XML Information Retrieval and INEX

Norbert Fuhr
University of Duisburg-Essen
Outline of Talk

I. Models and methods for XML retrieval

III. Interactive retrieval

V. Evaluation of 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
  - Exploit visual memory
XML retrieval allows users to retrieve document components that are more focussed, e.g. a subsection of a book instead of an entire book.

SEARCHING = QUERYING + BROWSING
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**
Conceptual model

Structured documents

Documents

Indexing

Document representation

Inverted file + structure index

Content + structure

Query

Formulation

Query representation

Matching content + structure

Retrieval function

Retrieval results

Query representation

Retrieval function

Retrieval results

Presentation of related components

Relevance feedback
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?
Challenge 2: augmentation weights

Nested document components:
- which components contribute best to content of Article?
- how to estimate weights (e.g. size, number of children)?
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?
Vector space model

article index → RSV → normalised RSV
abstract index → RSV → normalised RSV
section index → RSV → normalised RSV
sub-section index → RSV → normalised RSV
paragraph index → RSV → normalised RSV

merge

tf and idf as for fixed and non-nested retrieval units

(IBM Haifa, INEX 2003)
Language model

element language model
collection language model
smoothing parameter $\lambda$

element score

high value of $\lambda$ leads to increase in size of retrieved elements

element size
element score
article score

rank element

query expansion with blind feedback
ignore elements with $\leq 20$ terms

results with $\lambda = 0.9$, 0.5 and 0.2 similar

(University of Amsterdam, INEX 2003)
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
Part II: Interactive retrieval
Interactive Track

- **Investigate behaviour of searchers when interacting with XML components**
  - Empirical foundation for evaluation metrics
  - 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
I've been asked to make my Fortran compiler compatible with Fortran 90 so I'm interested in the features Fortran 90 added to the Fortran standard before it. I'd like to know about compilers (they would have been new when they were introduced), especially compilers whose source code might be available. Discussion of people's experience with these features when they were new to them is also relevant. An element will be judged as relevant if it discusses features that Fortran 90 added to Fortran.
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]/bdy[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]/bdy[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]/bdy[4]/sec[2]/ss1[9]/ss2[10]
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 0.9 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 [141].

A related method using joint statistical models of local features was developed by Pockert 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 😞

   Ayman Kayssi; Ralph Achkar; Mark Azar; Joseph Samaha

   - 1 Introduction
   - 10 Emerging technologies
     - 10.2 GPS ICs and cell phones
   - 5 G3 components
   - 6 Mobile site


   Yi-Bing Lin

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

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 channel) connection before and after the handoff. A communicating mobile user moves out of the base station served by MSC<sub>1</sub> and enters the area covered by MSC<sub>2</sub>. The handoff follows these steps:

- MSC<sub>1</sub> requests MSC<sub>2</sub> to perform handoff measurement. MSC<sub>2</sub> then selects a candidate base station, BS<sub>2</sub>, for handoff. That is, MSC<sub>2</sub> finds a base station that covers the mobile phone and has a free radio channel to cover the call. MSC<sub>2</sub> returns the signal-quality parameter values and other information to MSC<sub>1</sub>.
- MSC<sub>1</sub> checks if the mobile phone has made too many handoffs or if intersystem trunks are not available. If so, MSC<sub>1</sub> exits the procedure. Otherwise, MSC<sub>1</sub> asks MSC<sub>2</sub> to set up a voice channel. Suppose that a voice channel is available in BS<sub>2</sub>. MSC<sub>2</sub> asks MSC<sub>1</sub> to start the radio link transfer.
- MSC<sub>1</sub> sends the mobile phone a handoff order. The mobile phone tries to synchronize to BS<sub>2</sub>. After the mobile phone connects to BS<sub>2</sub>, MSC<sub>2</sub> informs MSC<sub>1</sub> that the handoff is successful. MSC<sub>1</sub> then connects the call path (trunk) to MSC<sub>2</sub> and completes the handoff.

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: Evaluation of XML retrieval
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

- Documents ~750MB: 16,819 articles in XML format from IEEE Computer Society; 8 million elements!

- INEX 2002
  - 60 topics, inex_eval metric

- INEX 2003
  - 66 topics, use subset of XPath, inex_eval and inex_eval_ng metrics

- INEX 2004
  - 75 topics, subset of 2003 XPath subset (NEXI)
    - Official metric: inex_eval with averaged different “assumed user behaviours”
    - Others: inex_eval_ng, XCG, t2i, ERR, PRUM, …

- INEX 2005
  - 87 topics, NEXI
    - Official metric: XCG
<title>Open standards for digital video in distance learning</title>

<description>Open technologies behind media streaming in distance learning projects</description>

<narrative>I am looking for articles/components discussing methodologies of digital video production and distribution that respect free access to media content through internet or via CD-ROMs or DVDs in connection to the learning process. Discussions of open versus proprietary standards of storing and sending digital video will be appreciated. </narrative>
InEX Topics: Content-and-structure

.setTitle createState('@article[about(.,'formal methods verify correctness aviation systems')]/sec/* [about(.,'case study application model checking theorem proving')]</title>

decription Find documents discussing formal methods to verify correctness of aviation systems. From those articles extract parts discussing a case study of using model checking or theorem proving for the verification. </description>

.narrative To be considered relevant a document must be about using formal methods to verify correctness of aviation systems, such as flight traffic control systems, airplane- or helicopter- parts. From those documents a section-part must be returned (I do not want the whole section, I want something smaller). That part should be about a case study of applying a model checker or a theorem prover to the verification. </narrative>
Ad hoc retrieval: Tasks

- **Content-only (CO):** aim is to decrease user effort by pointing the user to the most specific relevant elements (2002 - )

- **Strict content-and-structure (SCAS):** retrieve relevant elements that exactly match the structure specified in the query (2002, 2003)

- **Vague content-and-structure (VCAS):**
  - retrieve relevant elements that may not be the same as the target elements, but are structurally similar (2003)
  - retrieve relevant elements even if do not exactly meet the structural conditions; treat structure specification as hints as to where to look (2004)

- **[V/S][V/S]CAS:** distinction between interpretation of target element and of support element: VVCAS, SVCAS, VSCAS, SSCAS (2005)
Relevance in information retrieval

- A document is **relevant** if it “has significant and demonstrable bearing on the matter at hand”.

- Common assumptions in information retrieval laboratory experimentation:
  - Objectivity
  - Topicality
  - Binary nature
  - Independence

(Borlund, JASIST 2003)
Relevance in XML retrieval

- A document is **relevant** if it “has significant and demonstrable bearing on the matter at hand”.

- Common assumptions in laboratory experimentation:
  - Objectivity
  - Topicality
  - Binary nature
  - Independence
Relevance in XML retrieval: INEX

- Topicality not enough
- Binary nature not enough
- Independence is wrong

Relevance = (0,0) (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)

exhaustivity = how much the section discusses the query: 0, 1, 2, 3

specificity = how focused the section is on the query: 0, 1, 2, 3

If a subsection is relevant so must be its enclosing section, ...

(based on Chiaramella et al, FERMI fetch and browse model 1996)
Relevance - to recap

- find smallest component (--> specificity) that is highly relevant (--> exhaustivity)

- **specificity**: extent to which a document component is focused on the information need, while being an informative unit.

- **exhaustivity**: extent to which the information contained in a document component satisfies the information need.
Overlapping (nested) result elements retrieval runs
Overlapping (nested) reference elements in recall-base
Relevance propagates up!

- ~26,000 relevant elements on ~14,000 relevant paths
- Propagated assessments: ~45%
- Increase in size of recall-base: ~182%
Retrieve the best XML elements according to content and structure criteria:

- Most exhaustive and most specific = (3,3)

- Near misses = (3,3) + (2,3) (1,3)  <-- specific
- Near misses = (3,3) + (3,2) (3,1)  <-- exhaustive
- Near misses = (3,3) + (2,3) (1,3) (3,2) (3,1) (1,2) …

- Focussed retrieval = no overlapping elements
Two four-graded dimensions of relevance

- How to differentiate between (1,3) and (3,3), …?

- **Several “user models”**
  - **Expert and impatient**: only reward retrieval of highly exhaustive and specific elements (3,3)
  - Expert and patient: only reward retrieval of highly specific elements (3,3), (2,3) (1,3)
  - …
  - **Naïve and has lots of time**: reward - to a different extent - the retrieval of any relevant elements; i.e. everything apart (0,0)

- Use a **quantisation function** for each “user model”
Examples of quantisation functions

Expert and impatient

\[ f_{\text{strict}}(\text{exh}, \text{spec}) = \begin{cases} 1 & \text{if } \text{exh} = 3 \text{ and } \text{spec} = 3 \\ 0 & \text{otherwise} \end{cases} \]

Naïve and has a lot of time

\[ f_{\text{generalised}}(\text{exh}, \text{spec}) = \begin{cases} 1.00 & \text{if } (\text{exh}, \text{spec}) = (3,3) \\ 0.75 & \text{if } (\text{exh}, \text{spec}) \in \{(2,3), (3,2), (3,1)\} \\ 0.50 & \text{if } (\text{exh}, \text{spec}) \in \{(1,3), (2,2), (2,1)\} \\ 0.25 & \text{if } (\text{exh}, \text{spec}) \in \{(1,1), (1,2)\} \\ 0.00 & \text{if } (\text{exh}, \text{spec}) = (0,0) \end{cases} \]
Based on **precall** (Raghavan et al, TOIS 1989) itself based on expected search length (Cooper, JASIS 1968)

Use several quantisation functions

In its form cannot consider
- overlap in retrieval runs
- overlap in recall-base

Not easy way to extend to consider BOTH
## Official INEX 2004 Results for CO topics

<table>
<thead>
<tr>
<th>Rank</th>
<th>Systems (runs)</th>
<th>Avg Prec</th>
<th>% Overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>IBM Haifa Research Lab(CO-0.5-LAREFIENMENT)</td>
<td>0.1437</td>
<td>80.89</td>
</tr>
<tr>
<td>2.</td>
<td>IBM Haifa Research Lab(CO-0.5)</td>
<td>0.1340</td>
<td>81.46</td>
</tr>
<tr>
<td>3.</td>
<td>University of Waterloo(Waterloo-Baseline)</td>
<td>0.1267</td>
<td>76.32</td>
</tr>
<tr>
<td>4.</td>
<td>University of Amsterdam(UAms-CO-T-FBack)</td>
<td>0.1174</td>
<td>81.85</td>
</tr>
<tr>
<td>5.</td>
<td>University of Waterloo(Waterloo-Expanded)</td>
<td>0.1173</td>
<td>75.62</td>
</tr>
<tr>
<td>6.</td>
<td>Queensland University of Technology(CO_PS_Stop50K)</td>
<td>0.1073</td>
<td>75.89</td>
</tr>
<tr>
<td>7.</td>
<td>Queensland University of Technology(CO_PS_099_049)</td>
<td>0.1072</td>
<td>76.81</td>
</tr>
<tr>
<td>8.</td>
<td>IBM Haifa Research Lab(CO-0.5-Clustering)</td>
<td>0.1043</td>
<td>81.10</td>
</tr>
<tr>
<td>9.</td>
<td>University of Amsterdam(UAms-CO-T)</td>
<td>0.1030</td>
<td>71.96</td>
</tr>
<tr>
<td>10.</td>
<td>LIP6(simple)</td>
<td>0.0921</td>
<td>64.29</td>
</tr>
</tbody>
</table>
XCG: XML cumulated gain

- Based on cumulated gain measure for IR (Kekäläinen and Järvelin, TOIS 2002)

- Accumulate gain obtained by retrieving elements up to a given rank; thus not based on precision and recall

- Require the construction of an ideal recall-base and associated ideal run, with which retrieval runs are compared

- Consider overlap in both retrieval runs and recall-base

(SIGIR 2004, INEX 2004)
Cumulated Gain

- Gain vector (G) from ranked document list
- Ideal gain vector (I) from documents in recall-base
- Cumulated gain (CG)

\[ CG[i] = \sum_{j=1}^{i} G[j] \]

Col = \langle d_4, d_5, d_2, d_3, d_1 \rangle

G = \langle 3, 0, 1, 3, 2 \rangle

I = \langle 3, 3, 2, 1, 0 \rangle

CG_G = \langle 3, 3, 4, 7, 9 \rangle

CG_I = \langle 3, 6, 8, 9, 9 \rangle
Ideal Recall-base and Run

- **Ideal recall-base** - which non-overlapping elements do we keep?
  - Derived based on retrieval tasks

- **Ideal run** - how do we order the above elements?
  - Ordering elements of the ideal recall-base by relevance value $r_v$
XCG - Ideal run vs. actual run

Recall-base:

- Ideal gain vector:
  \[ I[i] = rv(c_i) \]
  \( (rv(c_i) \text{ from ideal recall-base}) \)

- Actual gain vector:
  \[ G[i] = rv(c_i) \]
  \( (rv(c_i) \text{ from full recall-base}) \)

Ranked result list:

- Recall-base:
  - (3,3)
  - (3,2)

- Ideal result list:
  - (3,1)
  - (3,2)
  - (1,2)
  - (1,3)

- Actual result list:
  - (3,3)
  - (3,2)
<table>
<thead>
<tr>
<th>Rank</th>
<th>Team Description</th>
<th>MAnCG</th>
<th>ovrlp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>U. of Tampere (UTampere CO average)</td>
<td>0.3725</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>U. of Tampere (UTampere CO fuzzy)</td>
<td>0.3699</td>
<td>0</td>
</tr>
<tr>
<td>3.</td>
<td>U. of Amsterdam (UAmS-CO-T-FBack-NoOverl)</td>
<td>0.3521</td>
<td>0</td>
</tr>
<tr>
<td>4.</td>
<td>Oslo U. College (4-par-co)</td>
<td>0.3418</td>
<td>0.07</td>
</tr>
<tr>
<td>5.</td>
<td>U. of Tampere (UTampere CO overlap)</td>
<td>0.3328</td>
<td>39.58</td>
</tr>
<tr>
<td>6.</td>
<td>LIP6 (bn-m1-eqt-porder-eul-o.df.t-parameters-00700)</td>
<td>0.3247</td>
<td>74.31</td>
</tr>
<tr>
<td>7.</td>
<td>U. California, Berkeley (Berkeley CO FUS T CMBZ FDBK)</td>
<td>0.3182</td>
<td>50.22</td>
</tr>
<tr>
<td>8.</td>
<td>U. of Waterloo (Waterloo-Filtered)</td>
<td>0.3181</td>
<td>13.35</td>
</tr>
<tr>
<td>9.</td>
<td>LIP6 (bn-m2-eqt-porder-o.df.t-parameters-00195)</td>
<td>0.3098</td>
<td>64.2</td>
</tr>
<tr>
<td>10.</td>
<td>LTI, CMU (Lemur CO KStem Mix02 Shrink01)</td>
<td>0.2953</td>
<td>73.02</td>
</tr>
</tbody>
</table>

[?] rank by inex_eval
Conclusion and future work

- Difficult research issues in XML retrieval are not ‘just’ about the effective retrieval of XML documents, but also about what and how to evaluate!

INEX 2006:
- New Collection: Wikipedia (with addtl. markup)
- Tasks:
  - Adhoc (Content / Content and structure)
  - Interactive
  - Multimedia
  - Relevance feedback
  - Document mining
  - XML entity ranking (new)
  - Natural language processing
  - Heterogeneous collection