Indexing and Searching XML Documents Based on Content and Structure Synopses

Weimin He, Leonidas Fegaras, and David Levine

University of Texas at Arlington, CSE
Arlington, TX 76019-0015
{weiminhe,fegaras,levine}@cse.uta.edu

Abstract. We present a novel framework for indexing and searching schema-less XML documents based on concise summaries of their structural and textual content. Our search query language is XPath extended with full-text search. We introduce two novel data synopsis structures that correlate textual with positional information in an XML document and improves query precision. In addition, we present a two-phase containment filtering algorithm based on these synopses that improves the searching process. Our experimental evaluation shows that our data synopses indexing scheme outperforms the standard XML indexing scheme based on inverted lists; the query evaluation based on our data synopses is more accurate than related approximate approaches that do not consider positional information; our two-phase containment filtering algorithm is more efficient than a single-phase brute force algorithm.

1 Introduction

As XML has become the *de facto* form for representing and exchanging data, there is an increasing interest in indexing and searching text-centric XML documents. Recently, XML query languages, such as XPath and XQuery, have been extended with full-text search capabilities. These queries are potentially more precise than simple IR-style keyword-based queries, not only because each search keyword can be associated with a structural context, which is typically the path to reach the keyword in a document, but structural constraints can also be used to specify the structural relationship between multiple search keywords.

Consider, for example, the running query Q used throughout the paper:

```
//auction//item[location ~ "Dallas"]
   [description ~ "mountain" and "bicycle"]/price
```

against a pool of indexed XML documents. It searches for the prices of all auction items located in Dallas that contain the words "mountain" and "bicycle" in their description. When searching for documents that satisfy this query, we do not want to waste any time by considering those that do not match the structural constraints of the query or those that do not contain the search keywords at relative positions as specified by the structural relationships in the query. For

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 58–69, 2007.

example, we do not want to consider a document that, although has items located in Dallas, none of these items has both "mountain" and "bicycle" in their descriptions, even though there may be other items in this document, which are not located in Dallas but have both "mountain" and "bicycle" in their titles.

Current XML indexing techniques, such as [6], combine structure indexes and inverted lists extracted from XML documents to fully evaluate a full-text query against these indexes and return the actual XML fragments that answer the query. This is accomplished by performing containment joins over the sorted inverted lists derived from the element and keyword indexes. Since all elements and keywords have to be indexed, such indexing schemes may consume a considerable amount of disk space and may be time-consuming to build. More importantly, the query evaluation based on these indexes may involve many joins against very long inverted lists that may consider many irrelevant documents at the early stages. Although many sophisticated techniques have been proposed to improve these joins by skipping the irrelevant parts of these lists, it is still an open research problem to make them effective for a large document pool.

In this paper, we present a new framework for indexing and searching schemaless XML documents based on condensed summaries extracted from the structural and textual content of the documents. Instead of indexing each single element or term in a document, we extract a structural summary and a small number of data synopses from the document, which are indexed in a way suitable for query evaluation. The result of a query evaluation is a list of document locations that best match the query. A document location includes meta information about the document, such as the document URL, structural summary, and description. Based on the retrieved meta information, the client can choose some of the returned document locations and request a full evaluation of the query over the chosen documents using any existing XML query engine and return the XML fragments as query answers. To find all indexed documents that match the structural relationships in a query, the *query footprint* is extracted from the query and is converted into a pipelined plan to be evaluated against the indexed structural summaries. The resulting document locations that match the query footprint are further filtered out using the data synopses associated with the search predicates in the query and returned to the client.

2 Related Work

There is an increasing interest in recent years for full-text search over XML documents. Khalifa *et al* [1] propose a bulk algebra called TIX, which integrates simple IR scoring schemes into a traditional pipelined query evaluator for an XML database. TeXQuery [2] supports a powerful set of fully composable full-text search primitives, which can be seamlessly integrated into the XQuery language. In [3], the authors present a framework that relaxes a full-text XPath query by dropping some predicates from its closure and scoring the approximate answers using predicate penalties. XRank [5] extends Google-like keyword search to XML. The authors propose an algorithm for scoring XML elements that