REUSING PAST QUERIES TO FACILITATE INFORMATION RETRIEVAL

Gilles Hubert[1], Josiane Mothe[1,2]
[1]IRIT/SIG-EVI, Université Paul Sabatier, 118 route de Narbonne F-31062 Toulouse cedex 9
[2]Institut Universitaire de Formation des Maîtres, 56 avenue de l'URSS, F-31078 Toulouse cedex
hubert@irit.fr, mothe@irit.fr

Keywords: information retrieval, past search experience, version management, query reformulation, recommendation.

Abstract: This paper introduces a new approach of query reuse in order to help the user to retrieve relevant information. Past search experiences are a source of information that can be useful for a user trying to find information answering his information need. For example, a user searching about a new subject can benefit from past search experiences carried out by previous users about the same subject. The approach presented in this paper is based on collecting the different search attempts submitted to a search engine by a user trying to fulfil an information need. This approach takes mainly advantage of implicit links that exist between the different search attempts that try to satisfy a single information need. Search experiences are modelled according to the concepts defined in the domain of version management. This modelling provides multiple possibilities to reuse past experiences notably to recommend terms for query reformulation or documents judged relevant by other users.

1 INTRODUCTION

Everyone agrees to recognize that experience is an invaluable thing and that it is important to pass it on to those who have little of it.

In a context of information retrieval (IR), search experiences performed in the past by previous users can be a useful source of information for new users for example. Nevertheless, few systems exploit this source of information. As underlined by (Klink 2004) a weak point of ad-hoc information retrieval systems is their absence of memory and their inability to learn. All the information about a retrieval are lost immediately after the presentation of the result list to the user.

Nevertheless, past search experiences can allow other users to better formulate their information need, to speed up their search, or to broaden their search for example. Many cases can take advantage of past search exploitation. For example, a user who searches for a document he previously seen but who does not remember the query that led to it should be interested in exploiting his past search experiences. Moreover, a user searching about a new subject could benefit from past search experiences carried out by previous users about the same subject. A user can benefit from the search experiences carried out by a group of users and vice versa.

We propose in this paper a way to overcome the lack of memory of an information retrieval system (IRS). The principle is to represent and store past search experiences notably regarding the different attempts generally carried out successively until the one that leads to a result satisfying the information need. The proposition is notably based on the use of the "version" concept. The version concept was notably defined to manage the evolution of complex objects. A search experience can be considered as a complex object and so its evolution can be managed through versions. Furthermore, this model offers more possibilities to exploit past search experiences.

This paper is organized as follows. Section 2 presents related works that deal with the reuse of past search experiences and draws up a synthesis of the different ways proposed for exploiting past searches. We introduce in section 3 the concepts through which we propose to manage past search experiences and the different ways to exploit past search experiences. Section 4 describes how the management of past experiences through versioning can be implemented in an IRS. Finally, section 5 concludes this paper and suggests future work.

2 RELATED WORK

2.1 Survey of existing approaches

Different studies following different objectives were interested in the reuse of past search experiences.

Various works are based on the storage of past queries along with their result list. Raghavan and Sever (1995) define similarity measures to retrieve past optimal queries that are used to reformulate new queries or to propose the results of past optimal queries. More recently, Klink (2004) proposes to learn from old queries and their result documents in order to expand the submitted query. The CIE system (Collaborative Index Enhancement) proposed by Selberg and Etzioni (1998) uses result documents of past searches or referenced documents to build additional indices. The system then fuses the results obtained with “usual” search engines and with the additional indices resulting from past searches. In the context of collaborative search, Fu et al. (2004) propose a system that provides a graphical visualization of query clusters close to each new submitted query. Then the user can select a query from clusters and submit it to the search engine.

Otherwise, a second group of proposals (Amitay et al., 2005 ; Kemp & Ramamohanarao, 2002) take an interest in Document Transformation. Document indices are modified according to past search experiences. The study presented in (Kemp & Ramamohanarao, 2002) deals with the use of the queries that led to judge a document relevant to transform the index of this document. Amitay et al. (2005) introduce the concept of reformulation session as the series of query reformulations issued by a user in order to satisfy a single information need. These different reformulations are used to transform the representations of the relevant documents judged relevant in the result of the last query reformulation.

Finally, a third type of approaches uses the principles of case-based reasoning (Aamodt, 1994). The system COSYDOR (Jeribi & Rumpler, 2002) uses case-based reasoning on instances that describe search experiences. An instance gathers information about the user, the query, the result documents, and the result evaluations. Similarity measures are defined to retrieve similar instances and the Rocchio’s relevance feedback principle (Rocchio, 1971) is extended to extract words from the similar instances in order to expand new queries submitted by users. Iszlai and Egyed-Zsigmond (2006) propose a system that uses case-based reasoning to annotate and search images. Cases are constituted of traces of retrieval (keywords) and navigations through image galleries. In addition to the usual process, suggestions of keywords and images resulting from retrieved similar cases are proposed to the user.

2.2 Experience exploitation synthesis

Past search experiences can be exploited through different points related to user assistance during his search process. Different exploitations of past searches can be found in existing works and can be divided as follows:

- Propositions of query reformulations (Islay & Egyed-Zsigmond, 2006 ; Klink, 2004 ; Jeribi & Rumpler, 2002),
- Uses of optimal queries instead of the submitted query (Fu et al., 2004 ; Raghavan & Sever, 1995),
- Propositions of documents resulting from similar past retrievals (Islay & Egyed-Zsigmond, 2006 ; Selberg & Etzioni, 1998 ; Raghavan & Sever, 1995),
- Propositions of document index enhancements (Amitay et al, 2005 ; Kemp & Ramamohanrao, 2002).

This paper presents a solution based on the notion of reformulation session and the storage of past search experiences in an information retrieval system. The principle is to store information describing the retrieval that led to a satisfying result and the previous unsatisfying attempts. The approach particularly stores and exploits the succession links existing between different retrieval attempts to satisfy a single information need. The past search experiences are considered as information source to propose different kinds of suggestions to the user. Ways to provide the first three kinds of exploitations listed above are introduced in this paper.

3 SEARCH EXPERIENCES

Our approach aims at integrating the management of search experiences and exploiting their evolution in an information retrieval system. According to our approach, a search experience gathers a succession of search engine retrievals in order to satisfy a given information need. These retrievals correspond to query reformulations submitted each time to the search engine. In our model, these retrievals are considered as evolutions of an initial retrieval and are managed through the concept of version. A search session lasts while the query evolutions and result consultation are related to the same information need than the one expressed at the beginning of the session.

3.1 Retrieval

A “retrieval” gathers a query and a result. A query is a list of keywords. A result is a list of documents that can be judged relevant or irrelevant by the user, or that remain not judged.

3.2 Query Reformulation

According to the same information need, query reformulation consists in modifying a query submitted to the search engine and submitting the modified query to constitute a new retrieval.

After the user has submitted a query to the search engine and a result list has been presented to the user, the query can evolve through a manual process, a semi-automatic process, or an automatic process:

- The user modifies manually the query by adding or removing keywords,
- The system performs a query reformulation process soliciting interactively the user (Taghva et al., 2004 ; Efthimiadis & Robertson, 1989) for example for documents judged relevant by the user (Salton & McGill, 1986),
- The system performs an automatic analysis of the first result or external information and then proposes or directly applies possible modifications of the query (Benammar et al., 2002 ; Mitra et al., 1998 ; Xu & Croft, 1996),
- The system extracts information from past search experiences related to the same information need (Klink, 2004 ; Jeribi & Rumpler, 2002 ; Fitzpatrick & Dent, 1997).

3.3 Reformulation session

A reformulation session is a succession of query reformulations that aim at satisfying a single information need (Amitay et al., 2005). So, a reformulation session gathers a succession of combinations of query and list of result documents linked by implicit links. However, in existing systems these links are not stored and thus not exploited.

3.4 Retrieval Versioning

Our approach takes an interest in the implicit links existing between the different reformulations of a query answering a given information need. These links that are not currently kept seem to be nevertheless a useful source of information. The different successive retrievals are successive evolutions of a single search and so can be modelled as versions. In our approach, version management provides the capability to store explicitly these links as “evolution” links between versions of retrieval (cf. Figure 1).

3.5 Search Experience

The notion of reformulation session introduced by Amitay et al. (2005) is reused and extended to integrate the links that exist between the different reformulations of a given query. A search experience is thus modelled as a set of versions of retrieval linked by evolution links (cf. Figure 1).

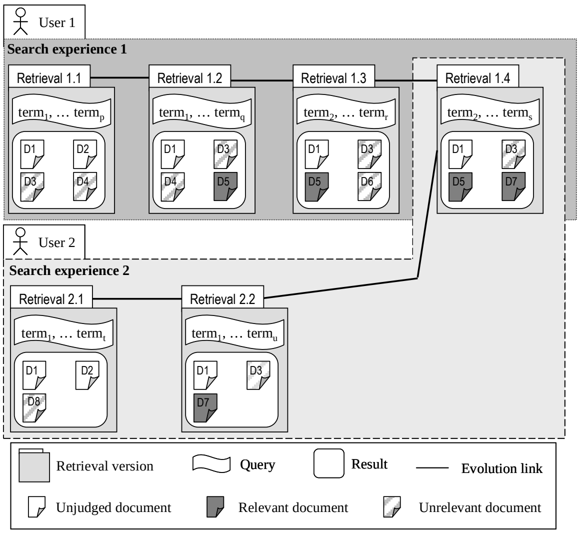

Figure 1: Search experiences

4 VERSIONING EXPLOITATION

Different exploitations of past experiences can be carried out. We propose firstly to present how our modelling of search experiences can be used to propose term recommendations (to reformulate a query), query recommendations (to replace the initial query), or document recommendations.

From a query expressed by the user, the stored versions of past retrievals can be used to propose recommendations. The initial query is compared to the queries in the stored versions of retrievals. If a high similarity is estimated (for example, over a given threshold defined by the user or after a learning phase, or resulting from experiments), different recommendations can be proposed to the user:

- keywords used in the queries of the closest past experiences can be used for term recommendations,

- last query formulations of the experiences containing the closest versions of retrievals can be proposed to the user in replacement of the initial query. For example in Figure 1, if the ‘Retrieval 1.2’ is found similar to a new query, the last query formulation in ‘Retrieval 1.4’ related to the ‘Retrieval 1.2’ can be proposed to the user,
- documents judged relevant in the results of past experiences containing the closest versions of retrievals can be proposed to the user.

The search in past experiences can be based on:

- Only the query defined by the user before submitting it to the search engine,
- The query and its result list returned by the search engine,
- The query, the result list and the document contents.

Depending on the cases, appropriate similarity measures have to be applied:

- Similarity between queries,
- Similarity between result lists,
- Similarity between query and document.

5 IMPLEMENTATION

The implementation of our approach can be based on different principles related to:

- Modelling of search experiences. It concerns the definition of stored information with regard to queries, retrieved documents, …,
- Version management. It concerns the definition of versioning adapted to the problem of part experience reuse,
- Similarity. It concerns the definition of similarity measures taking into account the elements handled (queries, result lists, and documents).

5.1 Modelling Search Experiences

A search experience is considered in our approach as a set of retrieval versions linked by evolution links. Each retrieval is constituted of a query and a result. A query is a list of keywords. A result is a list of documents retrieved by a search engine. Every document is considered as a set of keywords. The documents presented to the user can be judged relevant or irrelevant, or remain not judged.

In our approach, search experiences can be modelled, in a simplified manner, as follows (Figure 2):

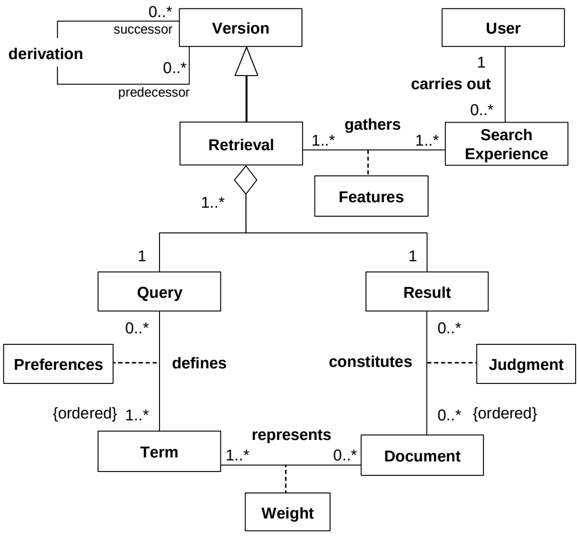

Figure 2: Simplified UML class diagram model describing search experiences

5.2 Version Management

Evolution of information search can be managed through object versioning. In our approach, a retrieval is a complex object gathering a query and a result list of documents. A new version is created every time a query reformulation is done and submitted to the search engine. Various works related to version management were carried out in the domain of software configuration management (Conradi & Westfechtel, 1998), or in the domain of databases (Jomier & Cellary, 2000 ; Andonoff et al., 1998 ; Katz, 1990). Solutions have been notably proposed to limit the volume of versions created.

We defined a framework to manage versions of complex objects in databases. This framework was implemented through a prototype (Andonoff et al., 1998). This framework notably makes it possible to create object databases integrating version management of complex objects, to maintain object databases including versions, and to query object databases including versions through a textual SQL-like language and a graphical language.

5.3 Similarity

In the context of information retrieval integrating reuse of past search experiences, different similarity measures must be defined (cf Section 4). These similarity measures are based on the different concepts handled, i.e. queries, result lists, and documents. The main similarity measure to define is the one used in the usual retrieval process, i.e. similarity between query and document. Additional similarity measures have to be defined between queries, between result lists, and between documents.

5.3.1 Query-Document and Inter-Document Similarities

The query-document similarity intervenes firstly in the "usual" ad-hoc retrieval process to treat a query. Our approach is based on a vector space model (Salton et al., 1975). Documents and queries are represented as vectors of weighted terms. The cosine measure can be used to compute a similarity score. However, we have defined a search engine being adaptable to different contexts that is based on a scoring function highly configurable according to the search context.

$$Score(A,B) = \left(\sum_i g(t_i, A) \cdot h(t_i, B) \right) \cdot q(A,B) \qquad (1)$$

Where A and B are vectors

$g(t_i, A)$ Function that estimates the importance of the term t_i in the vector A

$h(t_i, B)$ Function that estimates the importance of the term t_i in the vector B

$q(A, B)$ Function that estimates the global matching between the vectors A and B

In the context of this paper, the search engine has to be able firstly to retrieve a list of documents responding to the query. This type of search corresponds to the usual ad-hoc retrieval. In this case, the scoring function can be defined as follows:

$$Score(Q,D) = \left(\sum_i f_{t_i,Q} \cdot \frac{f_{t_i,D}}{d_{t_i,C_D}} \right) \cdot \varphi^{\frac{N_{D,Q}}{\min(N_D, N_Q)}} \qquad (2)$$

where Q is a query and D is a document

$f_{t_i,Q}$ Frequency of the term t_i in the query Q

$f_{t_i,D}$ Frequency of the term t_i in the document D

d_{t_i,C_D} Number of documents in the corpus C_D (i.e. all the documents) that contain the term t_i

$N_{D,Q}$ Number of terms common to the document D and the query Q

N_D Number of distinct terms of the document D

N_Q Number of distinct terms of the query Q

φ Positive real. The value can be adjusted depending on the corpus features.

An adaptation the scoring function was notably experimented in the context of ad-hoc retrieval in collections of XML documents (Hubert, 2006).

Similarity between documents can be evaluated with the cosine measure widely used (Salton et al., 1975). Since queries and documents are both represented as vectors of terms, the same scoring function can also be used for similarity between documents In this case, one document plays the role of query.

5.3.2 Inter-query and Inter-Result Similarities

Similarity between queries can be defined simply according to the proportion of terms common to both queries. However, this measure does not take into account term order in queries. A solution evoked by Fitzpatrick and Dent (1997) to compare result lists of documents is to use the term positions in both queries. This principle can be integrated to our scoring function as follows:

$$Score(Q,Q') = \left(\sum_i wpos_{t_i,Q} \cdot wpos_{t_i,Q'} \right) \cdot \varphi^{\frac{N_{Q,Q'}}{\min(N_Q, N_{Q'})}} \qquad (3)$$

Where

$wpos_{t_i,Q}$ Weight function associated to the position of the term t_i in the query Q

$wpos_{t_i,Q'}$ Weight function associated to the position of the term t_i in the query Q'

$N_{Q,Q'}$ Number of terms common to the queries Q and Q'

N_Q Number of distinct terms of the query Q

$N_{Q'}$ Number of distinct terms of the query Q'

Similarity between result lists is analogous to similarity between queries when considering result lists as lists of document identifiers and queries as lists of terms. Similarity between result lists can be estimated simply by the proportion of common documents between both lists, and integrating also the position of documents in both result lists.

6 CONCLUSIONS

This paper deals with reuse of past searches in the context of information retrieval. Past search experiences are generally lost just after the result list returned by the search engine is presented to the user. This paper describes a solution to overcome this limit by storing past search experiences. The proposition is based on the idea that a search is generally a succession of different retrieval attempts. The search ends when a query formulation leads to a result that satisfies the information need. In our approach, information searches are considered as complex objects that evolve until succeeding. This evolution of complex object is managed through the concept of version. Versioning notably offers multiple possibilities to exploit past search experiences. Different possible exploitations are illustrated in this paper. An implementation of the approach in an information retrieval system is introduced.

Currently, this work represents a first step. A second step will consist in evaluating the contribution of past experience reuse. Kemp and Ramamohanarao (2002) underlined that there was no collection really suited for this kind of evaluation and recent studies are still based on self-made test collections. This second step goes through the definition of an appropriate testbed. Furthermore, an advantage of the search experience modelling presented in this paper is that it offers different possibilities to exploit past experiences. Therefore, an extension of this work will be oriented to the possibilities to exploit past experiences and the way to propose the exploitation results to users. Finally, another advantage of this model is that the notion of search experience can be extended to the notion of evolving retrieval context. Future work will be so related to contextual information retrieval.

REFERENCES

Aamodt, A., Plaza, E., 1994. *Case-based reasoning: foundational issues, methodological variations, and system approaches*. AI Communications, 7, 1, pp. 39-59.

Amitay, E., Darlow A., Konopnicki,, D., Weiss, U., 2005. Queries as Anchors: Selection by Association. *Sixteenth ACM Conference Hypertext* (pp. 193-201).

Andonoff, E., Hubert, G., Le Parc, A., 1998. A Database Interface Integrating a Querying Language for Versions. *2nd East European Symposium ADBIS, LNCS 1475* (pp. 200-211).

Benammar, A., Hubert, G., Mothe, J., 2002. Automatic Profile Reformulation Using a Local Document Analysis. *24th BCS-IRSG European Colloquium ECIR,. LNCS 2291* (pp. 124-134).

Conradi, R., Westfechtel, B., 1998. Version models for software configuration management. ACM Computing Surveys, Volume 30, Issue 2, pp. 232-282.

Corvaisier F., Mille A., Pinon J.-M., 1997. Information retrieval on the World Wide Web using a decision making system. *International conference RIAO* (pp. 284-295).

Efthimiadis, E. N., Robertson, S. E., 1989. *Feedback and interaction in information retrieval*. Perspectives in Information Management. Butterworths, pp. 257-272.

Fitzpatrick, L., Dent, M., 1997. Automatic feedback using past queries: social searching?. *20th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 306-313).

Fu L., Dion Goh D. H.-L., Foo S. S.-B., Supangat Y., 2004. Collaborative Querying for Enhanced Information Retrieval, *8th European Conference ECDL, LNCS 3232* (pp. 378-388).

Hubert, G., 2006. XML Retrieval Based on Direct Contribution of Query Components. *4th International Workshop INEX 2005, LNCS 3977* (pp. 172-186).

Iszlai Z., Egyed-Zsigmond E., 2006. User centered image management system for digital libraries. *2nd international Conference on Document Image Analysis For Libraries (Dial'06) - Volume 00* (pp. 164-171).

Jéribi, L., Rumpler, B., 2002. *Instance Cooperative Memory to Improve Query Expansion in Information Retrieval Systems*, Journal of Universal Computer Science, vol. 8, no. 6, pp. 591-601.

Jomier G., Cellary W., 2000. The Database Version Approach. Networking and Information Systems Journal, 3, 1, pp. 177-214.

Katz, R. H., 1990. Toward a unified framework for version modeling in engineering databases. *ACM Computing. Surveys*, Volume 22, Issue 4, pp. 375-409.

Kemp, C., Ramamohanarao, K., 2002. Long-Term Learning for Web Search Engines. *6th European Conference on Principles of Data Mining and Knowledge Discovery. LNCS 2431* (pp. 263-274).

Klink S., 2004. *Improving Document Transformation Techniques with Collaborative Learned Term-Based Concepts*, LNCS 2956, pp. 281-305.

Mitra, M., Singhal, A., Buckley, C., 1998. Improving Automatic Query Expansion. *21st Annual International ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 206-214).

Raghavan, V. V., Sever, H., 1995. On the reuse of past optimal queries. *18th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 344-350).

Rocchio Jr., J. J., 1971. *Relevance feedback in information retrieval. The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ, USA, pp. 313-323.

Salton, G., McGill, M. J., 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

Salton, G., Wong, A., Yang, C. S., 1975. *A vector space model for automatic indexing*. Communication of the ACM, 18 (11), pp. 613-620.

Selberg, E., Etzioni, O., 1998. *Experiments with Collaborative Index Enhancement*, University of Washington Technical Report UW-CSE-98-06-01.

Taghva K., Borsack J., Nartker T., Condit A., 2004. *The role of manually-assigned keywords in query expansion*, Information Processing & Management, Volume 40, Issue 3, pp. 441-458.

Xu J. and Croft W. B., 1996. Query Expansion Using Local and Global Document Analysis. *19th Annual International ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 4-11).