

The information we capture is recorded on a session basis only. This means that we do not keep profiles of individual people (or say, computers) over a longer period of time, since we do not track IP addresses or anything alike. So if the same user visits SEREBIF on two different days, the system will treat

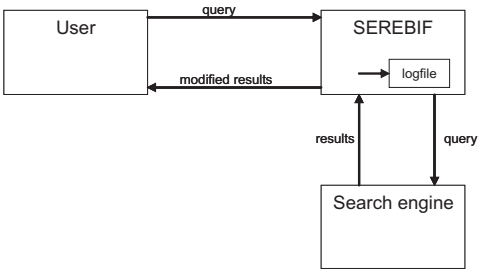

Figure 1: Communication for a search query.

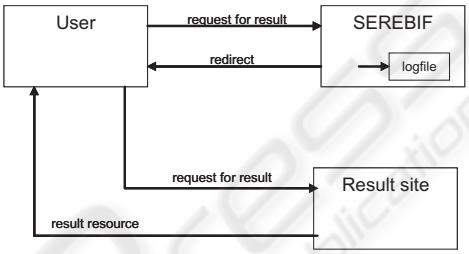

Figure 2: Communication for a result request.

this as two completely different sessions (of presumably different users). The reason for choosing this behaviour lies in our intention to keep the privacy of the users at the highest possible level while still making use of the feedback information. Information we keep track of for each session are:

- Entered queries, including timestamps
- Returned results delivered by the underlying search engine (limited to the first page of results)
- Clicked result links, including timestamps

3.2 Feedback Processing

After capturing, we apply a first preprocessing to divide the data into search sessions. These sessions contain one or more queries (possibly including query chains), the results from the underlying search engine for the queries, the clicked results for each query and an estimation of the duration the user stayed on each clicked result. We also plan to include external information into these sessions, such as synonyms for the contained query terms. To do this, we can make use of an external thesaurus like WordNet (Miller, 1995). Using such additional information, we can later apply techniques like query expansion to refine the results (Gong et al., 2005).

In this preprocessing step, we have to define the visiting duration of the result sites a user clicked on. The captured data only contains the time a user requested a result, and maybe timestamps of subsequent actions in this session. In the simplest case, we can

assume that the visiting time of a result is the duration from the result request to the next known action for that session (if any). However, this is certainly no precise measure, since a user can open multiple results for a query one after another in different windows (or browser tabs), without having looked at any of them already. Therefore, we experiment with different approaches to build an estimation model for the result visit duration.

The gathered information is afterwards merged in a network storage based on the ANIMA system (Schommer and Schroeder, 2005), (Schroeder et al., 2007). The reason for choosing ANIMA is mainly due to its ability to cope with transactional streaming data. The input data contains information about search sessions of users, which can easily be broken down into transactions for ANIMA. In addition, ANIMA is able to store only the most important aspects of the information without the need of storing all feedback data. This helps to keep the network at a reasonable size, especially if we think of having large amounts of captured feedback data. So if a certain result has only been clicked on by a single user for a given query, this information will be forgotten after a certain amount of time. This makes sense because we want to store the general trend of all users, without giving too much weight to single and isolated user actions.

The search sessions are considered as transactions for the ANIMA network. Building up such a network provides us with information about typical queries, results, and measures about the relevance of the results by the underlying search engine. Figure 3 shows how the storage looks like for SEREBIF. For our storage network, we use three different types of nodes (in the picture arranged in different layers):

Query terms (T): The single terms contained in user queries

Queries (Q): The full queries that have been entered

Documents (D): The resulting documents that have been provided by the underlying search engine

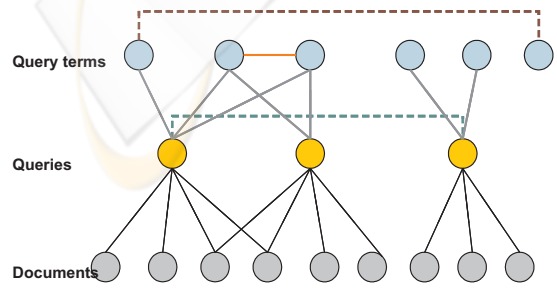

Figure 3: Storage architecture.

Different links between these nodes can exist, and they might be weighted to indicate the strength of the relationship. We propose the following connections:

TxT: Relationships between query terms. This can be based on co-occurence (terms tend to occur in queries together) but can also indicate a semantical relationship like synonyms.

TxQ: The connection between queries and query terms is a rather trivial one - it just indicates that a term was contained in that query.

QxQ: Connections between different queries can be established if it is found that users tried to find some information with multiple consecutive (reformulated) queries.

QxD: These links indicate that documents are of some relevance for a query. They are initially given by the information of the underlying search engine. The connection weights are subject to changes, so the importance of documents for queries can shift as time goes by. The weights are for example increased for documents which users actually looked (clicked) upon. The visiting time of documents is also taken into account here.

3.3 Feedback Integration

The information stored in the network can later be used in different ways to enhance further search engine results. First and most importantly, the clicks and visit times of the results can influence the ranking of the results. This means that if, for a given query and result list, the users steadily tend to visit the fifth result more often or longer than the first four results, it will be promoted higher in the list. However, we can also use the storage for other purposes. The connections known by query chains can help to guide users to the right information. If a user enters a certain query for which there are reformulated queries known from previous searches, it is possible to suggest these reformulated queries as alternatives. Similar things can also be done using the connections between query terms. Either additional terms can be suggested to the user, or they can automatically be included into the query which SEREBIF forwards to the underlying search engine. By that way, we support the users in refining the search into the desired direction and possibly eliminating ambiguities that might exist for a query.

4 CURRENT STATUS

The current prototype can be seen at the URL http://mine.uni.lu/~weires/serebif/search.php. To the user, SEREBIF looks just like a usual simple search engine (see figure 4), even though it in fact only redirects the queries to the underlying search engine. The

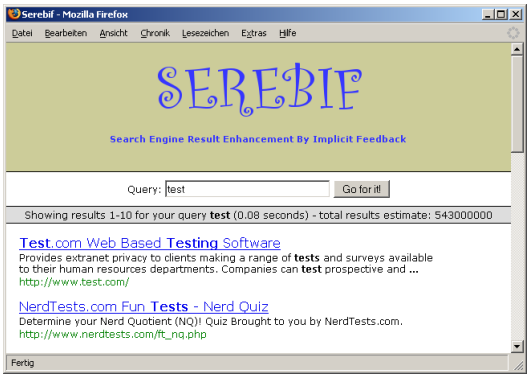

Figure 4: SEREBIF user interface.

implementation of SEREBIF is currently able to capture the user interaction as described. As underlying search engine, we are using the Google SOAP Search API[1].

Since the prototype is running just for a few weeks now, the recorded data set is not very big yet. We promoted our system mainly to fellow researchers and captured user information for about 150 sessions, including more than 650 queries and a total of over 1900 query terms up to now. We expect this amount of data to be steadily growing while we continue realizing our storage architecture.

5 CONCLUSIONS

In this paper, we described the SEREBIF system, which captures implicit feedback collected from the users of an existing search engine. This information is used to enhance the results of later search queries. To a certain extent, the search system adapts to the behaviour of the majority of the users, which is supposed to increase the quality of the results.

Our main future work currently lies in finishing and evaluating our prototype as described. There are also some other problems that we want to deal with in the future. We have to define how to merge the feedback information with the (unknown) ranking function of the underlying search engine in an appropriate way. Since we do not know the detailed ranking values of the results returned by the search engine, it is difficult to determine how much feedback from the users is needed to push a result up the list a certain amount. However, we plan to develop an own search engine to base SEREBIF upon, to be more independent of external systems. In this case, we would be able to access the exact ranking values for the result list, which makes this problem much easier to deal with.

Another problem of our approach is the vulnerability against fake feedback, which could possibly be exploited to artificially influence the ranking. If such a user-based approach would be included into a large commercial search engine, people could try to automatically generate false feedback for the system to push certain pages up the result list. This could e.g. be done by writing programs which submit a certain query to SEREBIF and always request the result in the result list which is desired to be promoted. Solutions have to be found to cope with this, too.

[1] http://code.google.com/apis/soapsearch/

REFERENCES

Gong, Z., Cheang, C. W., and U, L. H. (2005). Web query expansion by wordnet. In Andersen, K. V., Debenham, J. K., and Wagner, R., editors, *DEXA*, volume 3588 of *Lecture Notes in Computer Science*, pages 166–175. Springer.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD*, pages 133–142. ACM.

Kelly, D. and Belkin, N. J. (2004). Display time as implicit feedback: understanding task effects. In Sanderson, M., Järvelin, K., Allan, J., and Bruza, P., editors, *SIGIR*, pages 377–384. ACM.

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Radlinski, F. and Joachims, T. (2005). Query chains: learning to rank from implicit feedback. In Grossman, R., Bayardo, R., and Bennett, K. P., editors, *KDD*, pages 239–248. ACM.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *JASIS*, 41(4):288–297.

Schommer, C. and Schroeder, B. (2005). Anima: Associate memories for categorical data streams. In *Proceedings of the 3rd International Conference on Computer Science and its Applications (ICCSA)*.

Schroeder, B., Hilker, M., and Weires, R. (2007). Dynamic association networks in information management. In *Proceedings of the 4th International Conference on Machine Learning and Data Analysis (MLDA)*.