XML Search and XQuery Full-Text

Sihem Amer-Yahia
Yahoo! Research
Community Systems Group

Stanford guest lecture
Feb. 12th, 2007
Outline

• Motivation
• Challenges
• Languages
  – XQuery Full-Text
  – INEX
• Research overview
Motivation

• XML is able to represent a mix of structured and text information:
  – XML applications: digital libraries, content management.

• Need for a language to search XML documents
H. R. 2739

To address rising college tuition by strengthening the compact between the States, the Federal Government, and institutions of higher education to make college more affordable.

IN THE HOUSE OF REPRESENTATIVES

MAY 26, 2005

Mr. TIERNEY (for himself, Ms. McCOLLUM of Minnesota, Mr. GEORGE MILLER of California, Mr. KILDEE, Mr. EMANUEL, Mr. BISHOP of New York, Mr. PAYNE, Ms. WOOLSEY, Mrs. McCARTHY, Mr. WU, Mr. DAVIS of Illinois, Mr. GRIJALVA, Mr. MEEHAN, Mr. BECERRA, Mr. REYES, Mr. GONZALEZ, Ms. LINDA T. SÁNCHEZ of California, Mr. MCGOVERN, Ms. DELAURO, Mr. OWENS, Mr. HINOJOSA, Mr. KUCINICH, Mr. HOLT, Mr. CASE, Mr. VAN HOLLEN, Mr. ORTIZ, Mr. GUTIERREZ, Mr. CARDOZA, Mrs. JONES of Ohio, Ms. BALDWIN, Mr. WEXLER, Mr. BARROW, Mr. JEFFERSON, Mr. RYAN of Ohio, Ms. SOLIS, Ms. VELÁZQUEZ, and Ms. SCHAKOWSKY) introduced the following bill; which was referred to the Committee on Education and the Workforce

A BILL

To address rising college tuition by strengthening the compact between the States, the Federal Government, and institutions of higher education to make college more affordable.

Be it enacted by the Senate and House of Representatives of the United States of America in Congress assembled,
<bill bill-stage = "Introduced-in-House">
  <congress> 109th CONGRESS </congress>
  <session> 1st Session </session>
  <legis-num> H. R. 2739 </legis-num>
  <current-chamber> IN THE HOUSE OF REPRESENTATIVES </current-chamber>
  <action>
    <action-date date = "20050526"> May 26, 2005 </action-date>
    <action-desc><sponsor name-id = "T000266"> Mr. Tierney </sponsor> (for himself, and <cosponsor name-id = "M001143"> Ms. McCollum of Minnesota </cosponsor>, <cosponsor name-id = "M000725"> Mr. George Miller of California </cosponsor>) introduced the following bill; which was referred to the <committee-name committee-id = "HED00"> Committee on Education and the Workforce </committee-name>
    </action-desc>
  </action>
</bill>
109th

1st session

... Committee on Education ...

Mr. Jefferson

... and the Workforce
Outline

• Motivation
• Challenges
• Languages
• Research overview
Challenges: DB and IR

<congress>
109th

<session>
1st session

<action>

<action-desc>

<sponsor>

<co-sponsor>

TEXT

XPATH/XQUERY
IR engines
Challenges

• Searching over Structure+Text
  – express complex full-text searches and combine them with structural searches.
  – specify a search context and return context.

• Scores and Ranking
  – specify a scoring condition,
    • possibly over both full-text and structured predicates
  – obtain k best results based on query relevance scores
Motivation

• Current XML query languages are mostly “database” languages
  – Examples: XQuery, XPath

• Provide very rudimentary text/IR support
  – \texttt{fn:contains(e, keywords)}
  – Returns true iff element e contains keywords

• No support for complex IR queries
  – Distance predicates, stemming, …

• No scoring
Full-Text Task Force (FTTF) started in Fall 2002 to extend XQuery with full-text search capabilities: IBM, Microsoft, Oracle, the US Library of Congress.

First FTTF documents published on February 14, 2004. (public comments are welcome!):
http://www.w3.org/TR/xmlquery-full-text-use-cases/
http://www.w3.org/TR/xmlquery-full-text-requirements/

XQuery Full-Text highly influenced by TeXQuery.

Published a working draft describing the syntax and semantics of XQuery Full-Text on July 9, 2004. Latest version on May 1st 2006:
http://www.w3.org/TR/xquery-full-text/
Example Queries

• From XQuery Full-Text Use Cases Document
  – Find the titles of the books that contain the phrases “Usability” and “Web site” in this order, in the same paragraph, using stemming if necessary to match the tokens
  – Find the titles of the books that contain “Usability” and “testing” within a window of 3 words, and return them in score order

• Such queries are used, e.g. in legal applications
Related Work in IR

• XSEarch, XIRQL, JuruXML, XXL, ELIXIR
  – Not integrated with a powerful language for structured search, such as XQuery
  – Lack expressive power
  – No fully composable
  – Not easily extensible
XML FT Search Definition

- **Context expression**: XML elements searched:
  - pre-defined XML elements.
  - XPath/XQuery queries.

- **Return expression**: XML fragments returned:
  - pre-defined meaningful XML fragments.
  - XPath/XQuery to build answers.

- **Search expression**: FT search conditions:
  - Boolean keyword search.
  - proximity distance, scoping, thesaurus, stop words, stemming.

- **Score expression**:
  - system-defined scoring function.
  - user-defined scoring function.
  - query-dependent keyword weights.
Outline

• Motivation
• Challenges
  – XQuery Full-Text
  – INEX
• Research overview
Four Classes of Languages

- Keyword search
  "book xml"
- Tag + Keyword search
  book: xml
- Path Expression + Keyword search
  /book[./title about "xml db"]
- XQuery + Complex full-text search
  for $b in /book
  let score $s := $b ftcontains "xml" && "db" distance 5
**XML Search Languages**

**Keyword-only**
- Nearest concept *(Schmidt, Kersten, Windhouwer, ICDE 2002)*
- XRank *(Guo, Botev, Shanmugasundaram, SIGMOD 2003)*
- Schema-free XQuery *(Li, Yu, Jagadish, VLDB 2003)*
- INEX Content-Only queries *(Trotman, Sigurbjornsson, INEX 2004)*
- XKSearch *(Xu & Papakonstantinou, SIGMOD 2005)*

**Tag+Keyword**
- XSEarch *(Cohen, Mamou, Kanza, Sagiv, VLDB 2003)*

**Path+Keyword**
- XPath 2.0 *(http://www.w3.org/TR/xpath20/)*
- XIRQL *(Fuhr, Großjohann, SIGIR 2001)*
- XXL *(Theobald, Weikum, EDBT 2002)*
- NEXI *(Trotman, Sigurbjornsson, INEX 2004)*
TeXQuery and XQuery Full-Text

- Extends XPath/XQuery with fully composable full-text primitives.
- Scoring and ranking on all predicates.

2003
TeXQuery (AT&T Labs, Cornell U.)

Since 2004
IBM, Microsoft, LoC, Elsevier
Oracle, MarkLogic

XQuery Full-Text
Drafts

http://www.w3.org/TR/xquery-full-text/
XQuery in a Nutshell

- Input/Output: sequence of items
  - atomic types, elements, attributes, processing instructions, comments,...
- XPath core navigation language.
- Variable binding.
- Element construction.

**Example:**

Return books on XML indexing and ranking sorted by price:

```xml
for $item in //books/book
let $pval := $item//price
where fn:contains($item/title, "XML")
and fn:contains($item, "indexing")
and fn:contains($item, "ranking")
and $item/price < 50
order by $pval
return <result>{$item/title, $item//authors}</result>
```

- Sub-string operations: `fn:start-with()`, `fn:end-with()`
- No relevance ranking.
Syntax Overview

Two new XQuery constructs

- **FTContainsExpr**
  - Expresses “Boolean” full-text search predicates
  - Seamlessly composes with other XQuery expressions

- **FTScore**
  - Extension to FLWOR expression
  - Can score FTContainsExpr and other expressions
FTContainsExpr := FTWord | FTAnd | FTOr | FTNot | FTMildNot | FTOrder | FTWindow | FTDistance | FTScope | FTTimes | FTSelection (FTMatchOptions)*

books//section [ . ftcontains ("usability" with stemming occurs 4 times && "Software" case sensitive) window at most 3 ordered with stopwords ]

FTScore

for $b SCORE $s in FUZZY
    //books [ ./title ftcontains "XML" 0.4 and ./section ftcontains ("indexing" with stemming && "ranking" with thesaurus "synonyms") distance 5 and ./price < 50 ]

order by $s
return <result score="{$s}"> {$b/title, $b//authors} </result>
FTContainsExpr

• Like other XQuery expressions
  – Takes in sequences of items (nodes) as input
  – Produces a sequence of items (nodes) as output

• Can seamlessly compose with other XQuery expressions
ContextExpr \texttt{ftcontains} FTSelection

- ContextExpr (any XQuery expression) is context spec
- FTSelection is search spec
- Returns true iff at least one node in ContextExpr satisfies the FTSelection

**Examples**

- \texttt{//book ftcontains 'Usability' \&\& 'testing' distance 5}
- \texttt{//book[./content ftcontains 'Usability' with stems]/title}
- \texttt{//book ftcontains /article[author='Dawkins']/title}
FTSelection

• Encapsulates all full-text conditions in FTContainsExpr

• Works in a new data model called **AllMatch**
  – Operates on positions within XML nodes (more fine grained than XQuery data model)
  – Fully composable; similar to composition of relational (and XML) operators!

![Evaluation Diagram]

Evaluate to AllMatch
FTSelection Composability

• ‘Usability’
• /book[author='Dawkins']/title
• (‘Usability’ && /book[author='Dawkins']/title) same sentence
  • (‘Usability’ && /book[author='Dawkins']/title) same sentence window 5
• All of these evaluate to an AllMatch!
  – Allows arbitrary composition of full-text primitives
FTMatchOption

• Can be applied on any FTSelection to specify aspects such as stemming, thesauri, case, etc.
  – Fully composable with other context modifiers and FTSelections

• Examples
  – ‘Usability’ && ‘testing’ with stems
  – ‘Usability’ && ‘testing’ with stems window 5 without stop words
  – ‘Usability’ && ‘testing’ with stems window 5 without stop words case insensitive
In any order:

FOR $v$ [SCORE $s$]? [AT $i$]? IN [FUZZY] Expr
LET ...
WHERE ...
ORDER BY ...
RETURN

Example

FOR $b$ SCORE $s$ in /pub/book[. ftcontains “Usability” && “testing”]
ORDER BY $s$
RETURN <result score={$s$}> $b$ </result>
Example

FOR $b SCORE $s in FUZZY /pub/book[. ftcontains “Usability” && “testing”]
ORDER BY $s
RETURN <result score={$s}> $b </result>
Semantics Issues

- Evaluate to a Sequence of items
- Nest XQuery Expressions into XQuery
- FTSelection
- Nest XQuery Expressions into TeXQuery
- Evaluate to a FullMatch
FullMatch Overview

• FTSelections are fully composable
• Extensible with respect to new FTSelections
  – Only have to define semantics w.r.t. FullMatch
• Clean way to specify semantics of FTSelections
  – Like specifying semantics of relational operators
• Provides basis for optimizing complex queries
• FullMatch can be interpreted as a propositional formula over word positions in DNF
The usability of software measures how well the software provides support for quickly achieving specified goals. The users must not only be well-served, but feel well-served.
Sample Query

$doc ftcontains
('usability' with stems &&
'Rose')

window at most 10
('usability' with stems && 'Rose')

window at most 10
<book id='1000'><author>Elina Rose</author><content><p>The usability of software measures how well the software provides support for quickly achieving specified goals. </p><p>The users must not only be well-served, but must feel well-served.</p></content></book>
Semantics of FTStringSelection

'usability' with stems

FullMatch
  SimpleMatch
    StringInclude
      Token: usability
      Pos: 11
  SimpleMatch
    StringInclude
      Token: users
      Pos: 29

'rose'

FullMatch
  StringInclude
    Token: Rose
    Pos: 6
('usability' with stems && 'Rose')
window at most 10
Semantics of FTAndConnective

'usability' with stems

FullMatch

SimpleMatch

StringInclude
Token: usability
Pos: 11

SimpleMatch

StringInclude
Token: users
Pos: 29

×

FullMatch

SimpleMatch

StringInclude
Token: Rose
Pos: 6

'Rose'
Semantics of FTAndConnective

FullMatch

SimpleMatch

StringInclude
Token: usability
Pos:11

StringInclude
Token: Rose
Pos:6

StringInclude
Token: users
Pos:29

String Include
Token: Rose
Pos:6

'usability' with stems && 'Rose'
('usability' with stems && 'Rose')
window at most 10
Semantics of FTWindowSelection

Elina Rose

The usability of software measures how well the software provides support for quickly achieving specified goals. Users must be well-served, but must feel well-served.
Semantics of FTWindowSelection

('usability' with stems && 'Rose')
window at most 10
FullMatch Benefits

- FullMatch has a hierarchical structure
- Thus FullMatch can be represented as XML
- Semantics of FTSelections can be specified as transformation from input XML FullMatches to the output XML FullMatch
- Thus, semantics of FTSelections can be specified in XQuery itself!
- Full-text conditions and structural conditions represented in the same framework
  - Enables joint optimization and evaluation
Welcome to the GalaTex Website!
GalaTex is a conformance implementation of the W3C XQuery 1.0 and XPath 2.0 Full-Text standard proposal, an XML full text search language. GalaTex is built on top of Galax.
GalaTex's primary goal is to serve as a reference implementation for the XQuery 1.0 and XPath 2.0 Full-Text extension language. This site contains information about GalaTex, some documentation, on-line demos, W3C full-text uses cases and the semantic implementation of the full-text primitives. GalaTex and this Web site keep evolving so please come back again soon.

For any questions, please contact Emiran Curtmola, Sihem Amer-Yahia or Mary Fernández.

Latest News

* **August, 2005.** The GalaTex pre-release 0.5.1 source-code is available! If you are a GalaTex user or experimenter, please drop us a note.
* **August, 2004.** The first prototype for GalaTex demo is now available! This
declare variable $location { "docs/xml_3/" };
declare variable $xmlfile { doc(fn:concat($location, "_data3c.xml")) };

Choose a usecase query:

Q3: 12.2.3 Ordered Window Query

Submit Query

XQuery Full-Text expression:

(: Q3: 12.2.3 Find all books about "users feeling well-served" :) 

<results>
  for $book in $xmlfile/books/book
  where $book//content[ . ftcontains "user" && "feel"
              && "well-served" within window
                  at most 15 ordered ]

  return $book
</results>
GalaTex (http://www.galaxquery.org/galatex)

- Preprocessing & Inverted Lists Generation
- Full-Text Primitives (FTWord, FTWindow, FTTimes etc.)
- Galax XQuery Engine

XQFT Query

GalaTex Parser

Equivalent XQuery Query

Inverted lists

<xml>
<doc>
Text Text Text Text
</doc>
</xml>

evaluation

<doc>
Text Text Text
</doc>

GalaTex ((http://www.galaxquery.org/galatex))

.xml

(PI)

Positions API
Y a h o o C on f ide n tia l

Outline

• Motivation
• Challenges
• Languages
  – XQuery Full-Text
  – INEX
• Research overview
• Evaluate effectiveness of content-oriented XML retrieval systems
• Ongoing effort to define:
  – documents
  – queries (topics)
  – relevance assessments
  – metrics

http://inex.is.informatik.uni-duisburg.de/
Some 25 years ago, 26 if we are to be precise, a small group of computer scientists decided that their discipline not only had a past, it had a history. A history is a very different thing from a past. A past is a series of events; some good, bad, pleasing, embarrassing,.....

A history, however, looks at the deep trends of modern life and asks where they have been, where they are now, and where they are going. It is a discipline that looks to the future as much as it retells the story of the past. Those of us involved with the Annals believe that the stored program electronic computer helps us understand almost
Two types of topics

- Content-only (CO) topics
  - ignore document structure
  - simulates users, who do not have any knowledge of the document structure or who choose not to use such knowledge

- Content-and-structure (CAS) topics
  - contain conditions referring both to content and structure of the sought elements
  - simulate users who do have some knowledge of the structure of the searched collection
• Narrowed Extended XPath I
  – Designed for content-oriented XML search (i.e. “aboutness”)
  – query conditions on structure interpreted as hints to find content

• IEEE document collection growth
  – 12,107 to 659,388 documents
  – 8M to 30M elements
  – 494MB to 60GB (total size)

+ontologies -aumonyms

//article [ about (., ontologies) ]

//article [ about (., ontologies) ]//sec [ about (., ontologies case study) ]
I'm interested in knowing how ontologies are used to encode knowledge in real world scenarios. I'm writing a report on the use of ontologies. I'm particularly interested in knowing what sort or concepts and relations people use in their ontologies.

Case studies in the use of ontologies

I'm writing a report on the use of ontologies. I'm interested in knowing how ontologies are used to encode knowledge in real world scenarios. I'm particularly interested in knowing what sort or concepts and relations people use in their ontologies. I'm not interested in general ontology frameworks or technical details about tools for ontology creation or management. An example relevant result contains a description of the real world phenomena described by the ontology and also lists some of the concepts used and relations between concepts.
• Precision and recall are not enough:
  – relevance is a binary property (items are relevant or not)
  – relevance of one item independent from other items
  – user spends a constant time on each element
  – user looks at an ordered list and stops at some point

• The problem with retrieving elements:
  – specificity and exhaustiveness matter
  – size of retrieved elements varies => time spent varies
  – near-misses: some elements could be found by browsing
Metrics

• **inex-eval (precall)**
  – quantisation functions to capture specificity and exhaustivity
  – ignores possible overlap between elements

• **inex-eval-ng**
  – incorporate overlap and element size in precision and recall
  – consider only increment in text size of elements already seen

• **cumulative gain**
  – favors specificity
  – computed as the sum of relevance score up to that element
  – favors deeper nodes
Outline

• Motivation
• Challenges
• Languages
• Research Overview
Some papers

- Designed **TeXQuery** (Amer-Yahia, Botev, Shanmugasundaram, WWW 2004), and **XQuery Full-Text**, a full-text extension of XPath/XQuery (Amer-Yahia et al, [http://www.w3.org/TR/xquery-full-text/](http://www.w3.org/TR/xquery-full-text/), W3C Draft) and developed **GalaTex**, a conformant open-source implementation. (Curtmola, Amer-Yahia, Brown, Fernandez, XIME-P 2005)

- **Beyond DB**: Formalized a query semantics that consistently extends classical XPath semantics to account for XPath relevance ranking. **FleXPath** (Amer-Yahia, Lakshmanan, Pandit, SIGMOD 2004)

- **Beyond IR**: Developed a family of scoring methods for XML on both structure and content that are consistent with **tf*idf**. (Amer-Yahia, Koudas, Marian, Srivastava, Toman, VLDB 2005)

- Developed efficient algorithms for topK processing. **Whirlpool** (Marian, Amer-Yahia, Koudas, Srivastava, ICDE 2005)
//book [ ./info [ ./author ftcontains "Dickens" and ./title ftcontains "Great Expectations" ] and ./edition ]
Some XPath Relaxations

- Leaf node deletion
- Edge generalization
- Subtree promotion
- ...

Query

Data
//book [ ./info [ ./author ftcontains “Dickens” and ./title ftcontains “Great Expectations” ] and ./edition ]

pc($1,$2) and pc($2,$3) and pc($2,$4) and pc($1,$5) and ($1.tag = book) and ($2.tag = info) and ($3.tag = author) and ($4.tag = title) and ($5.tag = edition) and contains($3, “Dickens”) and contains($4,”Great Expectations”)
• Logical representation of query using predicates on structure and content.

• Compute *query closure* using inference rules below:
  – $pc(x, y)$ implies $ad(x, y)$
  – $ad(x, y), ad(y, z)$ implies $ad(x, z)$
  – $ad(x, y), contains(y, FTExp)$ implies $contains(x, FTExp)$
  – …

• Drop predicates.
• Compute *query core* (unique).
Example of XPath Relaxation

**query**

```
info
    ^
   /      |
author || Dickens
       v        v
   title || Great Expectations
          ^
         /      |
book    edition
```

**relaxed query**

```
info
    ^
   /      |
author || Dickens
       v        v
   title ||
            v
Great Expectations
          ^
         /      |
book    edition
```

- `pc($1,$2)` and `pc($2,$3)` and
- `pc($2,$4)` and `pc($1,$5)` and
- `($1.tag = article)` and `($2.tag = info)` and
- `($3.tag = author)` and
- `($4.tag = title)` and `($5.tag = edition)` and
- `contains($3, "Dickens")` and
- `contains($4,"Great Expectations")`

- `pc($1,$2)` and `ad($1,$3)` and
- `pc($2,$4)` and `ad($1,$5)` and
- `($1.tag = article)` and `($2.tag = info)` and
- `($3.tag = author)` and
- `($4.tag = title)` and `($5.tag = edition)` and
- `contains($3, "Dickens")` and
- `contains($4,"Great Expectations")`
Spanning XPath Relaxations

- Framework for defining new relaxations.
- Orthogonal to approximation on content.
- Answers to relaxed query contain answers to exact query.
- Score of answer to relaxed query should be no higher than score of answer to more exact query.
<table>
<thead>
<tr>
<th>Document Retrieval</th>
<th>XML Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Document</strong></td>
<td><strong>XML fragment</strong> (result is a subtree rooted at an element with a given tag and satisfying content and structure in query)</td>
</tr>
<tr>
<td><strong>Keyword</strong></td>
<td><strong>Path + Keyword</strong></td>
</tr>
<tr>
<td><code>&lt;idf&gt;</code> (<em>inverse document frequency</em>) is a function of the fraction of documents that contain the keyword</td>
<td><code>&lt;idf&gt;</code> is a function of the fraction of returned fragments that match the query tree pattern</td>
</tr>
<tr>
<td><code>&lt;tf&gt;</code> (<em>term frequency</em>) is a function of the number of occurrences of the keyword in the document</td>
<td><code>&lt;tf&gt;</code> is a function of the number of ways the query tree pattern matches the returned fragment</td>
</tr>
</tbody>
</table>
A Family of Scoring Methods

- **Binary scoring**
  - Low quality
  - Fast computation
- **Path scoring**
- **Twig scoring**
  - High quality
  - Expensive computation

Query

```
||  ||  ||  ||  ||
author  title  info  edition
Dickens  Great Expectations
```
What does XML mean anyway?

• EDS: Encyclopedia of Database Systems
  – definitions and illustrations of basic terminology, concepts, methods, and algorithms,
  – references to literature, and cross-references to other entries and journal articles.
  – Not a textbook

• April 15: Initial list of entries for XML

• Send to sihem@yahoo-inc.com