GalaTex: A Conformant Implementation of the XQuery Full-Text Language

Emiran Curtmola [†] Sihem Amer-Yahia [§] Philip Brown [§] Mary Fernández [§]

[†] University of California San Diego
9500 Gilman Drive
La Jolla, CA 92093
ecurtmola@cs.ucsd.edu

[§] AT&T Labs Research
180 Park Ave
Florham Park, NJ 07932
{sihem,pebrown,mff}@research.att.com

ABSTRACT

We describe GalaTex, the first complete implementation of XQuery Full-Text, a W3C specification that extends XPath 2.0 and XQuery 1.0 with full-text search. XQuery Full-Text provides composable full-text search primitives such as keyword search, Boolean queries, and keyword-distance predicates. GalaTex is intended to serve as a reference implementation for XQuery Full-Text and as a platform for addressing new research problems such as scoring full-text query results, optimizing XML queries over both structure and text, and evaluating top-k queries on scored results. GalaTex is an all-XQuery implementation initially focused on completeness and conformance rather than on efficiency. We describe its implementation on top of Galax, a complete XQuery implementation.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms: Experimentation, Standardization, Languages

Keywords: XQuery, Full-Text, Conformant prototype

1. INTRODUCTION

The ability to search both the structure and text content of XML documents is gaining importance with the increase of large XML repositories such as the US Library of Congress (LOC) documents, medical data in XML such as HL7, and the INEX (TREC for XML) data collection. Querying XML repositories rich in text content requires sophisticated full-text search features ranging from matching individual keywords to combining matches with Boolean operators and with word distances, stemming, and stop words.

XML querying is a well-studied topic, with several powerful database-style query languages such as XPath 2.0 [6] and XQuery 1.0 [5] set to become W3C standards. However, due to the fact that their data model does not represent words and their positions in input documents, XPath and XQuery provide only limited substring matching functions for text search.

XQuery Full-Text [7] is an extension of XPath and XQuery that supports fully composable full-text search primitives defined on a data model of words and positions. The language is highly inspired by TeXQuery [1], a proposal to the W3C Full-Text Task Force. XQuery Full-Text provides powerful full-text search primitives (called *FTSelections*) such as simple word search, Boolean queries, word distance as well as stemming, regular expressions and stop words. It also supports scoring and top-k ranking of query results. *FTSelections* are defined on a data model, called *AllMatches*, which represents words and their positions in documents. Because the semantics of each *FTSelection* is defined in terms of operators on the *AllMatches* data model, the *FTSelections* are fully composable. We refer the reader to the language specification [7] and the language use cases [8] for more details on the language.

XQuery Full-Text supports scoring and ranking of query results and permits any ranking method that satisfies the XQuery Full-Text scoring requirements [7, 9]. In GalaTex, we adapt the probabilistic relational algebra [4] to *AllMatches* by extending each full-text primitive with the ability to manipulate scores. Our implementation satisfies the XQuery Full-Text scoring requirements.

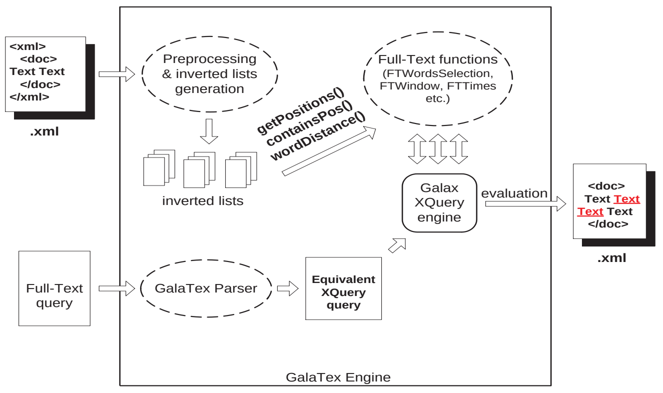
When implementing GalaTex, we have focused more on completeness and conformance than on efficiency. Thus, GalaTex can serve as a reference implementation of XQuery Full-Text and as a platform for experimenting new research ideas for scoring XML data, optimizing XML queries on both structure and content, and evaluating top-k queries.

2. GALATEX

Numerous strategies exist for implementing XQuery Full-Text – at least as many strategies as there are for implementing XQuery itself! Possible strategies include extending an existing XQuery engine with native support for the XQuery Full-Text data model and operators; extending an existing full-text search engine to serve as an XQuery Full-Text co-processor; or translating XQuery and XQuery Full-Text into another query language, such as SQL. XQuery Full-Text relies on the *AllMatches* data model that captures words and their positions. Regardless of the implementation strategy chosen, the key implementation problems are representing the *AllMatches* data model, implementing the semantics for each *FTSelection*, and making the word positions used in the input documents accessible to the *AllMatches* data model.

Because new languages benefit from the rapid development of experimental implementations, our strategy was to employ XML and XQuery directly to implement XQuery Full-Text.

The GalaTex architecture is depicted in Figure 1.



<xml> <doc> Text Text </doc> </xml>
.xml
Preprocessing & inverted lists generation
Full-Text functions (FTWordsSelection, FTWindow, FTTimes etc.)
getPositions() containsPos() wordDistance()
inverted lists
Galax XQuery engine
evaluation
<doc> Text Text Text Text </doc>
.xml
Full-Text query
GalaTex Parser
Equivalent XQuery query
GalaTex Engine

Figure 1: Architecture of GalaTex

In the upper left of Figure 1, GalaTex preprocesses input documents, and for each distinct word, produces one XML document containing all the positions of that word. Each position is encoded as Dewey values that record the depth-first node path from the XML document root to each node. These documents essentially contain *inverted lists* which map words to their positions.

In the lower left of Figure 1, GalaTex translates XQuery Full-Text queries into equivalent XQuery queries by mapping each *FTSelection* into a call to its corresponding XQuery function. The XQuery functions themselves (upper right of Figure 1) are implemented in an XQuery library module, where each function implements one *FTSelection* primitive. They take one or more *AllMatches* values and produce an *AllMatches* value.

An *AllMatches* value specifies all possible position solutions to a full-text search query and can be viewed as a propositional logic formula in disjunctive normal form [1]. We represent instances of the *AllMatches* data model using XML values.

Consider the following full-text query which returns those books containing paragraphs with words similar to *usability* and *software* case sensitive within ten words of each other:

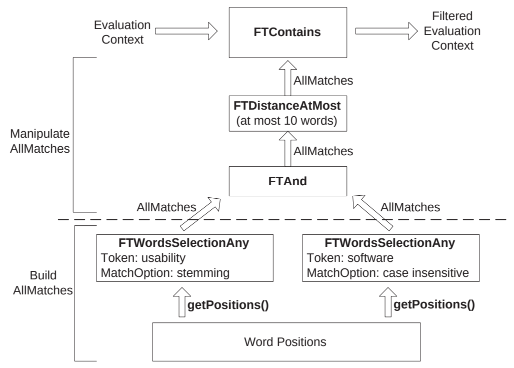
```
//book[.//p ftcontains
("usability" with stemming) && ("software" case sensitive)
with distance at most 10 words]/title
```

The parser generates this equivalent XQuery expression:

```
//book[( let $ec_1:= ( .//p ) return
 fts:FTContains( $ec_1,
  fts:FTWordDistance(-1, 10,
   fts:FTAnd(
    fts:FTWordsSelectionAny( $ec_1, "usability",
     fts:FTStemOption( "with stemming" ), "1"),
    fts:FTWordsSelectionAny( $ec_1, "software",
     fts:FTCaseOption( "case sensitive" ), "2")))))]/title
```

GalaTex also offers support for match options. They are modifiers that apply to each of the search words. Possible match options are *case sensitive, special characters, regular expressions, stemming, stop words, ignore elements, language selection, thesaurus, and diacritics* [7]. A match option has the effect of expanding one search word to a set of words that becomes the new set of search words for the current full-text query.

A possible evaluation plan for the given translation query is shown in Figure 2. A bottom-up evaluation of the plan builds *AllMatches* for the two search keywords which become inputs to *FTSelections*: *FTAnd* and *FTDistance*. Finally, *FTContains* filters the evaluation context and returns only those nodes that contain at least one match that satisfies all the constraints in the final *AllMatches*.



Evaluation Context
FTContains
Filtered Evaluation Context
AllMatches
FTDistanceAtMost (at most 10 words)
AllMatches
Manipulate AllMatches
FTAnd
AllMatches
AllMatches
FTWordsSelectionAny Token: usability MatchOption: stemming
FTWordsSelectionAny Token: software MatchOption: case insensitive
Build AllMatches
getPositions()
getPositions()
Word Positions

Figure 2: An XQuery Full-Text evaluation plan

GalaTex is implemented on top of the Galax XQuery engine [3]. On the right of Figure 1, Galax takes the input documents, the translated query and the library module of XQuery functions, evaluates the translated query and yields the result as an XML document. The final result contains the XML value in which the search words are highlighted. A demonstration of GalaTex and of the XQuery Full-Text use cases is available at *http://www.galaxquery.org/galatex/* and will also be available along with this poster.

3. CONCLUSION

We described GalaTex, the first conformant implementation of XQuery Full-Text that is able to query XML documents both on structure and text content. GalaTex uses XML and XQuery to implement XQuery Full-Text, which permits implementation on top of any existing XQuery engine. In addition to a command-line interface, GalaTex includes a browser interface that permits users to execute both the XQuery Full-Text use cases [8] and their own queries. Our strategy to implement the XQuery Full-Text language using XML and XQuery is general and expedient. Ultimately, we want GalaTex to be both complete *and* efficient. We explore improvements on full-text search and full-text scoring in [2].

4. REFERENCES

[1] S. Amer-Yahia, C. Botev, J. Shanmugasundaram. TeXQuery: A Full-Text Search Extension to XQuery. WWW 2004.
[2] E. Curtmola, S. Amer-Yahia, P. Brown, M. Fernández. GalaTex: A Conformant Implementation of the XQuery Full-Text Language. AT&T Technical Report, 2004.
[3] M. Fernández, J. Siméon, B. Choi, A. Marian, G. Sur. Implementing XQuery 1.0: The Galax Experience. VLDB 2003.
[4] N. Fuhr, T. Rölleke. A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. ACM TOIS 15(1), 1997.
[5] The World Wide Web Consortium. XQuery 1.0: An XML Query Language. W3C Working Draft. http://www.w3.org/TR/xquery/
[6] The World Wide Web Consortium. XML Path Language (XPath) 2.0. W3C Working Draft. http://www.w3.org/TR/xpath20/
[7] The World Wide Web Consortium. XQuery 1.0 and XPath 2.0 Full-Text. W3C Working Draft 09 July 2004. http://www.w3.org/TR/2004/WD-xquery-full-text-20040709/
[8] The World Wide Web Consortium. XQuery and XPath Full-Text Use Cases. W3C Working Draft 09 July 2004. http://www.w3.org/TR/xmlquery-full-text-use-cases/
[9] The World Wide Web Consortium. XQuery and XPath Full-Text Requirements. W3C Working Draft 02 May 2003. http://www.w3.org/TR/xmlquery-full-text-requirements/.