Querying XML Using Structures and Keywords in Timber

Cong Yu
Department of EECS
University of Michigan
congy@eecs.umich.edu

H. V. Jagadish
Department of EECS
University of Michigan
jag@eecs.umich.edu

Dragomir R. Radev
School of Information
University of Michigan
radev@umich.edu

ABSTRACT

This demonstration will describe how Timber, a native XML database system, has been extended with the capability to answer XML-style structured queries (e.g., XQuery) with embedded IR-style keyword-based non-boolean conditions. With the original structured query processing engine and the IR extensions built into the system, Timber is well suited for efficiently and effectively processing queries with both structural and textual content constraints.

1. INTRODUCTION

Timber [3] is a native XML database that is designed to provide efficient storage and query processing to XML data. Like relational database systems, it supports a declarative query language (XQuery), set-at-a-time query processing, and cost-based optimizing. Efficient structure join algorithms and access methods are implemented in the system to provide fast processing of XML-style structured queries. However, with more and more data being stored in XML format, the ability to query the textual content of the XML documents in IR style has become necessary to fully utilize the power of XML. Towards this end, we have extended Timber to support keyword based IR-style querying.

2. CHALLENGES

Introducing IR-style keyword-based searching against XML data poses several challenges to both traditional database systems and IR systems. The challenge for database system resides in that IR-style queries are not exact. Therefore, results from a query need to be ranked according to their relevance to the query concept (usually expressed in the form of a set of keywords). The current XQuery specification and various XML database implementations lack the support for this (A full-text requirement for XQuery has recently come out, please see [4]). The challenge for IR systems is two-fold. First, XML makes it possible for informed users to ask more precise queries by specifying the structure of the element where the information is contained. Second, the presence of structures within a document mandates the retrieval of information at a granularity smaller than the whole document.

3. STRUCTURED KEYWORD QUERY

To address the above issues, we have proposed IR extensions to an XML query algebra and language (XQuery) [1]. Integrated IR supports based on those ideas are built into Timber (e.g, keyword based free text search, result scoring and ranking, etc.) to demonstrate that XML data can be queried effectively using both structural and keyword-based textual conditions. In particular, an access method based on a stack-based algorithm is developed to efficiently retrieve and score the set of elements that are ancestors of textual elements (in Timber, the textual content of an element is stored as a textual element by itself) containing at least one of the user provided keywords. Those scored elements can be subsequently supplied to the next operator in the evaluation tree to be processed on other conditions (e.g., a structural join). Finally, a sort based on the score can be employed to rank the qualified elements and return only the top results according to the user's specification. The fulfillment of the structural requirements and the IR style scoring based on keywords are therefore fully integrated into one execution.

We will be presenting a demo of structured and keyword-based querying in Timber, using sample queries selected from INEX topics [2] and the W3C draft on XQuery full-text use cases [4]. We will show how the queries containing both keywords and structural requirement can be expressed in the evaluation plan format of Timber and how they are processed inside Timber step by step until a sequence of scored (using a user provided function) and ranked elements is returned.

4. REFERENCES

[1] S. Al-Khalifa, C. Yu, and H. V. Jagadish. Querying structured text in an XML database. In *SIGMOD*, San Diego, CA, June 2003.

[2] Initiative for the evaluation of XML retrieval (INEX). http://qmir.dcs.qmul.ac.uk/inex/.

[3] H. V. Jagadish, et. al. TIMBER: A native XML database. *The VLDB Journal*, 11(4):274–291, 2002.

[4] XQuery and XPath full-text requirements and use cases. http://www.w3.org/TR/2003/WD-xmlquery-full-text-requirements-20030214/.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007.