Advances in XML retrieval: The INEX Initiative

Norbert Fuhr
University of Duisburg-Essen
Germany
Outline of Talk

I. Models and methods for XML retrieval

III. Interactive retrieval

V. Views on XML retrieval
Part I: Models and methods for XML retrieval
Structured Document Retrieval

- Traditional IR is about finding relevant documents to a user’s information need, e.g. entire book.

- SDR allows users to retrieve document components that are more focussed to their information needs, e.g. a chapter, a page, several paragraphs of a book instead of an entire book.

- The structure of documents is exploited to identify which document components to retrieve.

  - Structure improves precision
XML retrieval allows users to retrieve document components that are more focussed, e.g. a subsection of a book instead of an entire book.
Queries

- Content-only (CO) queries
  - Standard IR queries but here we are retrieving document components
  - “London tube strikes”

- Structure-only queries
  - Usually not that useful from an IR perspective
  - “Paragraph containing a diagram next to a table”

- Content-and-structure (CAS) queries
  - Put constraints on which types of components are to be retrieved
    - E.g. “Sections of an article in the Times about congestion charges”
    - E.g. Articles that contain sections about congestion charges in London, and that contain a picture of Ken Livingstone, and return titles of these articles
  - Inner constraints (support elements), target elements
Content-oriented XML retrieval

Return document components of varying granularity (e.g. a book, a chapter, a section, a paragraph, a table, a figure, etc), relevant to the user’s information need both with regards to content and structure.

SEARCHING = QUERYING + BROWSING
Structured documents

Documents

Indexing

Document representation

Inverted file + structure index

Retrieval function

Retrieval results

Content + structure

Query

Formulation

Query representation

Matching content + structure

Presentation of related components
Challenge 1: term weights

No fixed retrieval unit + nested document components:

- how to obtain document and collection statistics (e.g. tf, idf)
- inner aggregation or outer aggregation?
**Nested document components:**

- which components contribute best to content of Article?
- how to estimate weights (e.g. size, number of children)?
Challenge 3: component weights

Different types of document components:
- which component is a good retrieval unit?
- is element size an issue?
- how to estimate component weights (frequency, user studies, size)?
Challenge 4: overlapping elements

Nested (overlapping) elements:
- Section 1 and article are both relevant to “XML retrieval”
- which one to return so that to reduce overlap?
- should the decision be based on user studies, size, types, etc?
Approaches …

divergence from randomness

Bayesian network

vector space model

machine learning

cognitive model

language model

Boolean model

belief model

logistic regression

probabilistic model

extending DB model

natural language processing

fusion

component statistics

term statistics

proximity search

tuning

relevance feedback

phrase

parameter estimation

smoothing

ontology

collection statistics

fused

natural language processing

divergence from randomness
Controlling Overlap

- Start with a component ranking, elements are re-ranked to control overlap.
- Retrieval status values (RSV) of those components containing or contained within higher ranking components are iteratively adjusted.

1. Select the highest ranking component.
2. Adjust the RSV of the other components.
3. Repeat steps 1 and 2 until the top $m$ components have been selected.

(SIGIR 2005)
XML retrieval

- Efficiency: Not just documents, but all its elements
- Models
  - Statistics to be adapted or redefined
  - Aggregation / combination
- User tasks
  - Focussed retrieval
  - No overlap
  - Do users really want elements

- Link to web retrieval / novelty retrieval
- Interface and visualisation
- Clustering, categorisation, summarisation
- Applications
  - Intranet, the Internet(?), digital libraries, publishing companies, semantic web, e-commerce
Evaluation of XML retrieval: INEX

- Evaluating the effectiveness of **content-oriented** XML retrieval approaches

- **Collaborative** effort ⇒ participants contribute to the development of the collection
  - queries
  - relevance assessments

- Similar methodology as for TREC, but adapted to XML retrieval
INEX test suites

- Corpora:
  - 16,819 articles in XML format from IEEE Computer Society (~750MB)
  - Wikipedia snapshot from April 2006 (660,000 articles, 4.6 GB)

- Queries:
  - 280 queries for IEEE-CS
  - 111 queries for Wikipedia

- Relevance judgments
  For the top 100 answers from each participant

- Collaborative effort:
  queries and relevance judgments from the 50-70 annual participants
Part II: Interactive retrieval
Investigate behaviour of searchers when interacting with XML components

1. Empirical foundation for evaluation metrics
2. What makes an effective search engine for interactive XML IR?

Content-only Topics
- topic type an additional source of context
  - 2004: Background topics / Comparison topics
  - 2005: Generalized task / complex task
- Each searcher worked on one topic from each type

Searchers
- “distributed” design, with searchers spread across participating sites
Baseline system

Search Result

1: (0.247) Scalable Feature Mining for Sequential Data  
Neal Lesh Mitsubishi Electric Research Lab Mohammed J. Zaki Rensselaer Polytechnic Institute Mitsunori Ogihara University of Rochester  
Result path: /article[1]/hdy[4]/sec[5]

2: (0.204) Probability and Agents  
Marco G. Valtorta University of South Carolina mgv@cse.sc.edu Michael N. Huhns University of South Carolina huhns@sc.edu  
Result path: /article[1]/hdy[4]/sec[3]

3: (0.176) Combining Image Compression and Classification Using Vector Quantization  
Karen L. Oehler Member IEEE Robert M. Gray Fellow IEEE  

4: (0.175) Text-Learning and Related Intelligent Agents: A Survey  
Dunja Mladenic J. Stefan Institute  

5: (0.175) Detecting Faces in Images: A Survey  
Ming-Hsuan Yang Member IEEE David J. Kriegman Senior Member IEEE Narendra Ahuja Fellow IEEE  
Result path: /article[1]/hdy[4]/sec[2]/ss1[9]/ss2[10]
Baseline system

Table of Contents

1 Introduction
2 Detecting faces in a single image
   2.1 Knowledge-Based
      Top-Down Methods
   2.2 Bottom-Up
      Feature-Based Methods
      2.2.1 Facial Features
      2.2.2 Texture
      2.2.3 Skin Color
      2.2.4 Multiple Features
   2.3 Template Matching
      2.3.1 Predefined Templates
      2.3.2 Deformable Templates
   2.4 Appearance-Based Methods
      2.4.1 Eigenfaces
      2.4.2 Distribution-Based Methods
      2.4.3 Neural Networks
      2.4.4 Support Vector Machines
   2.4.5 Sparse Network of Winnows
   2.4.6 NaiveBayes Classifier
   2.4.7 Hidden Markov Model
   2.4.8 Information-Theoretical Approach
   2.4.9 Inductive Learning
2.5 Discussion
3 Face image databases and performance evaluation

2.4.6 NaiveBayes Classifier

In contrast to the methods in [[107], [128], [154]] which model the global appearance of a face, Schneiderman and Kanade described a NaiveBayes classifier to estimate the joint probability of local appearance and position of face patterns (subregions of the face) at multiple resolutions [[140]]. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes are much more distinctive than the pattern found around the cheeks. There are two reasons for using a NaiveBayes classifier (i.e., no statistical dependency between the subregions). First, it provides better estimation of the conditional density functions of these subregions. Second, a NaiveBayes classifier provides a functional form of the posterior probability to capture the joint statistics of local appearance and position on the object. At each scale, a face image is decomposed into four rectangular subregions. These subregions are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns, and the statistics of each projected subregion are estimated from the projected samples to encode local appearance. Under this formulation, their method decides that a face is present when the likelihood ratio is larger than the ratio of prior probabilities. With an error rate of 0.36 percent on data set 1 in [[128]], the proposed Bayesian approach shows comparable performance to [[128]] and is able to detect some rotated and profile faces. Schneiderman and Kanade later extend this method with wavelet representations to detect profile faces and cars [[114]].

A related method using joint statistical models of local features was developed by Piccirelli et al. [[124]]. Local features are extracted by applying multiscale and multi-resolution filters to the input image. The distribution of the features vectors (i.e., filter responses) is estimated by clustering the data and then forming a mixture of Gaussians. After the model is learned and further refined, test images are classified by computing the likelihood of their feature vectors with respect to the model. Their experimental results on face and car detection show interesting and good results.
Some quantitative results

- How far down the ranked list?
  - 83 % from rank 1-10
  - 10 % from rank 11-20

- Query operators rarely used
- 80 % of queries consisted of 2, 3, or 4 words

- Accessing components
  - ~2/3 was from the ranked list
  - ~1/3 was from the document structure (ToC)

- 1st viewed component from the ranked list
  - 40% article level, 36% section level, 22% ss1 level, 4% ss2 level

- ~ 70 % only accessed 1 component per document
Qualitative results: User comments

- Document structure provides context 😊
- Overlapping result elements 😞
- Missing component summaries 😞
- Limited keyword highlighting 😞
- Missing distinction between visited and unvisited elements 😞
- Limited query language 😞
   - 1 Introduction
   - 10 Emerging technologies
     - 10.2 GPS ICs and cell phones
   - 5 G3 components
   - 6 Mobile site

   - 2 Mobility management
   - 3 Handoff
     - 3.2 INTERSYSTEM HANDOFF
   - 4 Roaming
     - 4.2 REGISTRATION
     - 4.3 CALL DELIVERY
**INTERSYSTEM HAND OFF**

Our description of the intersystem handoff follows IS-41[2] (GSM follows similar procedures), and we assume network-controlled handoff. Figure 3 illustrates the trunk (voice or data circuit) connection before and after the handoff. A communicating mobile user moves out of the base station served by MSC1 and enters the area covered by MSC2. The handoff follows these steps:

- MSC1 requests MSC2 to perform handoff measurement. MSC2 then selects a candidate base station, BS2, for handoff. That is, MSC2 finds a base station that covers the mobile phone and has a free radio channel to cover the call. MSC2 returns the signal-quality parameter values and other information to MSC1.
- MSC1 checks if the mobile phone has made too many handoffs or if intersystem trunks are not available. If so, MSC1 exits the procedure. Otherwise, MSC1 asks MSC2 to set up a voice channel. Suppose that a voice channel is available in BS2. MSC2 asks MSC1 to start the radio link transfer.
- MSC1 sends the mobile phone a handoff order. The mobile phone tries to synchronize to BS2. After the mobile phone connects to BS2, MSC2 informs MSC1 that the handoff is successful. MSC1 then connects the call path (trunk) to MSC2 and completes the handoff.

![Diagram of an intersystem handoff](image)

**Figure 3: Before (a) and after (b) an intersystem handoff**
User comments

- Context of retrieved elements in resultlist 😊
- No overlapping elements in resultlist 😊
- Table of contents and query term highlighting 😊
- Display of related terms for query 😞
- Distinction between visited and unvisited elements 😊
- Retrieval quality 😞
Part III: Views on XML Retrieval
XML structure: 1. Nested Structure

- XML document as hierarchical structure
- Retrieval of elements (subtrees)
- Typical query language does not allow for specification of structural constraints
- Relevance-oriented selection of answer elements: return the most specific relevant elements
XML structure: 2. Named Fields

- Reference to elements through field names only
- Context of elements is ignored (e.g. author of article vs. author of referenced paper)
- Post-Coordination may lead to false hits (e.g. author name – author affiliation)
- [Kamps et al. (TOIS 4/06)]: XML retrieval quality does not suffer from restriction to named fields

Example: Dublin Core

```xml
<oai_dc:dc
xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>Generic Algebras
  </dc:title>
  ...<dc:title>
  <dc:creator>A. Smith (ESI), B. Miller (CMU)</dc:creator>
  <dc:subject>Orthogonal group, Symplectic group</dc:subject>
  <dc:date>2001-02-27</dc:date>
  <dc:format>application/postscript</dc:format>
  <dc:source>ESI preprints</dc:source>
  <dc:language>en</dc:language>
</oai_dc:dc>```
XML structure: 3. XPath

/document/chapter[about(./heading, XML) AND about(./section/*, syntax)]
Full expressiveness for navigation through document tree (+links)
- Parent/child, ancestor/descendant
- Following/preceding, following-sibling, preceding-sibling
- Attribute, namespace

Selection of arbitrary elements

Too complex for users?
Higher expressiveness, especially for database-like applications:
  — Joins
  — Aggregations
  — Constructors for restructuring results

Example: List each publisher and the average price of its books.

```xquery
FOR $p IN distinct(document("bib.xml")//publisher)
LET $a := avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
  <publisher>
    <name> {$p/text()} </name>
    <avgprice> {$a} </avgprice>
  </publisher>
```

How many papers on digital libraries by Ed Fox?
XML Content Typing

Content Typing

Object Types

Data Types

Text only

Nested structure

Named fields

XPath

XQuery

Structure
Example query:

//chapter[about(.,
 XML query language]
XML content typing: 2. Data Types

- Data type: domain + (vague) predicates
  - Language (multilingual documents) / (language-specific stemming)
  - Person names / “his name sounds like Jones”
  - Dates / “about a month ago”
  - Amounts / “orders exceeding 1 Mio $”
  - Technical measurements / “at room temperature”
  - Chemical formulas

- Close relationship to XML Schema, but
  - XMLS supports syntactic type checking only
  - No support for vague predicates
XML content typing: 3. Object Types

- Object types: Persons, Locations, Companies, ...

Pablo Picasso (October 25, 1881 - April 8, 1973) was a Spanish painter and sculptor. In Paris, Picasso entertained a distinguished coterie of friends in the Montmartre and Montparnasse quarters, including André Breton, Guillaume Apollinaire, and writer Gertrude Stein.

To which other artists did Picasso have close relationships?

Did he ever visit the USA?

- Named entity recognition methods allow for automatic markup of object types
- Object types support increased precision
Tag semantics?

Object type hierarchies:
- Person
  - Scientist
    - Physicist
  - Artist
    - Poet
    - Actor
    - Singer

Role hierarchies:
- Creator
  - Author
  - Editor
DAML+OIL for semantic XML IR?
DAML+OIL for semantic XML IR? (cont’d)

DAML+OIL...

- ... may allow for semantic retrieval from XML collections
- ... may be useful for retrieval from federated collections (using different DTDs)
- ... currently supports XML for literals only
- ... does not provide appropriate query language
- ... does not support uncertain inference
Conclusion and future work

- Research issues in XML retrieval
  - Effective retrieval of XML documents
  - What and how to evaluate

- Interactive XML retrieval:
  - Empirical foundation for the need for element retrieval (instead of full documents)

- Views on XML:
  - Large variety of possible applications
  - But lack of appropriate test collections

- XML and Semantic Web technologies:
  - Potentially useful, especially in limited domains (but open research issues)
Thank you for your attention!

More info about INEX:
http://inex.is.inf.uni-due.de

Questions?