Personalizing XML Full Text Search in PIMENTO

Sihem Amer-Yahia[1], Irini Fundulaki[2] and Laks Lakshmanan[3]

[1] Yahoo! Research
USA
sihem@yahoo-inc.com
[2] ICS-FORTH
Greece
fundul@ics.forth.gr
[3] University of British Columbia
Canada
laks@cs.ubc.ca

XML search is increasing in popularity as more and larger XML repositories are becoming available. The accuracy of XML search varies across different systems and a lot of effort is put into designing scoring functions tailored to specific datasets. For example, the INEX effort [8] aims at improving the search relevance of IEEE XML data collections. To the best of our knowledge none of the existing XML search solutions incorporates *user information* to determine relevant query answers. In PIMENTO we argue that there is no scoring function that can meet all user-related information and advocate the idea of incorporating *user profiles* into XML search in order to customize query answers and improve search quality.

Personalization is used in a variety of applications: in telecommunications it is used to direct user calls based on the caller context, in Web search the ranking of query answers may be modified using the user's navigational and search patterns. In the relational database context, query personalization has been studied extensively [6,12] and shown to be effective in practice.

In PIMENTO a user profile is composed of two kinds of preference rules: *scoping rules* and *ordering rules*. Scoping Rules are used to expand or restrict the original query result. Ordering Rules are combined with query scoring to customize the ranking of query answers, hence overriding the ranking strategy of the query engine.

Query personalization in PIMENTO is defined as the process of rewriting a user query using scoping rules and ranking query answers using ordering rules. Enforcing scoping rules is not straightforward: there can be a large number of rewritings of the user query when scoping rules are considered. To enforce efficiently scoping rules we take into account existing query relaxation work [2,15]. A key contribution of our approach is that scoping rules can be incorporated into a single query plan without requiring actual query rewriting.

Ultimately, the user is only interested in the top several answers. Consequently, understanding how to combine user profiles with topk processing is a key aspect of efficient query personalization. A core contribution of PIMENTO

Dagstuhl Seminar Proceedings 08111
Ranked XML Querying
http://drops.dagstuhl.de/opus/volltexte/2008/1534

is the formalization of query processing in an algebra and the definition of an ordering rules-aware topk operation that achieves effective pruning while guaranteeing soundness of query evaluation, i.e., always return the correct topk answers. Even if their query score is low, user-preferred answers should not be pruned. The introduction of ordering rules requires to revisit well-established topk pruning conditions such as the threshold algorithm defined in [7].

References

1. R. Agrawal and et. al. A framework for Expressing and Combining Preferences. In *SIGMOD*, 2000.
2. S. Amer-Yahia and et. al. FleXPath: Flexible Structure and Full-Text Querying for XML. In *SIGMOD*, 2004.
3. S. Borzsonyi and et. al. The Skyline Operator. In *ICDE*, 2001.
4. N. Bruno and et. al. Evaluating Top-k Queries over Web-Accessible Databases. In *ICDE*, 2002.
5. S. Chaudhuri and et. al. Evaluating Top-k Selection Queries. In *VLDB*, 1999.
6. J. Chomicki. Preference formulas in relational queries. *ACM TODS*, 28(4):427–466, 2003.
7. R. Fagin and et. al. Optimal Aggregation Algorithms for Middleware. In *PODS*, 2001.
8. Initiative for the Evaluation of XML Retrieval. http://inex.is.informatik.uni-duisburg.de:2004/.
9. W. Kiessling. Foundations of Preferences in Database Systems. In *VLDB*, 2002.
10. W. Kiessling and et. al. Preference XPATH: A Query Language for E-Commerce. In *Konferenz fur Wirtschaftsinformatik*, 2001.
11. G. Koutrika and et. al. Rule-based query personalization in digital libraries. *IJDL*, 4(1):60–63, 2004.
12. G. Koutrika and et. al. Personalized Queries under a Generalized Preference Model. In *ICDE*, 2005.
13. C. Li and et. al. RankSQL: query algebra and optimization for relational top-k queries. In *SIGMOD*, 2005.
14. A. Marian and et. al. Adaptive Processing of Top-k Queries in XML. In *VLDB*, 2005.
15. Torsten Schlieder. Schema-Driven Evaluation of Approximate Tree-Pattern Queries. In *EDBT*, 2002.