

Summary of Research Activities

Ralf Schenkel

Max-Planck-Institut für Informatik
Saarbrücken, Germany
<http://www.mpi-sb.mpg.de/~schenkel>
schenkel@mpi-sb.mpg.de

1 Motivation and Background

The world has recently seen a tremendous proliferation in the use of XML for data exchange. In addition, XML is increasingly used as a replacement for HTML on the Web or in large intranets. As XML provides means to structure data and tag it with rich annotations, retrieving information in the XML-based Web is radically different from today's widely used keyword-based search engines. It is also radically different from querying XML databases with a fixed and known schema. Major challenges are (1) the heterogeneity of data on the Web, in intranets, and in federations of digital libraries and other loosely coupled XML sources, and (2) the distribution of information across several documents.

Heterogeneity of data is inevitable as there is no universal standard for representing arbitrary data in XML (and it is unlikely that there will ever be such a standard), so schemas used to represent data widely vary across different data sources, and some sources do not provide a schema at all. Widely adopted query languages for XML like XPath and XQuery are no longer appropriate for searching in such an environment as they cannot cope with the diversity of data. This calls for a new query paradigm in between the powerful, but complex and schema-dependent XQuery and the limited, keyword-based search that today's Web search engines provide. As users don't know (and typically don't care about) how the schema looks like, queries should not explicitly specify the structure of the data, but express the "information need" of users, a kind of "find what I mean" approach. Queries should express the user's guess how the data may be structured, and it's the system's task to find documents that match this guess. This definitely calls for a **ranked retrieval** approach returning results in descending order of estimated relevance, in contrast to the Boolean approach of existing XML query languages.

From a research point of view, this entails, among other things, the application of both structural and ontological similarity measures to match documents and queries. Additionally, as the system's notion of good results may not coincide with the user's, **relevance feedback** must be a core part of a system to establish user-aware instead of merely system-induced similarity measures. Finally, as the introduction of similarities highly increases the number of potentially relevant results, any efficient system must apply algorithms for query evaluation that are optimized to compute the most relevant results first.

Another major challenge for XML retrieval is raised by information being spread over several linked documents. An XML search engine should treat elements that are referenced through links similarly to "normal" child elements when evaluating path expressions in queries, which is typically done using some path indexing technique. However, two problems arise when taking links into account: (1) Links change the structure of XML documents, so they are no longer trees, but form a directed graph; (2) links generate interconnections of previously unconnected XML documents, yielding large sets of connected elements with long paths between them. While the latter problem can lead to path indexes that grow extremely large and take very long to build, the former even renders some of the established and highly efficient path indexes unusable.

2 Important Research Issues

The driving theme of my research has been *efficient and effective retrieval of information in heterogeneous XML collections*. Towards this goal, I have focused on the following important research issues:

- (1) Quantified ontologies, i.e., ontologies with a statistics-based notion of similarity between concepts, together with an algorithm to disambiguate the meaning of polysems within a given context.
- (2) Index support for efficient connectivity tests and evaluation of queries with the **descendants** axis on large and highly interlinked document collections.
- (3) Rank-aware query evaluation, i.e., the system should not materialize all results (including the marginally relevant ones), but compute only the most relevant results.
- (4) Advanced query languages and scoring models for heterogeneous and linked XML collections.
- (5) Relevance feedback for XML beyond the existing, document-level feedback techniques, based on user assessment of how well the retrieved results for the query matched the information need.

The following section gives a short overview on my work in these areas. What I describe here is joint work with Gerhard Weikum and several PhD students working in our group who I partly guide.

3 Selected Results

Quantified and Instance-Oriented Ontologies. While there are many proposed models for ontologies and some readily available ontologies like WordNet, they hardly provide a quantification for the similarity of related concepts. We developed such a quantification [6] that derives the similarity of directly neighbored concepts in the ontology graph from the correlation of the terms belonging to these concepts in a large Web crawl. This "edge-level" similarity is then transitively extended to all pairs of concepts using a variant of Dijkstra's shortest path algorithm, which has the nice property that, given a concept, it delivers related concepts in descending order of similarity. It is therefore sufficient to store the edge-level similarities and compute all others incrementally at runtime. We apply this ontology model in the XXL [9] and TopX [13, 14] search engines for XML retrieval.

Another problem with existing ontologies is that they hardly provide instances of concepts (e.g., names of actors or companies) or values of the properties of concepts (e.g., models or makers of cars). This information is partly available on the Web, e.g., as the content of HTML tables or form field menus. We adopted a collection of algorithms from the literature to extract (concept, instance) and (concept, property, value) tuples from Web pages, together with their frequency and information about the correlation of different tuples. From these statistics we derived fuzzy dependency rules which we used to query Deep Web sources and answer concept-based queries [2].

Path Indexes for XML. We developed two path indexes for XML that efficiently support evaluating queries with the **descendants** axis and are suitable for large, interlinked collections. The *HOPi index* [7, 8] leverages the existing concept of a two-hop cover for a directed graph for highly efficient indexing of connections in XML collections. Here, our main technical contributions are a structurally recursive divide-and-conquer algorithm for scalable index building, efficient index maintenance and distance support.

However, in a ranked retrieval setting, HOPi is not always the ideal choice as it does not support incremental delivery of results, which would be needed for a top- k algorithm. I therefore proposed the *FliX framework* [5] that uses existing indexing techniques as building blocks. It combines structurally similar parts of the collection, builds an index for each subcollection, and incrementally answers queries by "travelling" through the subcollections.

Evaluation of Top-k Queries. We extended Fagin’s Threshold Algorithm with sorted access for computing the most relevant results to a partial-match query on multidimensional data with a probabilistic prediction mechanism for the score of a candidate result, based on estimated value distributions for the dimensions used in a query. We approximated these distributions with Poisson distributions and histograms. Our approach significantly reduces the number of sorted accesses to the database and returns an approximate top-k result. Unlike other work on approximate ranking algorithms, we give probabilistic guarantees on the (small) error relative to ”exactly top-k” queries [12]. Our *TopX* search engine can dynamically expand ontological similarity conditions (using the algorithm mentioned above) [13] and evaluate content-and-structure queries on XML [14]. On the efficiency side, we developed a cost-based probabilistic scheduling strategy which we showed to be close to an empirical lower bound for any such algorithm in a text retrieval application [1].

Current work in this area aims at (1) developing better scoring functions beyond the standard Okapi BM25 model, including term proximity and element distances, (2) extending the top-*k* algorithm to support linked documents, and (3) including structural similarity for XML to support vague structural search.

Advanced query languages and scoring models. Similarly to SQL in the database world, existing query languages for XML, like XQuery and the XXL language developed in our group, are way too complicated for end users. At the other extreme end of the spectrum, simple keyword-based search cannot exploit the rich structure of XML. It is evident that we need a new query paradigm with an expressiveness between these extremes. The SphereSearch Engine [3] that has been developed in our group is a first step in that direction. SphereSearch makes use of linguistic tools to annotate certain classes of information (like persons and locations) in the XML collection. A user can then exploit the annotations to formulate queries in addition to standard keyword conditions. SphereSearch additionally allows to group query conditions, corresponding to the different entities they refer to, and to specify simple join conditions.

On the modeling side, SphereSearch introduced a new scoring model where the score of an element with respect to a content condition is accumulated from other elements in an environment, including elements that can be reached only through one or more link traversals. The result of a complex query with multiple groups is a list of Steiner trees formed by answers to each condition, ranked by an aggregation of the groups’ scores and the compactness of the Steiner tree. The engine includes a top-*k* algorithm for an efficient evaluation.

Relevance Feedback. Relevance feedback has been used for a long time in traditional IR to improve search results. However, this previous work cannot be directly applied to XML retrieval, because queries are much more complex than simple keyword queries and results are no longer complete documents, but document fragments. We were among the first who made the transition from simplistic keyword-based methods towards “real” XML relevance feedback [10, 11]. Given a keyword query and some initial results with user feedback, our approach computes a new, now structural query with content and structural constraints. Experiments with the INEX benchmark collection have shown that this approach outperforms keyword-based approaches.

We are currently working on a framework for feedback-driven XML retrieval [4] that includes adaptive reweighing of query conditions and personalized ontologies. Important aspects to consider include (1) the granularity for feedback (documents, elements, or paths), (2) a theoretical model for refining and expanding structural queries with new content and structural constraints, (3) a methodology to compare the effectiveness of different approaches for XML relevance feedback, and (4) finding an appropriate GUI for efficiently giving feedback.

References

1. H. Bast, D. Majumdar, R. Schenkel, M. Theobald, and G. Weikum. IO-Top- k : Index-optimized top- k query processing. In G. Alonso et al., editors, *32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 475–486, September 2006.
2. J. Graupmann, J. Cai, and R. Schenkel. Automatic query refinement using mined semantic relations. In *ICDE Workshop on Challenges in Web Information Retrieval and Integration (WIRI)*, 2005.
3. J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch Engine for unified ranked retrieval of heterogeneous XML and Web documents. In *31st International Conference on Very Large Databases (VLDB 2005)*, Trondheim, Norway, 2005. Morgan Kaufmann.
4. H. Pan, A. Theobald, and R. Schenkel. Query refinement by relevance feedback in an XML retrieval system. In *23rd International Conference on Conceptual Modeling (ER 2004)*, volume 3288 of *LNCS*, pages 854–855. Springer, 2004.
5. R. Schenkel. *FliX*: A flexible framework for indexing complex XML document collections. In *1st Int. Workshop on Database Technologies for Handling XML Information on the Web*, 2004.
6. R. Schenkel, A. Theobald, and G. Weikum. Ontology-enabled XML search. In H. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, editors, *Intelligent Search on XML Data*, volume 2818 of *Lecture Notes in Computer Science*, pages 119–131. Springer, Sept. 2003.
7. R. Schenkel, A. Theobald, and G. Weikum. HOPI: An efficient connection index for complex XML document collections. In *9th Int. Conference on Extending Database Technology (EDBT 2004)*, Heraklion, Greece, volume 2992 of *Lecture Notes in Computer Science*, pages 237–255. Springer, 2004.
8. R. Schenkel, A. Theobald, and G. Weikum. Efficient creation and incremental maintenance of the HOPI index for complex XML document collections. In *21st International Conference on Data Engineering (ICDE 2005)*, Tokyo, Japan, 2005. IEEE Computer Society.
9. R. Schenkel, A. Theobald, and G. Weikum. Semantic similarity search on semistructured data with the xsl search engine. *Information Retrieval*, 2005. in press.
10. R. Schenkel and M. Theobald. Feedback-driven structural query expansion for ranked retrieval of XML data. In *10th International Conference on Extending Database Technologies (EDBT 2006)*, Munich, Germany, Mar. 2006.
11. R. Schenkel and M. Theobald. Structural feedback for keyword-based XML retrieval. In M. Lalmas, A. MacFarlane, S. M. Rger, T. Tombros, Anastasios Tsikrika, and A. Yavlinsky, editors, *28th European Conference on Information Retrieval (ECIR 2006)*, pages 326–337, London, UK, Apr. 2006.
12. M. Theobald, R. Schenkel, and G. Weikum. Top- k query evaluation with probabilistic guarantees. In *30th Int. Conference on Very Large Databases (VLDB 2004)*, Toronto, Canada, 2004.
13. M. Theobald, R. Schenkel, and G. Weikum. Efficient and self-tuning incremental query expansion for top- k query processing. In *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, Salvador, Brazil, 2005. Sheridan Printing.
14. M. Theobald, R. Schenkel, and G. Weikum. An efficient and versatile query engine for TopX search. In *31st International Conference on Very Large Databases (VLDB 2005)*, Trondheim, Norway, 2005. Morgan Kaufmann.