ArHeX: An Approximate Retrieval System for Highly Heterogeneous XML Document Collections

Ismael Sanz[1], Marco Mesiti[2], Giovanna Guerrini[3], Rafael Berlanga Llavori[1]

(1) Universitat Jaume I, Castellón, Spain - {berlanga,Ismael.Sanz}@uji.es
(2) Università di Milano, Italy - mesiti@dico.unimi.it
(3) Università di Genova, Italy - guerrini@disi.unige.it

1 Introduction

Handling the heterogeneity of structure and/or content of XML documents for the retrieval of information is a fertile field of research nowadays. Many efforts are currently devoted to identifying approximate answers to queries that require relaxation on conditions both on the structure and the content of XML documents [1, 2, 4, 5]. Results are ranked relying on score functions that measure their quality and relevance and only the top-k returned.

Current efforts, however, are still based on some forms of homogeneity on the structure of the documents to be retrieved. The parent-child or ancestor descendant relationship among elements should be still preserved, and the problem of similarity at the tag level (whose solution often requires the use of an ontology) is seldom considered [6, 8]. Consider for example, two entity types `Book` and `Author` that are bound by the many-to-many `Write` relationship. Many XML representations are possible. Someone can model books documents by starting from the `Book` entity type and listing for each book its authors. Others can model books documents by starting from the `Author` entity type and listing for each author the books she wrote. Current approaches miss to find relevant solutions in collections containing both kinds of documents because they can relax the structural constraint (`book/author` becomes `book//author`) but they are not able to invert the relationship (`book/author` cannot become `author/book`). A more general problem is that current systems [3] support only a specific similarity function on XML documents, while in practice the concept of "similarity" strongly depends on the requirements of each particular application. This makes it difficult, if not impossible, to tailor these systems to particular requirements.

In this paper we present *ArHeX*, a system for approximate retrieval in the context of highly heterogeneous XML document collections. Our system is designed to support different similarity functions, including lexical (i.e., tag-oriented) and structural conditions in order to handle a wide variety of heterogeneous collections. In ArHex, a user can specify the pattern of data to be retrieved through a graphical interface. Moreover, she can specify *mandatory constraints* on some relationships among elements or on the element tags that should be preserved. By means of specifically tailored indexing structures and heuristics, ArHex is

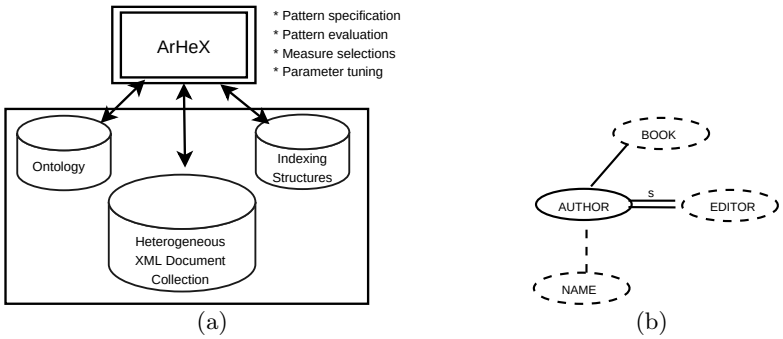

Figure **1.** (a) ArHeX architecture, (b) sample pattern

able to efficiently identify the approximate answers for the specified retrieval query ranked according to a similarity measure. Several parameters can be set and used to tune the behavior of the system to the application scenario in which it is employed.

2 ArHex System

ArHeX allows users to specify a suitable similarity measure for their collection, combining lexical and structural conditions. The lexical measures range from simple techniques based on the overlap of substrings to ontology-based measures. Indexes are tailored to the required measure for an efficient computation, using an inverted file-like structure. A peculiarity of our index is that we do not have an entry for each tag in the collection, but a normalization process is performed to group together similar tags relying on the tag similarity function preferred by the user.

ArHex also supports a set of similarity measures that can be employed in the selection and ranking of query results. The considered measures range from standard information retrieval measures (e.g. occurrence of query tags) to more sophisticated ones (e.g. structure based or sibling order based functions).

The developed system is equipped with the following functionalities.

– *Pattern specification.* The structures of user queries are represented as *patterns* in our system. A pattern is a graph in which the user can specify a "preference" in the parent-child, ancestor-descendant and sibling relationships existing among elements or on the tags of elements (depicted through dashed lines in the graphical representations). "Preference" means that higher scores are given to query answers presenting such a structure but also results that do not (or only partially) present such a structure are returned. Moreover, a user can specify stricter constraints that must occur in the returned results. Our constraints are classified in 3 categories: ancestor-descendant, same level, and tag constraints (as detailed in [7]). Figure 1(b) shows an example of pattern in which we search for books having an author, editor and name

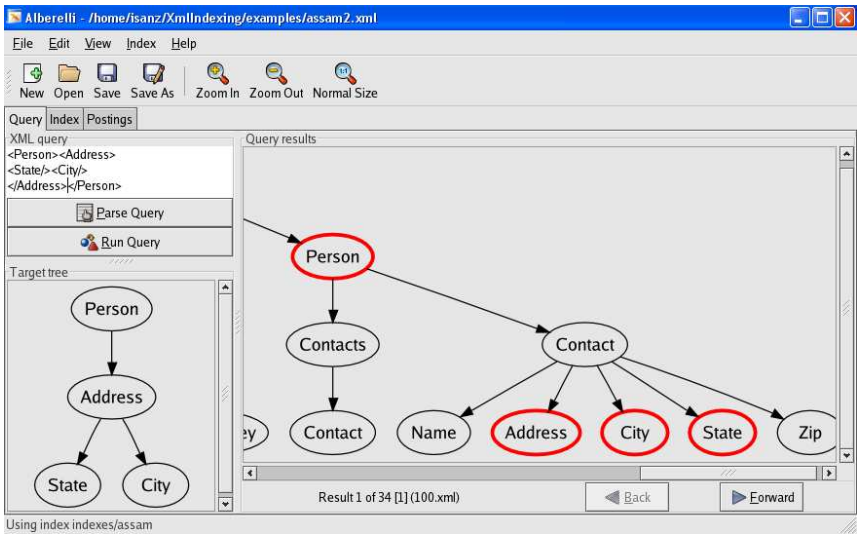

Figure **2.** ArHeX pattern evaluation facility

elements. The `book` element can be the parent or the child of the `author` element. The `name` element can be the child of the `author` element but can also appear in other positions. The `editor` element should be found in the same level of the `author` element. In ArHeX patterns are specified through a graphical interface and then mapped in an XML document.

– *Pattern evaluation.* The evaluation of a pattern in the collection is performed in different steps (details in [7]). First, through the inverted index, a *pattern index* organized in levels is generated containing the elements in the collection whose tags are similar to those in the pattern. Then, *fragments* are generated by considering the parent-child and ancestor-descendant relationships among elements in the collection. Furthermore, through the use of a similarity measure and the heuristic *locality principle* [6], fragments are combined in regions when the similarity of the pattern with respect to a region is higher than that with respect to each single pattern. Mandatory constraints are checked both during fragment and region construction depending on the category they belong to. Whenever a constraint is not met the corresponding fragment/region can be dropped or penalized according to user preferences. Finally, the similarity measure is employed to rank the top-k results. Figure 2 shows the evaluation of a pattern pointing out a similar region.

– *Measure selection.* Different measures can be applied for the evaluation of similarity between a pattern and a region depending on the application domain. ArHeX allows the selection from a set of predefined measures and the combination of existing ones. Finally, in the evaluation of a pattern in the collection, a user can visualize the differences of evaluation obtained through a subset of the considered measures.

– *Parameter tuning.* A user can tune the behavior of ArHeX to a specific scenario through a set of parameters that a graphical interface offers. For example, a user can specify the kind of tag similarity to employ (syntactic, semantic or both). Moreover, she can specify an extra weight to assign to elements in the pattern that are not found in similar regions or she can state when regions that do not meet the mandatory constraints should be dropped or penalized (and the weight to apply as penalty in the last case).

3 The Demonstration

The demonstration will show the following features:

Specification of user-defined similarity measures. The system includes a library of component-based lexical and structural similarity functions, which can be tailored to the user's needs. We will demonstrate the definition of tailored measures.

Queries on different real and synthetic collections of documents. The performance of similarity-based queries using the graphical interface will be presented, using different real and synthetic collections of documents. Different similarity measures will be exercised showing the precision and recall results.

Comparison of different measures. The system supports the interactive exploration of heterogeneous collection by allowing the use of several distinct similarity measures, in order to compare the results.

References

1. Amer-Yahia, S., Cho, S., Srivastava, D.: Tree Pattern Relaxation. EDBT. LNCS(2287). (2002) 496–513.
2. S. Amer-Yahia, N. Koudas, A. Marian, D. Srivastava, D. Toman. Structure and Content Scoring for XML. VLDB. (2005) 361–372.
3. G. Guerrini, M. Mesiti, I. Sanz. An Overview of Similarity Measures for Clustering XML Documents. Chapter in A. Vakali and G. Pallis (eds.), Web Data Management Practices: Emerging Techniques and Technologies. Idea Group.
4. A. Marian, S. Amer-Yahia, N. Koudas, D. Srivastava. Adaptive Processing of Top-k Queries in XML. ICDE. (2005) 162–173.
5. A. Nierman, H.V. Jagadish. Evaluating Structural Similarity in XML Documents. WebDB. (2002) 61–66.
6. I. Sanz, M. Mesiti, G. Guerrini, R. Berlanga Llavori. Approximate Subtree Identification in Heterogeneous XML Documents Collections. XSym. LNCS(3671). (2005) 192–206.
7. I. Sanz, M. Mesiti, G. Guerrini, R. Berlanga Llavori. Approximate Retrieval of Highly Heterogeneous XML Documents. Tech. report. University of Milano. (2005).
8. A. Theobald, G. Weikum. The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking. EDBT. LNCS(2287). (2002) 477–495.