ARCHITECTURE AND RETRIEVAL METHODS OF A SEARCH ASSISTANT FOR SCIENTIFIC LIBRARIES

I. Glöckner, A. Knoll

Technische Fakultät, Arbeitsgruppe Technische Informatik, Universität Bielefeld, D-33501 Bielefeld, Germany

Abstract: In this paper, we present the design and retrieval methodology of an intuitively operated retrieval assistant (RA) which supports the thematic search in databases of scientific libraries. The retrieval assistant establishes innovative and more adequate means for expressing a user's search interest by adopting aggregation operators of natural language (e.g. *almost all*, *as many as possible*), the interpretation of which is accomplished by novel methods from fuzzy set theory. These operators can be used in their intuitive meaning, i.e. just as in everyday language, for aggregating over sets of weighted search terms. The required scalability of the system is ensured through its multi-tier architecture, which disburdens both the clients and the external database servers by introducing an (arbitrarily replicable) intermediary to perform the computationally intensive aggregation step.

1 Introduction

The retrieval environments of scientific libraries are currently subject to a number of changes. Firstly, there is an ever increasing use of networking, which is now used to cooperatively maintain databases, and to offer external catalogs, e.g. of scientific publishers. In addition, not only the number of source databases, but also their typical size has increased, in particular by including references or electronic full-texts of scientific journal articles, which can easily push data volumes to a size of some 10^7 records. Thirdly, today's bibliographic retrieval systems have typically been equipped with WWW interfaces, and can now provide potentially world-wide access. We identify the following requirements on retrieval systems which result from these developments:

- the very large data bases have given rise to an increase in the size of the "hit lists", and a corresponding loss of clarity and utility of retrieval results. Therefore, the search quality of the libraries' retrieval systems must be further enhanced;
- the increasing number of accessible retrieval systems not only means an increasing amount of searchable information; it is the accompanying gain in diversity or *heterogeneity* of the sources. In order to facilitate multi-database searches, an *integrating approach* is required which hides the heterogeneity of the sources and establishes a combined and uniform way of searching the information sources in their entirety;

- the WWW access using 'passive' HTML-forms has caused a loss of interactivity; in our view, WWW users should be enabled to operate the retrieval system efficiently and cooperatively, i.e. in a way familiar from modern office programs on their computer's desktop.

In order to improve the search quality, one measure which comes to mind is that of using techniques for reducing the linguistic variety observed in document descriptions (or for suppressing its effects), e.g. stemming, approximate matching, decomposition of compound nouns and the use of thesauri. However, most of these methods can only be added to a retrieval system by changing its interna, such as the indexing or the index search process. Due to the increased autonomy of the sources, such modifications have often become impossible; almost all variables of retrieval quality are outside one's grip. Improvements must start with the aggregation methods, the way in which intermediate results are combined to the final result of the search.

2 Integration approach

A typical architecture for information integration, which respects the autonomy of the sources, is depicted in Fig. 1.[1] It mediates access to a complex system of multiple and possibly very heterogeneous information sources in such a way that to the user, the illusion of a local database with rich informational content emerges. System operation (in particular query syntax and result presentation) must appear uniform on the user side. This abstraction is made possible by "wrappers" which translate the user queries into queries on the individual sources, control their execution and re-translate the received results into the data model and format chosen for integration.[2] The results of the individual wrappers are merged by the mediator component into a global logical view. With usual approaches, which transform the user query *as a whole* in corresponding queries on the underlying sources, only the intersection of the functionality of all considered systems can be utilised. To avoid this, the retrieval assistant (RA) decomposes the user queries into elementary subqueries (usually for one search term). It is the subsequent aggregation which improves the retrieval quality. From the outside, then, the RA system appears as a local database with rich information content *and powerful fuzzy search options*—although the underlying data sources are based on two-valued indexing and querying. In order to produce this illusion, the mediator component in the above architecture is enhanced to a *fuzzy query interpreter* (FQI) which, in addition to mediation, also performs the decomposition of queries involving fuzzy operators, sends the subqueries to the wrappers, and applies the fuzzy aggregation operators expressed in the queries to compute the final query result.

[1] In web-based search services, such systems are called "meta search engines"; in the database field, the term "mediator" (Wiederhold (1992)) has established.

[2] With textual query results (e.g. HTML pages of CGI interfaces), the result transformation can involve some kind of information extraction.

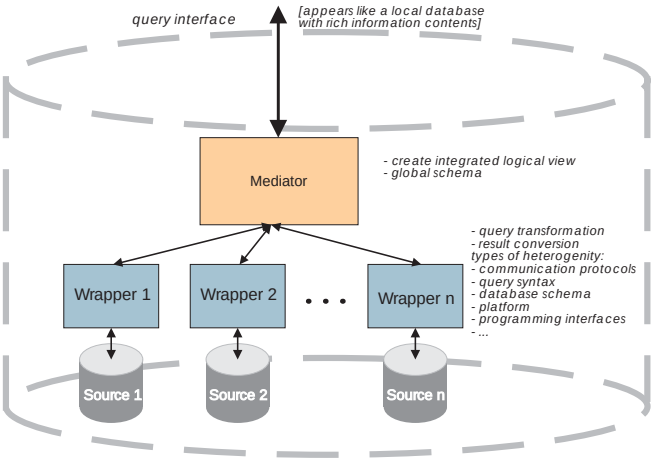

Figure 1: The basic integration architecture

The actual increase in retrieval quality which can be achieved with this strategy depends crucially on the choice of aggregation methods, which must be both "powerful" and "user-friendly" in order to unfold their potential under real-world conditions.

3 Fuzzy aggregation methodology

The Boolean retrieval model—which combines two-valued ("crisp") document evaluations using the connectives **and**, **or**, **not**—is still predominant in bibliographic retrieval systems. The RA attempts to improve upon this model (and alternative approaches), based on the following considerations.

1. The information loss caused by the yes/no-decisions of Boolean search should be avoided, by supporting *gradual evaluations of relevance*;
2. Query results should be ordered according to their relevance to the users' queries, i.e. some kind of *ranking* must be performed;
3. *Term-weights* should be supported in order to model varying degrees of term-user relevance;
4. Individual modes of querying are possible only when a *sufficiently expressive set of search operators* is exposed to the user. Hard-wired scoring criteria violate this condition.
5. In addition to pair-wise connectives, "holistic" aggregation operators should be offered which compute a *global trade-off* for sets of weighted criteria;
6. The search operators offered should be *meaningful* to users without special technical training, in order to ensure user-friendliness. This condition can be met by emulating the *aggregation operators of natural language*;
7. Only a *mathematically sound approach* can guarantee plausible results.

We wish to briefly remark on the last point, concerning the mathematical foundations of weighted retrieval. The topic has first been addressed

by Waller&Kraft (1979) who have proposed a "wish list" for mathematical models of weighted Boolean retrieval. However, the Waller-Kraft wish list (WKWL) has been shown to be inconsistent by Buell (1982). The assumptions underlying the wish list are incompatible with the *relevance-based* interpretation of term weights which views these as degrees of *term-user relevance* ("To which degree does the search term satisfy the user's search interest?"). Cater&Kraft (1987) have modified the WKWL to a consistent characterisation of "topological" weighted retrieval, which views term weights as "soft constraints" on the *term-document relevance* ("To which degree should the search term apply to an ideal matching document?").
In order to justify our preference of the relevance-based view, we will briefly discuss Waller&Kraft's claim (1979, p.244) that *"the [WKWL] model is a generalisation of a model of Boolean retrieval with discrete weights."* The WKWL assumes that a *single* weighting function $g : [0,1]\times[0,1] \longrightarrow [0,1]$ be sufficient, which combines a term-weight w and term-document relevance r to a gradual evaluation $g(w,r)$, in a way independent of aggregation context. In retrieval, g is applied to replace all weighted terms with combined evaluations in $[0,1]$. To these, the fuzzy connectives (min, max, $1-x$) are be applied in a subsequent step, which models the Boolean structure of the query.
Let us now consider *two-valued* (discrete) weighted retrieval, i.e. both term weights and term-document relevance only assume values 0 or 1. We can then restrict attention to weighting functions $g : \{0,1\} \times \{0,1\} \longrightarrow \{0,1\}$. A term weight of 0 indicates that the term should be ignored. With respect to conjunction or disjunction, this "ignoring" of a term corresponds to its replacement with the *identity* of the connective (0 in case of **or**, 1 in case of **and**). We hence obtain *different* weighting functions g_{and} and g_{or}:

w	r	$g_{\text{and}}(w,r)$	$g_{\text{or}}(w,r)$
0	0	1	0
0	1	1	0
1	0	0	0
1	1	1	1

This finding indicates that using a single weighting function is incompatible with two-valued weighted retrieval. In particular, the inconsistency of the relevance-based view with the WKWL is *not* caused by the presence of fuzziness, but rather reveals a misconception inherent to the wish list.
The relevance-based view must hence be justified differently. One possible solution is that of embedding weighted Boolean retrieval into the broader framework of fuzzy quantification:

> Weighted conjunction: **all** user-relevant terms are document-relevant
> Weighted disjunction: **some** user-relevant term is document-relevant

These expressions can be evaluated by computing $\widetilde{Q}(W,R)$, where $\widetilde{Q}$ is a fuzzy quantifier suitable for modelling **all** or **some**, W is a fuzzy set expressing the term-user relevance and R is a fuzzy set describing the gradual term-document relevance with respect to a document under consideration. This basic approach offers the opportunity to support a richer repertoire

NL expression	**possible definition of $Q(W,R)$**
all	$\begin{cases} 1 & : \ W \cap R \subseteq W \\ 0 & : \ \text{else} \end{cases}$
atleast k	$\begin{cases} 1 & : \ \lvert W \cap R \rvert \geq k \\ 0 & : \ \text{else} \end{cases}$
all except k	$\begin{cases} 1 & : \ \lvert W \rvert - \lvert W \cap R \rvert \leq k \\ 0 & : \ \text{else} \end{cases}$
some, a	$\begin{cases} 1 & : \ W \cap R \neq \varnothing \\ 0 & : \ \text{else} \end{cases}$
atleast r percent	$\begin{cases} 1 & : \ \lvert W \cap R \rvert \geq \lvert W \rvert r/100 \\ 0 & : \ \text{else} \end{cases}$
almost all	$\begin{cases} s(0.7, 0.9, \frac{\lvert W \cap R \rvert}{\lvert W \rvert}) & : \ W \neq \varnothing \\ 1 & : \ \text{else} \end{cases}$ using Zadeh's "s" function
a few	$\begin{cases} s(0.1, 0.4, \frac{\lvert W \cap R \rvert}{\lvert W \rvert}) & : \ W \neq \varnothing \\ 0 & : \ \text{else} \end{cases}$
many, as many as possible	$\begin{cases} \lvert W \cap R \rvert / \lvert W \rvert & : \ W \neq \varnothing \\ 1 & : \ \text{else} \end{cases}$

Table 1: Search Operators for Weighted Retrieval

of basic aggregation operators (e.g. in order to model "softer" conditions like **almost all** or **a few**), which are still easy-to-use because they can be applied for retrieval purposes *in the same way as in everyday language.* Recognising the benefits of fuzzy quantifiers, some promising attempts to apply these in information systems have been made, e.g. Bordogna&Pasi (1997). However, it was shown by Glöckner (1999) that existing approaches to fuzzy quantification fail to provide acceptable models of two-place quantifiers (the class of quantifiers needed for weighted retrieval). This study utilizes the Theory of Generalized Quantifiers of Barwise&Cooper (1981) to detect counter-examples for approaches to fuzzy-quantification, and to explain their failure as a violation of formal adequacy conditions. Glöckner (1997) has presented an alternative, axiomatic approach to fuzzy quantification ("DFS theory"). It is based on the concept of n-ary fuzzy quantifiers $\widetilde{Q} : \widetilde{\mathcal{P}}(E)^n \longrightarrow [0,1]$, where $\widetilde{\mathcal{P}}(E)$ is the fuzzy powerset of the base set E. So-called semi-fuzzy quantifiers $Q : \mathcal{P}(E)^n \longrightarrow [0,1]$, which are defined on crisp subsets of E only, serve as simplified descriptions which are mapped to corresponding fuzzy quantifiers $\mathcal{F}(Q)$ by applying a quantifier fuzzification mechanism (QFM) $\mathcal{F}$. The unrestricted set of QFMs is narrowed to a set of reasonable choices (called DFSes) by imposing a set of linguistically motivated axioms, which e.g. require the compatibility of $\mathcal{F}$ with negation and dualisation, preservation of monotonicity properties etc. Due to its axiomatic foundation, DFS theory can prevent implausible results like those of existing approaches on principled grounds.

Some two-place quantifiers of interest for weighted retrieval are presented in table 1. These operators express a number of straightforward ranking criteria. By directly offering these operators in a clickable menu and supporting their application to weighted search terms, the retrieval assistant gains *ease-of-use* compared to traditional systems in which users must try to obtain similar effects (e.g. "all terms except one") using Boolean connec-

tives and two-valued evaluations. Empirical tests will optimize the set of operators and the default choice of aggregation operator in the RA system. In Glöckner et.al. (1998), a histogram-based algorithm was presented which permits an efficient implementation of the resulting operators. Nevertheless, the aggregation procedure is still computationally costly. We will now describe the ranking problem and then present two techniques which will be used in the RA to ensure acceptable response times. Let E the set of search terms in a quantifying query, $W \in \widetilde{\mathcal{P}}(E)$ the fuzzy set of term weights specified by the user, and $\widetilde{Q}$ the chosen fuzzy quantifier. For each document d in the set of all documents D let use denote by $R_d \in \widetilde{\mathcal{P}}(E)$ the fuzzy set of those terms of E which occur in d. The task of the aggregation procedure is to compute a ranking (presentation order) $d_1, d_2, \ldots, d_N$ of the documents ($N = |D|$) such that $\widetilde{Q}(W, R_{d_i}) \geq \widetilde{Q}(W, R_{d_j})$ for all $1 \leq i \leq j \leq N$.[3] In order to reduce the computational effort of the aggregation procedure, we first observe that any pair of documents d, d' with $R_d = R_{d'}$ also has identical relevance degrees $\widetilde{Q}(W, R_d) = \widetilde{Q}(W, R_{d'})$. We can hence avoid multiple evaluation of a fuzzy quantifier for d' by fetching the result computed for some d with $R_d = R_{d'}$ from a *cache of aggregation results*. In our system, all external databases are based on two-valued indexing, i.e. R_d is always a crisp subset of E. Considering the equivalence classes with respect to $d \sim d'$ iff $R_d = R_{d'}$ there are maximally 2^m (m number of terms) classes. The number of classes can never exceed the number of documents, and will usually be much smaller. Users typically do not view the whole sequence of ranked documents, but only those on the first ranks. It is hence sufficient to compute only part of the ranking (first k ranks), rather than computing the full ranking (including a large number of documents which will never be visited), and to extend this partial ranking on demand. The *data stream approach* presented by Pfeiffer&Fuhr (1995) will be adapted to fuzzy quantifiers to incorporate this strategy into the RA system.

4 Scalable multi-tier architecture

The CORBA standard has been chosen to declare and implement all interfaces of the search assistant. It permits us to build the system in distributed component technology, independent of (networked) location, formalisms and platforms.

In addition, it facilitates the construction of scalable multi-tier architectures and the integration of mechanisms for load-balancing. The multi-tier architecture of the RA is displayed in Fig. 2. The main system components are the thin user front-end (TUF) which implements the graphical user interface, the fuzzy query interpreter (FQI) which carries out the computationally demanding result aggregation, the wrappers which translate the subqueries

[3] As usual, only those documents with relevance $\widetilde{Q}(W, R_d) > 0$ will be presented to the user.

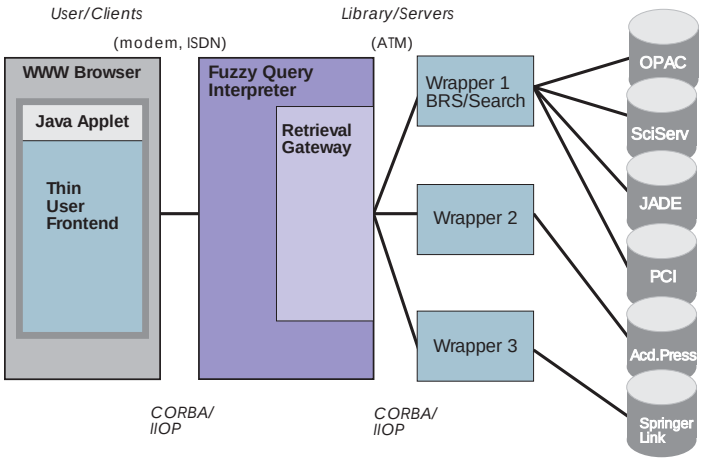

Figure 2: Multi-tier architecture of the RA system

generated by the FQI, and the database servers, which manage the document records and/or full-text data. The use of a separate component for query decomposition and fuzzy aggregation disburdens the user's computers because the TUF only needs to display precomputed results; it can focus on result presentation and the task of user interfacing (context-specific help etc.). The database servers are disburdened by the FQI because they only have to deliver results for very elementary queries (for one search term). The whole task of (fuzzy) result aggregation is covered by the FQI. Multiple copies of the FQI can be run on several host systems in order to meet requirements on response time when operated by a large number of simultaneous users. The RA system uses high-speed network connections (ATM) between the FQI and the database servers in order to ensure fast delivery of elementary results in the presence network traffic.

5 Discussion

Fühles-Ubach (1997) has conducted a comparative study on fuzzy information systems. As noted there, only few techniques from fuzzy set theory have matured to commercial products. This is mainly attributed to their *lack of operability*: user interfaces of these systems are often too complex; and to their *unacceptable response times*, caused by techniques which are not scalable. Other typical drawbacks of fuzzy retrieval systems are their *lack of integration* (proof of concept, but no embedding in a complete solution), and the *lack of comparability*, because the results are often not evaluated according to international standards. In designing the RA, we have taken pains to circumvent these typical shortcomings. As regards *scalability*, we have waived those techniques of fuzzy retrieval which are computationally too demanding (e.g., linguistic terms), and decided to utilise a multi-tier architecture which permits for load-balancing. Scalability will be evaluated on

a data set of more than 20 Mio. document records. The *ease of use* is mainly ensured by the rich yet meaningful set of fuzzy search operators, which emulate operators known from natural language (operability will be evaluated through user interviews). The RA is designed as a *complete system* which includes several databases, and will also embed further services, e.g. electronic document order and delivery via JASON. *Openness* and *interoperability* are achieved by utilising latest interface technology, and will be evaluated by integrating heterogeneous databases (see Fig. 2). The system has currently passed the design phase; special attention has been paid to the retrieval methodology, in order to foreclose problems with weighted retrieval from the outset. Once the implementation of RA is available (ca. 12/2000), we will complement this analysis by a statistical comparison with other approaches to weighted retrieval and fuzzy quantification, based on standardised tests.

References

BARWISE, J. and COOPER, R. (1981): Generalized Quantifiers and Natural Language. *Linguistics and Philosophy*, 4, 159-219.

BORDOGNA, G. and PASI, G. (1997): A Fuzzy Information Retrieval System Handling Users' Preferences on Document Sections. In Dubois, D. and Prade, H. and Yager, R.R. (Eds.), Fuzzy Information Engineering. Wiley, 265-281.

BUELL, D.A. (1982): An Analysis of Some Fuzzy Subset Applications to Information Retrieval Systems. *Fuzzy Sets and Systems*, 7, 35-42.

CATER, S.C. and KRAFT, D.H. (1987): TIRS: A Topological Information System Satisfying the Requirements of the Waller-Kraft Wish List. *10th ACM-SIGIR Conf. on Research and Development in Information Retrieval*, 171-180.

FÜHLES-UBACH, S. (1997): Analysen zur Unschärfe in Datenbank- und Retrievalsystemen. Doctorial Dissertation, Humboldt-Universität, Berlin.

GLÖCKNER, I. (1997): DFS – An Axiomatic Approach to Fuzzy Quantification. Technical Report TR97-06, Technische Fakultät, Universität Bielefeld.

GLÖCKNER, I. (1999): A Framework for Evaluating Approaches to Fuzzy Quantification. Technical Report TR99-03, Technische Fakultät, Universität Bielefeld.

GLÖCKNER, I., KNOLL, A. and WOLFRAM, A. (1998): Data Fusion based on Fuzzy Quantifiers. In *Proceedings of EuroFusion '98*, Great Malvern, UK.

PFEIFER, U. and FUHR, N. (1995): Efficient Processing of Vague Queries using a Data Stream Approach. *Proc. of the 18th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, ACM, New York, 189-198.

WALLER, W.G. and Kraft, D.H. (1979): A Mathematical Model of a Weighted Boolean Retrieval System. *Information Processing & Management*, 15, 235-245.

WIEDERHOLD, G. (1992): Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3), 38-49.