# Beyond XML Retrieval

**Mounia Lalmas**
Queen Mary, University of London


1

## XML

- XML: eXtensible Markup Language
  - XML is able to represent a mix of structured and text (unstructured) information
- XML applications: *data interchange, digital libraries, content management, complex documentation, etc.*
- XML repositories: *Library of Congress collection, SIGMOD DBLP, IEEE INEX collection, LexisNexis, …*

(http://www.w3.org/XML/)

2

## DB and IR view

- Data-centric view
  - XML as exchange format for structured data
  - Used for messaging between enterprise applications
  - Mainly a recasting of relational data
- **Document-centric view**
  - **XML as format for representing the logical structure of documents**
  - **Rich in text**
- Now increasingly both views (DB+IR)

3

## Document-centric XML retrieval

- Documents marked up as XML
  - E.g., assembly manuals, journal issues …
- Queries are user information needs
  - E.g., give me the section (element) of the document that tells me how to change a brake light
- Different from well-structured XML queries where one tightly specifies what he/she is looking for.

• Structure improves precision
• Exploit visual memory

4

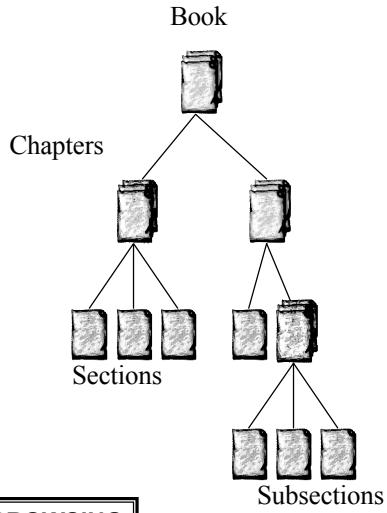
# Queries

- **Content-only (CO) queries**
  - Standard IR queries, but here we are retrieving document components
    - "Wine tasting in San Marino"
- **Content-and-structure (CAS) queries**
  - Put constraints on which types of components are to be retrieved
    - E.g. "Sections of an article about wine tasting in San Marino"
    - E.g. Articles that contain sections about wine tasting in San Marino, and that contain a picture of fortress, *and* return titles of these articles"

5

# XML retrieval vs. "flat" document retrieval

- No predefined unit of retrieval
- Dependency of retrieval units
- Aims of XML retrieval:
  - Not only to find relevant elements
  - But those at the appropriate level of granularity
  - **Focused retrieval**




**SEARCHING = QUERYING + BROWSING**

6

## 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
  - methodology
- Similar methodology as for TREC, but adapted to XML retrieval


http://inex.is.informatik.uni-duisburg.de/

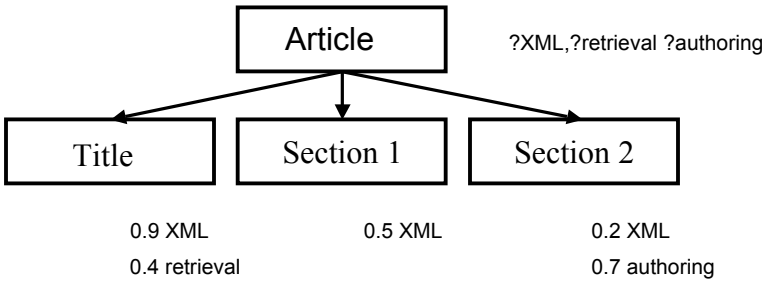
7

## Outline of the rest of the talk

- Challenges in XML retrieval
- Some approaches
  - only some, and not covering all the challenges
  - for all, see up-coming book (still being written)
- Beyond XML retrieval
  - beyond a-la-INEX XML retrieval

8

# Challenge 1: Term statistics

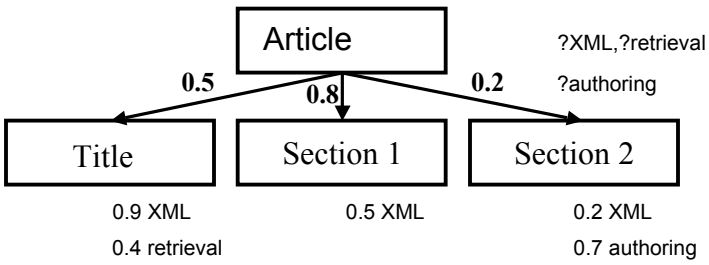



**No fixed retrieval unit + nested document components:**

- how to obtain element and collection statistics (e.g. tf, idf)?
- inner or outer calculation?

9

# Challenge 2: Relationship statistics

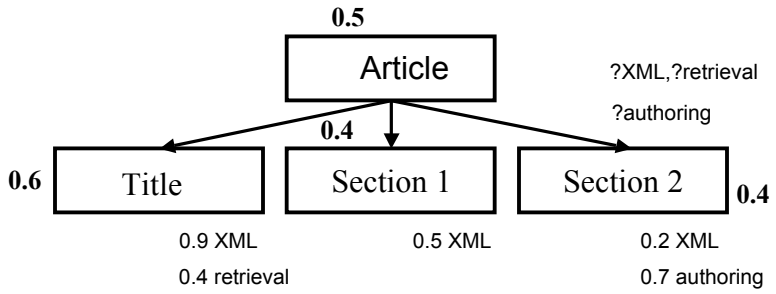



**Relationship between elements:**

- which sub-element(s) contribute best to content of its parent element and vice versa?
- how to estimate (or learn) relationship statistics (e.g. size, number of children, depth, distance)?

10

# Challenge 3: Structure statistics

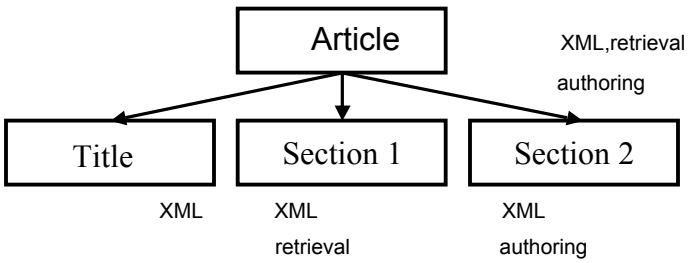



**Different types of elements:**

- which element is a good retrieval unit?
- is element size an issue?
- how to estimate (or learn) structure statistics (frequency, user studies, size, depth)?

11

# 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?

12

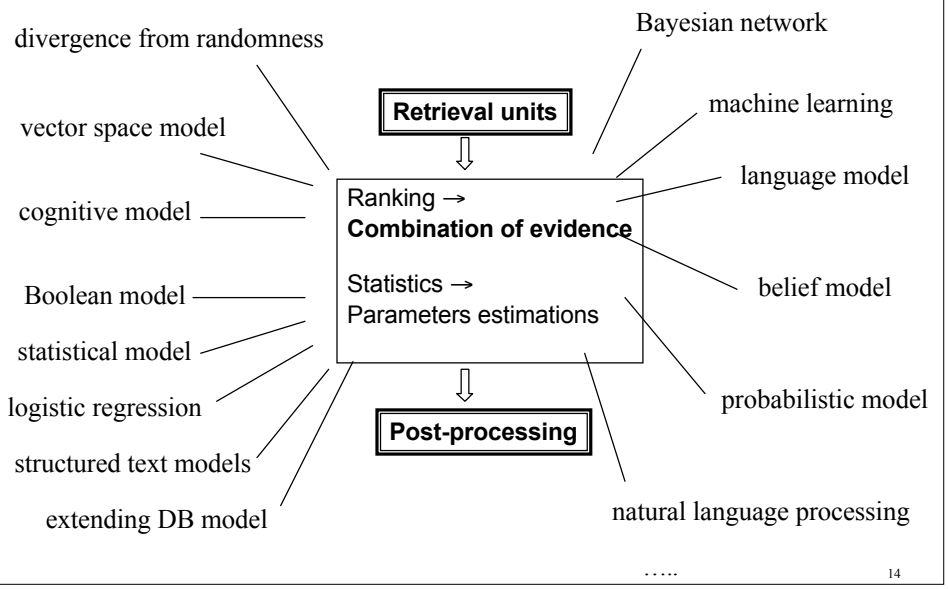
## Challenge 5: Expressing and interpreting structural constraints

- Ideally:
  - There is one DTD/schema
  - User understands DTD/schema
- In practice: rare
  - Many DTDs/schemas
  - DTDs/Schemas not known in advance
  - DTDs/Schemas change
  - Users do not understand DTDs/schemas
  - **How to expect "users" to express structural constraints?**

- Need to identify "similar/synonym" elements/tags
- Strict or vague interpretation of the structure
- Relevance feedback/blind feedback?

13

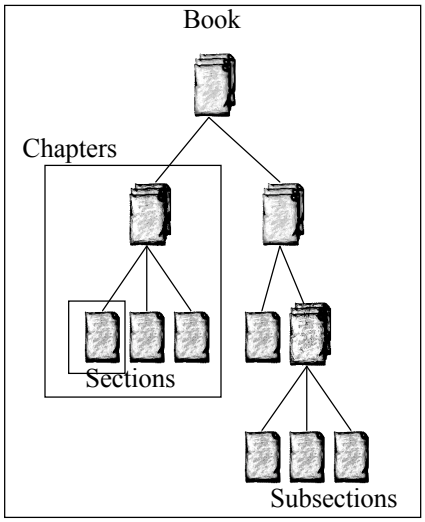
## Retrieval models …




14

# Retrieval units: What to Index?

- XML documents are trees
  - hierarchical structure
  - of nested elements (sub-trees)
- What should we put in the index?
  - there is no fixed unit of retrieval




15

# Retrieval units: XML sub-trees

**Assume a document like**

```
<article>
  <title>XXX</title>
    <abstract>YYY</abstract>
  <body>
    <sec>ZZZ</sec>
    <sec>ZZZ</sec>
  </body>
</article>
```

**Index separately**

- <article>XXX YYY ZZZ ZZZ </article>
- <title>XXX</title>
- <abstract>YYY</abstract>
- <body>ZZZ ZZZ</body>
- <sec>ZZZ</sec>
- <sec>ZZZ</sec>

16

## Retrieval units: XML sub-trees

- Indexing sub-trees is closest to traditional IR
  - each XML elements is bag of words of itself and its descendants
  - and can be scored as ordinary plain text document
- Advantage: well-understood problem
- Negative:
  - redundancy in index
  - terms statistics
  - Led to the notion of fixed indexing nodes
  - Problem: how to select them?
    - manually, frequency, relevance data

17

## Retrieval units: Disjoint elements

**Assume a document like**

```
<article>
  <title>XXX</title>
    <abstract>YYY</abstract>
  <body>
    <sec>ZZZ</sec>
    <sec>ZZZ</sec>
  </body>
</article>
```

**Index separately**

- <title>XXX</title>
- <abstract>YYY</abstract>
- <sec>ZZZ</sec>
- <sec>ZZZ</sec>

Note that <body> and <article> have not been indexed

18

## Retrieval units 2: Disjoint elements

- Main advantage and main problem
  - (most) article text is not indexed under /article
  - avoids redundancy in the index
- But how to score higher level (non-leaf) elements?
  - Propagation/Augmentation approach
  - Element specific language models

19

## Retrieval units: Distributed

- Index separately particular types of elements
- E.g., create separate indexes for
  - articles
  - abstracts
  - sections
  - subsections
  - subsubsections
  - paragraphs …

  } structure statistics
- Each index provides statistics tailored to particular types of elements
  - language statistics may deviate significantly
  - queries issued to all indexes
  - results of each index are combined (after score normalization)

20

# Retrieval units: Distributed

- Only part of the structure is used
  - Element size
  - Relevance assessment
  - Others
- Main advantages compared to disjoint element strategy:
  - avoids score propagation which is expensive at run-time
  - index redundancy is basically pre-computing propagation
  - avoid non-trivial parameters to train needed for propragation
- Indexing methods and retrieval models are “standard” IR
  - although issue of merging - normalization

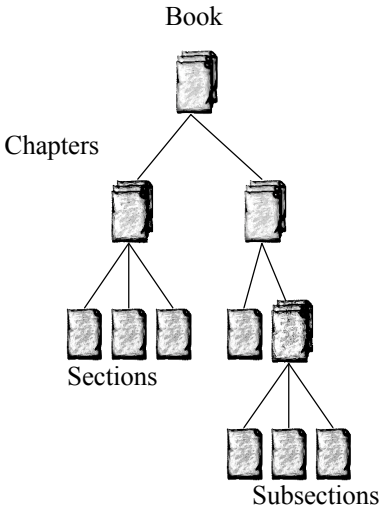
21

# Ranking: What and how to combine?

- XML documents are trees

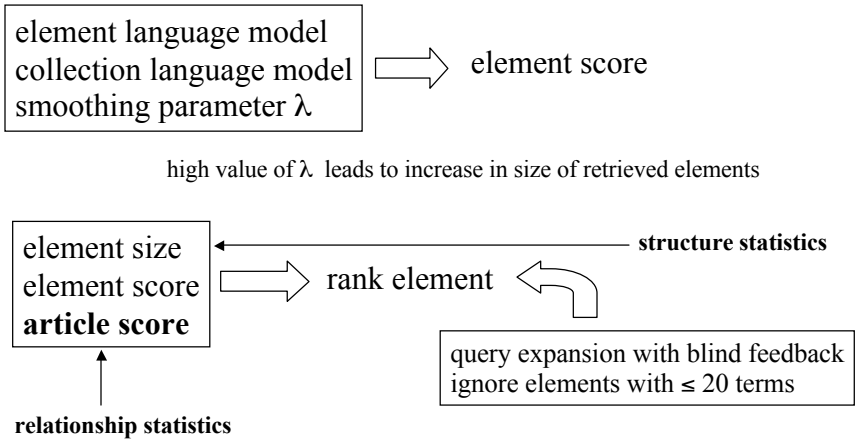
  elements are not independent

- What should we use to estimate the relevance of an element?




22

## Combination: Language model




(Sigurbjörnsson etal, INEX 2003, INEX 2004)

23

## Combination: Machine learning

- Use of standard machine learning to train a function that combines
  - Parameter for a given element type
  - Parameter ∗ score(element)
  - Parameter ∗ score(parent(element))
  - Parameter ∗ score (document)

structure statistics

relationship statistics

- Training done on relevance data (previous years)
- Scoring done using OKAPI

(Vittaut & Gallinari, ECIR 2006)

24

## What have we learned?

- Issue of how to start - what to index
- XML retrieval can be viewed as a **combination of evidence** problem
- No “clear winner” in terms of retrieval models
  - We still miss the benchmark/baseline approach
  - Lots of heuristics
- BUT WHAT SEEM TO WORK WELL ACROSS ALL MODELS:
  - Element
  - Document
  - Size

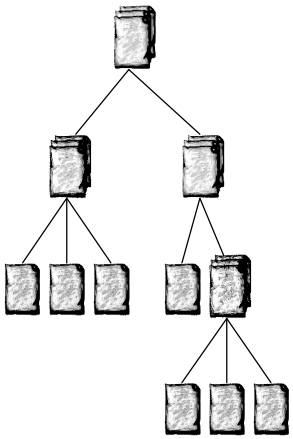
- ***Thorough investigation for all ranking models, all indexing approaches, and all evidence needed***

25

## Beyond XML retrieval

- **Focused retrieval**
- **Aggregated results**

- **Structural context summarization**
- **Beyond the logical structure**



26

## Beyond XML retrieval: Focused retrieval

- Best performance obtained using evidence from element, document, and element size, and this whatever the model.
  - How can we apply this to other so-called “focused” retrieval problem?
  - What other evidence, e.g. semantic tags, should be used?
  - What combination formalism should be used?

27

## Beyond XML retrieval: Aggregated results

- We know how to retrieve “snippets”.
- We know how to return “snippets” within a document (e.g. heatmap).
- How to combine/mix snippets from across documents to return **meaningful** aggregated results?
  - “Virtual” documents (from Chiaramella)
  - Refer to Vanessa Murdock presentation

28

# Heatmap

- Document ranking, and in each document, element ranking

**OntoSeek: Content-Based Access to the Web**
**Nicola Guarino, Claudio Masolo, Guido Vetere**

article[1]
fm[1]
abs[1]
p[1]
... and precision of content-based **retrieval**. Our OntoSeek system adopts ... large ontology based on **WordNet** for content matching.
bdy[1]
sec[3] (THE ROLE OF LINGUISTIC ONTOLOGIES)
ss1[1] (Some advantages)
p[4]
The **retrieval** quality improves considerably if ... linguistic ontology such as **WordNet**. For example, let's add **WordNet** to a simple matching ...
p[7]
... linguistic ontologies such as **WordNet** and structured representation formalisms can help an **information-retrieval** system to
sec[4] (ONTOSEEK)
... of a project on **retrieval** and reuse of object-oriented ... system designed for content-based **information retrieval** from online yellow pages ... mostly resulting from merging **WordNet**'s thesaurus into the Penman ... broad ontology endowed with **WordNet**'s powerful lexical interface, which ...

29

# Beyond XML retrieval: Structural context summarization

- Users require document context when viewing an elements result
- We know how to summarize the structure (ToC) of a document (depth, relevance, etc)
- How can we summarize the structure of the search results, to provide context for the whole search.
  - Not just clusters

30

# XML retrieval systems display:

dbdk_training in **Baseline System**

Search

HyREX

query was: text classification naive bayes
Results **1** - **10** of **100**.
Result pages: **1** 2 3 4 5 6 7 8 9 10 next

## 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*
**Result path:** `/article[1]/bdy[4]/sec[4]/ss1[2]/ss2[4]`

**4:** (0.175) **Text-Learning and Related Intelligent Agents: A Survey**
*Dunja Mladenic J. Stefan Institute*
**Result path:** `/article[1]/bm[5]/app[4]/sec[5]`

**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]`

31

# Providing context for the element

**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 Naive Bayes 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

Close Document

**To which extent this piece of information covers your problem or topic of interest:**
Unspecified submit

**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 93.0 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 Rickert et al. [[124] ]. Local features are extracted by applying multiscale and multiresolution 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.

**To which extent this piece of information covers your problem or topic of interest:**
Unspecified submit

- Unspecified
- Very useful & Very specific
- Very useful & Fairly specific
- Very useful & Marginally specific
- Fairly useful & Very specific
- Fairly useful & Fairly specific
- Fairly useful & Marginally specific
- Marginally useful & Very specific
- Marginally useful & Fairly specific
- Marginally useful & Marginally specific
- Contains no relevant information

32

## Beyond XML retrieval: Beyond the logical structure

- We know how to exploit the tags representing the logical structure to provide focused retrieval.
- What about other tags, e.g. semantic tags, formatting tags, template tags, etc?

33

## Beyond XML Retrieval

**Thank you**

34