
XML Retrieval

DB/IR in Theory

Web in Practice

Sihem Amer-Yahia

Yahoo! Research

Mariano Consens

University of Toronto

In collaboration with:

Ricardo Baeza-Yates

Yahoo! Research

Mounia Lalmas

Queen Mary, Univ. of London

VLDB 2007, Vienna, 26/09/07

Preliminaries

- DB focused on languages, expressiveness and efficient evaluation
- IR focused on scoring and relevance metrics
 - In practice, a limited set of operations and simple ranking go a long way
- Theory is scary (think XQuery)
- Practice is inspiring but looks ad-hoc

Notion of Relevance

- Data retrieval:
 - Syntax expresses semantics
- Information retrieval:
 - Ambiguous semantics
 - Relevance depends on user and context
 - There is no "perfect" retrieval system
- User assessments to evaluate system effectiveness

Overview

- Preliminaries
- Web in Practice
 - Search in Web 2.0
 - Microformats and Mashups
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Web 2.0 (from Wikipedia)



Rich Set of Buzzwords

(web) Search is a Basic Necessity



A (grossly inadequate) analogy: Toilets and Web 2.0

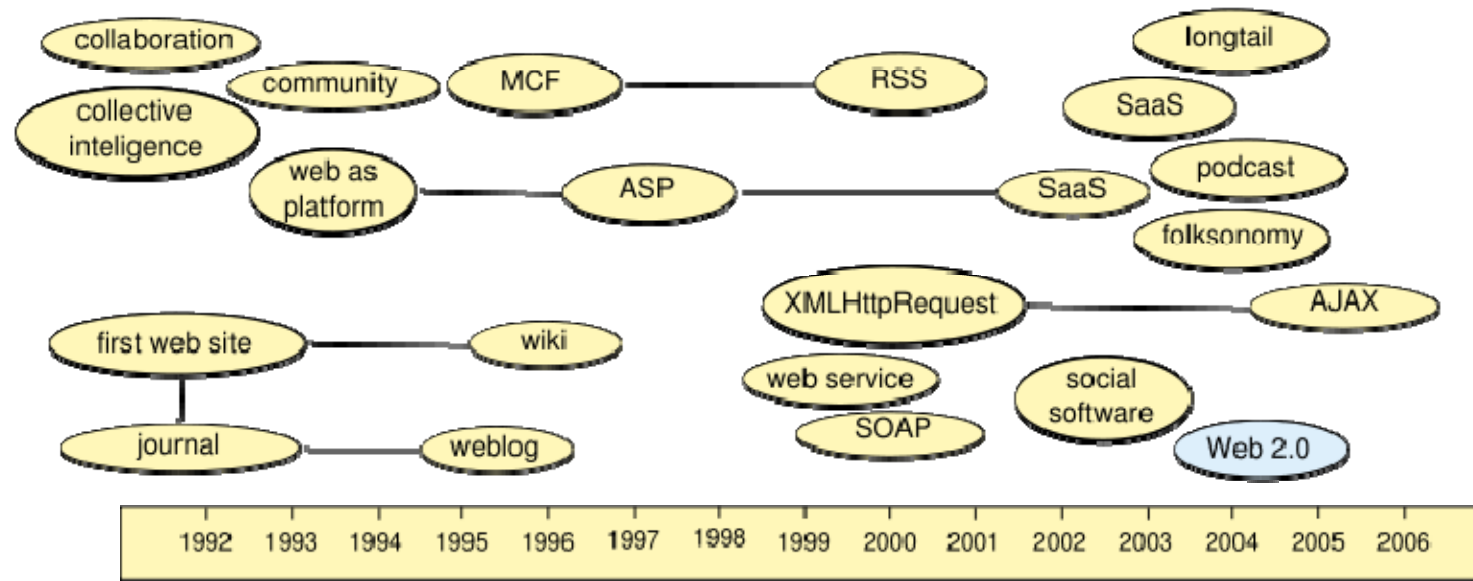
"Rich societies have developed quite complicated and expensive systems for removing human wastes from houses and cities, usually by dumping them, treated to one degree or another, into subsoils or bodies of water." Peter Bane, 2006

Rich Standard Infrastructure



Standard Pipes

XML

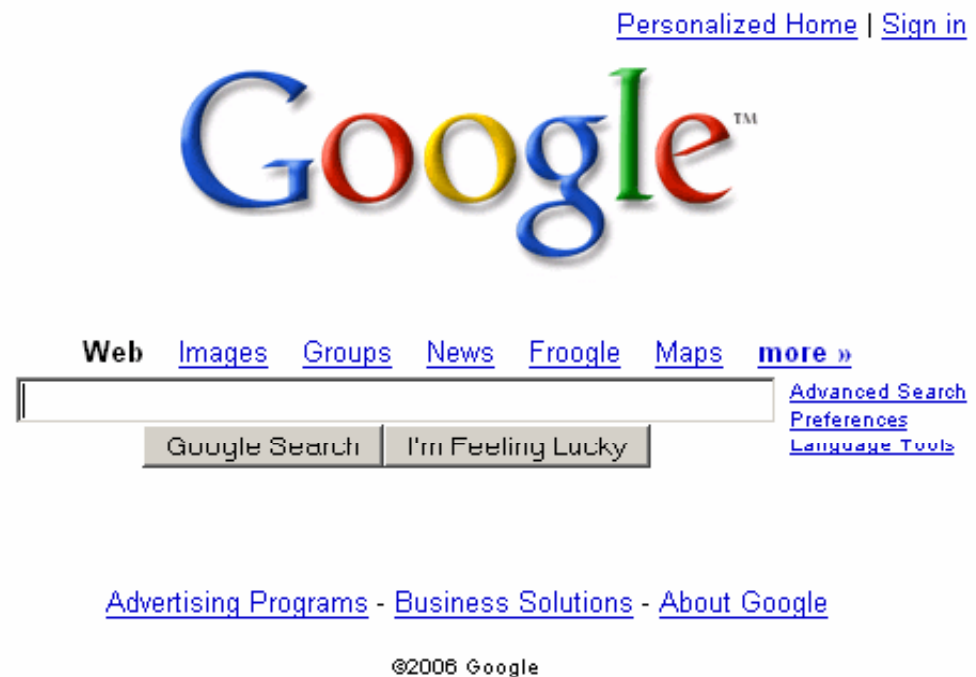


Big Infrastructure Sites



Water Treatment Plants

Search Engines Portals



Community Sites



The Importance of Mobility

The need to carry around technological solutions to basic necessities



Most Commonly Used is ...



Squat toilet

YAHOO! BUZZ Welcome, mconsens [Sign Out, My Account] Buzz Index Home - Help

TOP SEARCHES OF 2006

Want to play around more? [Make it interactive](#)

Top 10 Overall Searches	Top 10 News Story Searches	A Great Year in Pictures
1 Britney Spears	1 Steve Irwin death	 Tajikistan
2 WWE	2 Anna Nicole's son dies	Babasteve
3 Shakira	3 Iraq	 Joy Ride
4 Jessica Simpson	4 Israel and Lebanon	Kris Kros
5 Paris Hilton	5 U.S. elections	
6 American Idol	6 Fidel Castro stroke	
7 Beyonce Knowles	7 North Korea nuke	
8 Chris Brown	8 JonBenet confession	
9 Pamela Anderson	9 Saddam Hussein trial	
10 Lindsay Lohan	10 Danish cartoon	

"most popular searches" (2-3 keywords)

There are simple and sophisticated solutions to basic necessities

Need for more sophisticated search

Overview

- Preliminaries
- Web in Practice
 - Search in Web 2.0
 - Microformats and Mashups
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Microformats



- Community data formats
 - Personal Data: [hCard](#) (vCard)
 - Calendar and Events: [hCal](#) (iCal)
 - Social Networking: [XFN](#)
 - Reviews: [hReview](#)
 - Licenses: [rel-license](#)
 - Folksonomies: [rel-tag](#)
- Embedded in XHTML pages and RSS feeds
 - Also RSS Extensions (iTunes, Yahoo! Media, Geo, Google Base, 20+ more in use)

Example: hCal

```
<strong class="summary">Fashion Expo</strong> in  
<span class="location">Paris, France</span>:  
<abbr class="dtstart" title="2006-10-20">Oct 20</abbr>  
to <abbr class="dtend" title="2006-10-23">22</abbr>
```

- Large and growing list of websites
 - [Eventful.com](#)
 - [LinkedIn](#)
 - [Yedda](#)
 - [upcoming.yahoo.com](#)
 - [Yahoo! Local](#), [Yahoo! Tech Reviews](#)
- Benefit from shared tools, practices (hCalendar creator, iCal Extraction)

Semantic Mashups

- A "semantic" mashup can
 - Contact (hCard)
 - Friends (XFN,FOAF)
 - To attend a recommended event (hCal,hReview)
- Microformats are the lower-case semantic web
- Also Machine Tags (eg, flickr:user=me)
 - Tags that use a special syntax to define extra information about a tag
 - Have a namespace, a predicate and a value (sounds familiar?)

Search in Mashup Creation

The screenshot displays the pipes.io interface for a mashup titled "Upcoming.org Combined Feed". The interface includes a sidebar with various modules like "Fetch CSV", "Feed Auto-Disco", "Fetch Feed", "Fetch Data", "Fetch Site Feed", "Flickr", "Google Base", "Item Builder", "Yahoo! Local", and "Yahoo! Search". The main workspace shows a flowchart of modules connected by wires. The modules include:

- URL Builder**: Base: `http://upcoming.org`, Path elements: `text [wired]`, Query parameters: `text : text`.
- Fetch Feed**: URL: `http://upcoming.org/news/index.xml`.
- Sort**: Sort by: `item.pubDate` in `descending` order.
- Upcoming ID (text)**: Name: ID, Prompt: Upcoming ID, Position: `number`, Default: `text`, Debug: 20090.
- Hash (text)**: Name: Hash, Prompt: Hash, Position: `number`, Default: `text`, Debug: 1176cd2473.
- URL Builder**: Base: `http://`, Path elements: `text [wired]`, Query parameters: `text`.

The flowchart shows the "Fetch Feed" module connected to the "Sort" module, which is then connected to the "Upcoming ID (text)" and "Hash (text)" modules. The "URL Builder" modules are also connected to the "Fetch Feed" and "Sort" modules. The "Pipe Output" module is connected to the "Sort" module. The interface also shows a "Debugger" at the bottom right displaying "Pipe Output (126 items)".

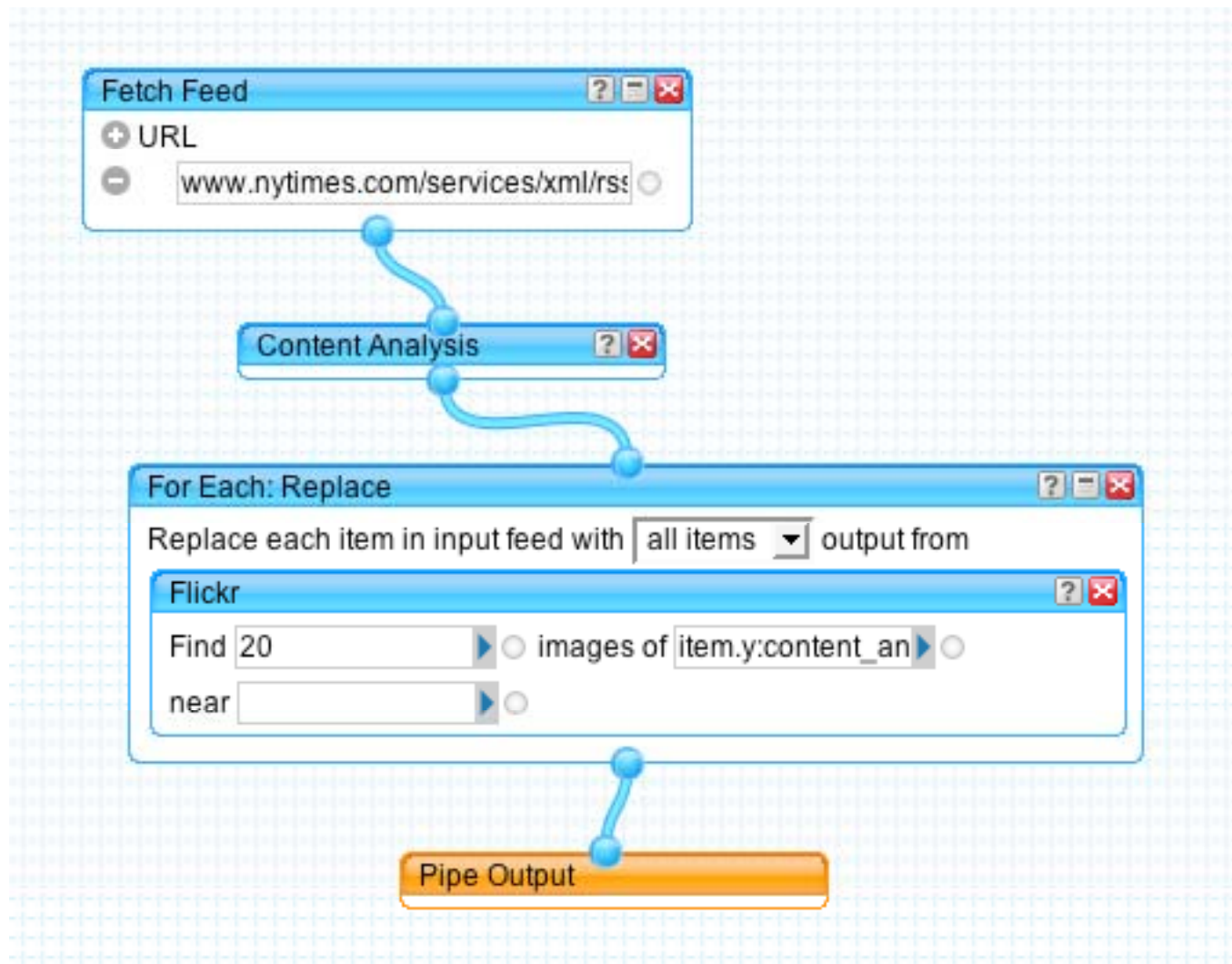
Fetch Data
This module retrieves any XML, JSON, iCal or KML file and tries to extract a list of elements using the provided path parameter.
Example: [Using the Fetch Data Module](#)
Learn more: [about this module](#)

Time taken: 0.626039s [Refresh](#)
▶ **lh28** posted a new comment to **Dwell on design**
▶ **Jan 28, 2008: Web Directions North 2008**

Mashup Tools

- Microsoft Popfly
- IBM ProjectZero
- Yahoo! Pipes
 - Allows developers to mash-up web data
 - drag and drop editor which enables user to connect multiple Internet data sources
 - a source is grabbed and searched!
 - both content and structure are queried

Yahoo! Pipes Demo



Yahoo! Pipes Demo

The screenshot displays the Yahoo! Pipes web interface. At the top, the title bar reads "Copy of Example: Using the Flickr Module" and "Pipe Saved". A "Run Pipe..." button is visible. Below the title bar, there are navigation buttons: "Layout", "Expand All", "Collapse All", "Back to My Pipes", "New", "Save", "Save a copy", and "Properties...".

On the left side, there is a "Sources" panel with a list of modules: Fetch CSV, Feed Auto-Disco, Fetch Feed, Fetch Data, Fetch Site Feed, Flickr, Google Base, Item Builder, Yahoo! Local, and Yahoo! Search. Below this are "User inputs", "Operators", "Url", "String", "Date", "Location", "Number", and "Favorites".

The main workspace shows a pipe configuration. A "Flickr" module is connected to a "Pipe Output" module. The Flickr module has the following configuration: "Find 10" (radio button selected), "images of" (radio button selected), "squat toilet" (text input), and "near vienna" (text input). A blue line connects the Flickr module to the Pipe Output module.

At the bottom right, there is a "Debugger: none" button.

Yahoo! Pipes Demo Result

Copy of Example: Using the Flickr Module

This Pipe demonstrates how you can use the Flickr module.

☆ [Edit Source](#) [Delete](#) [Publish](#) [Clone](#)

Use this Pipe

[+ MY YAHOO!](#) [+ Add to Google](#) [🔔 Get results by Email or Phone](#) [📡 More options ▶](#)

List

10 items

worst case scenario



Compost toilet interior



Family plan :-)



Catharine at Dinner



Author:
mconsens
([Edit profile](#))

Properties:

Not published
0 runs
0 clones

Tags:

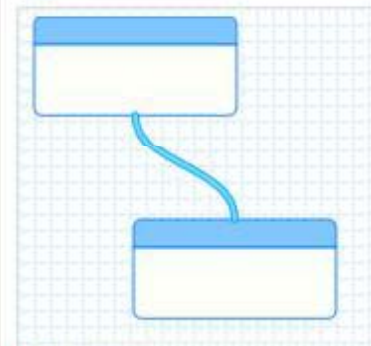
[add new tag](#)

Sources:

[flickr.com](#)
[api.flickr.com](#)

Modules:

flickr



[Edit Source](#)

Overview

- Preliminaries
- Web in Practice
 - Search in Web 2.0
 - Microformats and Mashups
- DB/IR in Theory
 - Retrieval Languages and Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Take Away

- Search is crucial when accessing Web 2.0 sources
- There is already demand for exploiting additional structure in Web 2.0 search
- Structure (XML) retrieval needs to:
 - be exposed to users/developers
 - support rich, context-dependent semantics
 - address efficiency and effectiveness

Overview

- Preliminaries
- Web in Practice
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Languages

- Keyword search
 - "squat"
- Tag + Keyword search
 - description: squat
- Path Expression + Keyword search
 - //image[./title about "squat"]
- XQuery + Complex full-text search
 - for \$i in //image
let score \$s := \$i ftscore "squat" && "toilet"
distance 2

Overview

- Preliminaries
- Web in Practice
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Retrieval Semantics

- Structure search incorporates conditions on the underlying structure of a collection
 - Schemas help
 - Schemas **prescribe** data and help **validation**
 - Provide **limited description** of valid instances
- New semantics
 - Lowest Common Ancestor
 - Query relaxation
 - Overlapping elements

Lowest Common Ancestor

- Retrieve most relevant fragment
-
- References:
 - Nearest Concept Queries (Schmidt et al, ICDE 2002)
 - XRank (Guo et al, SIGMOD 2003)
 - SchemaFree XQuery (Li et al VLDB 2004)
 - XKSearch (Xu & Papakonstantinou, SIGMOD 2005)

XRank

<workshop date="28 July 2000">

<title> XML and Information Retrieval: A SIGIR 2000 Workshop </title>

<editors> David Carmel, Yoelle Maarek, Aya Soffer </editors>

<proceedings>

<paper id="1">

<title> XQL and Proximal Nodes </title>

<author> Ricardo Baeza-Yates </author>

<author> Gonzalo Navarro </author>

<abstract> We consider the recently proposed language ... </abstract>

<section name="Introduction">

Searching on structured text is becoming more important with XML ...

→ <subsection name="Related Work">

The XQL language ...

</subsection>

</section>

...

<cite xmlns:xlink="http://www.acm.org/www8/paper/xmlql"> ... </cite>

</paper>

(Guo et al, SIGMOD 2003)

XRank

```
<workshop date="28 July 2000">
```

```
  <title> XML and Information Retrieval: A SIGIR 2000 Workshop </title>
```

```
  <editors> David Carmel, Yoelle Maarek, Aya Soffer </editors>
```

```
  <proceedings>
```

```
    <paper id="1">
```

```
      <title> XQL and Proximal Nodes </title>
```

```
      <author> Ricardo Baeza-Yates </author>
```

```
      <author> Gonzalo Navarro </author>
```

```
      <abstract> We consider the recently proposed language ... </abstract>
```

```
      <section name="Introduction">
```

```
        Searching on structured text is becoming more important with XML ...
```

```
        <subsection name="Related Work">
```

```
          The XQL language ...
```

```
        </subsection>
```

```
      </section>
```

```
    ...
```

```
    <cite xmlns:xlink="http://www.acm.org/www8/paper/xmlql" ... </cite>
```

```
  </paper>
```

```
...
```

XIRQL

```
<workshop date="28 July 2000">
  <title> XML and Information Retrieval: A SIGIR 2000 Workshop </title>
  <editors> David Carmel, Yoelle Maarek, Aya Soffer </editors>
  <proceedings>
    <paper id="1">
      <title> XQL and Proximal Nodes </title>
      <author> Ricardo Baeza-Yates </author>
      <author> Gonzalo Navarro </author>
      <abstract> We consider the recently proposed language ... </abstract>
      <section name="Introduction">
        Searching on structured text is becoming more important with XML ...
        <em>The XQL language </em>
      </section>
      ...
      <cite xmlns:xlink="http://www.acm.org/www8/paper/xmlql"> ... </cite>
    </paper>
  </proceedings>

```

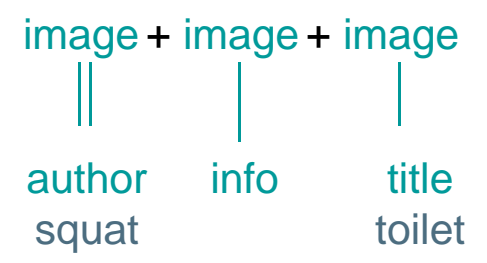
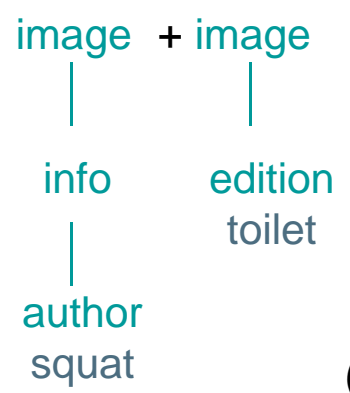
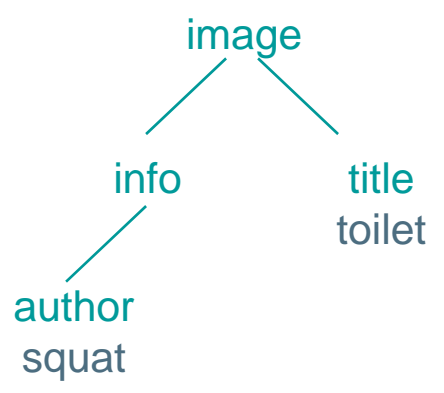
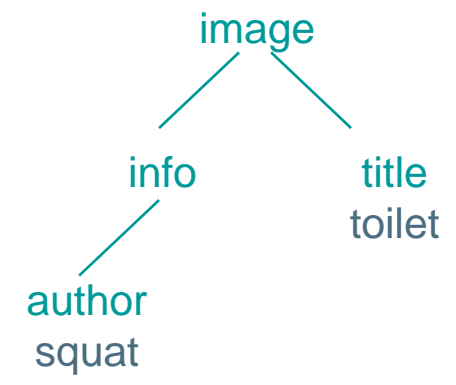
(Fuhr & Großjohann, SIGIR 2001)

...

XML Query Relaxation

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

Query

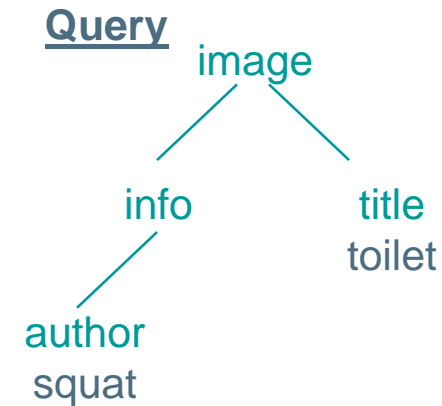


(Amer-Yahia et al, VLDB 2005)

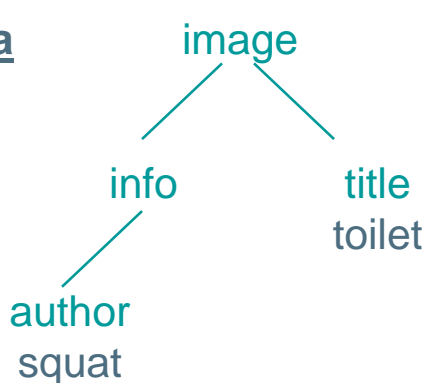
XML Query Relaxation

■ Tree pattern relaxations:

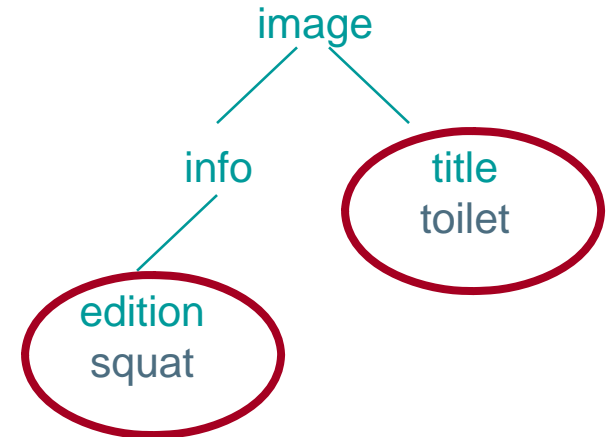
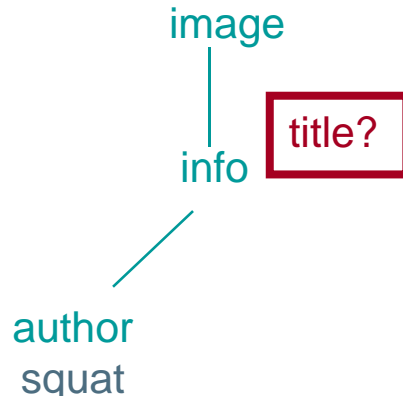
- Leaf node deletion
- Edge generalization
- Subtree promotion



Data



(Amer-Yahia, SIGMOD 2004) (Schlieder, EDBT 2002)
(Delobel & Rousset, 2002)



Controlling Overlap

What most approaches are doing:

- Given a ranked list of elements:

1. select element with the highest score within a path
2. discard all ancestors and descendants
3. go to step 1 until all elements have been dealt with

- (Also referred to as brute-force filtering)

Post-Processing Overlap

- Sometimes with some "prior" processing to affect ranking:
 - Use of a utility function that captures the amount of useful information in an element
$$\text{Element score} * \text{Element size} * \text{Amount of relevant information}$$
 - Used as a prior probability
 - Then apply "brute-force" overlap removal

(Mihajlovic etal, INEX 2005; Ramirez etal, FQAS 2006))

Post-Processing Overlap

- Score of elements containing or contained within higher ranking components are iteratively adjusted
(depends on amount of overlap "allowed")
 1. Select the highest ranking component.
 2. Adjust the retrieval status value of the other components.
 3. Repeat steps 1 and 2 until the top m components have been selected.

(Clarke, SIGIR 2005)

Post-Processing Overlap

Smart filtering

(Mass & Mandelbrod, INEX 2005)

Given a list of rank elements

-group elements per article

-build a result tree

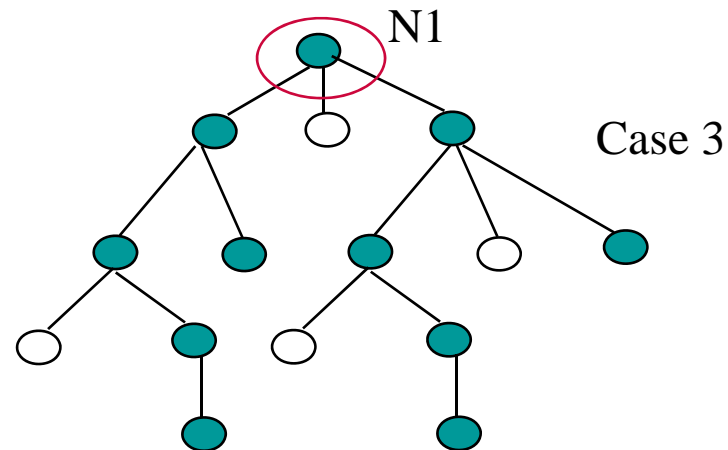
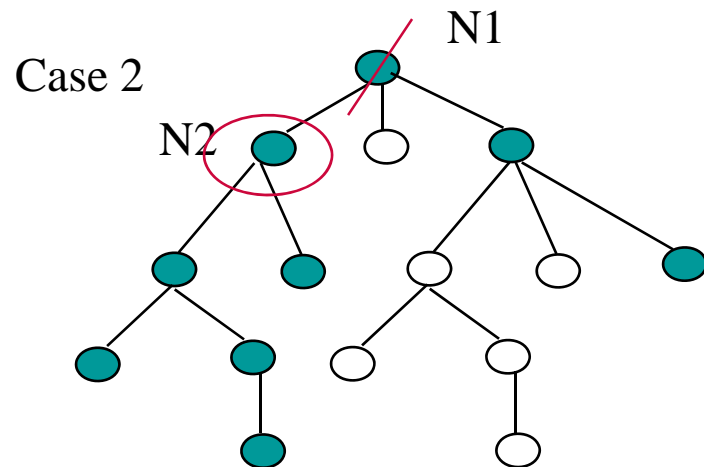
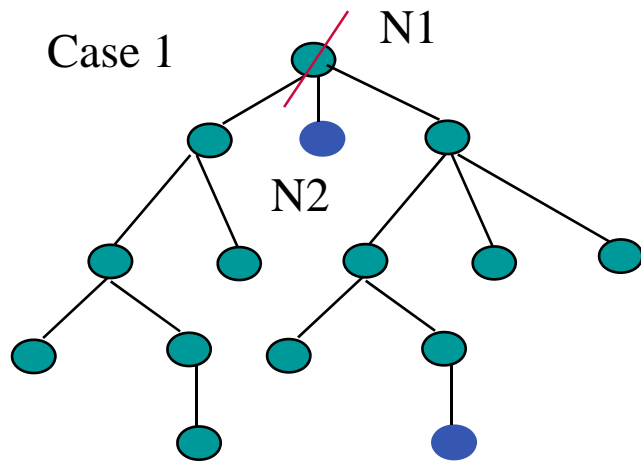
-“score grouping”: ● ● ○

-for each element N1

1. $\text{score } N2 > \text{score } N1$

2. concentration of good elements

3. even distribution of good elements



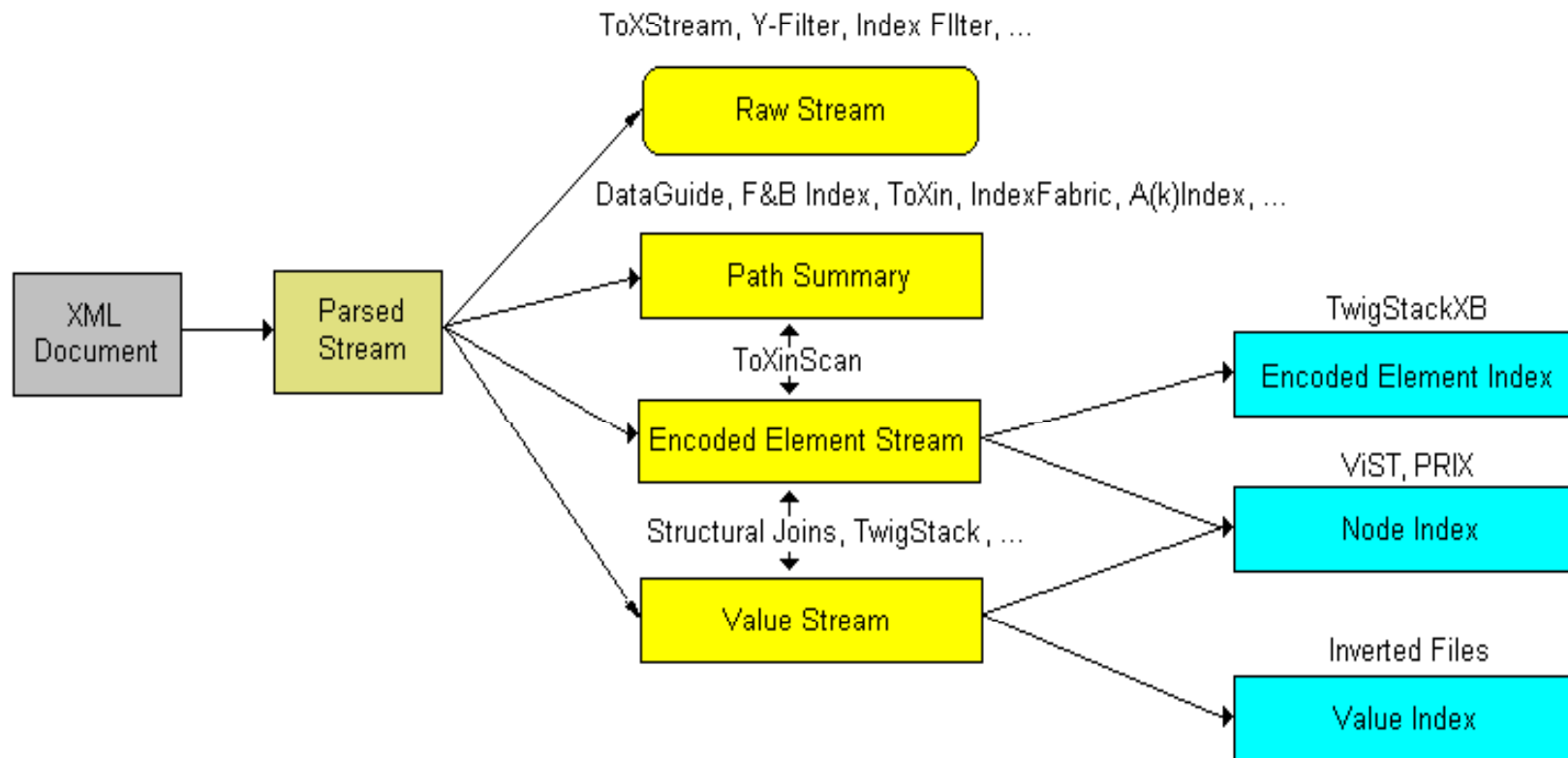
Languages

- Keyword search (CO Queries)
 - "xml"
- Tag + Keyword search
 - book: xml
- Path Expression + Keyword search (CAS Queries)
 - /book[./title about "xml db"]
- XQuery + Complex full-text search
 - for \$b in /book
let score \$s := \$b ftcontains "xml" && "db"
distance 5

Overview

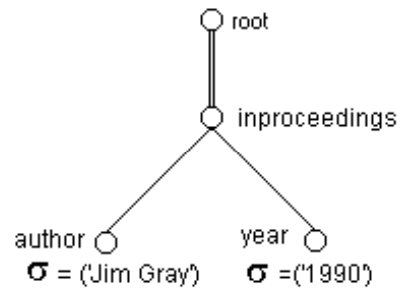
- Preliminaries
- Web in Practice
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Encodings, Summaries, Indexes Access Methods



Stack Algorithms

```
<dblp>
...
<inproceedings>
  <author> a1 </author>
  <author> a2 </author>
  <year> y1 </year>
</inproceedings>
...
<article>
  <author> a3 </author>
  <author> a4 </author>
  <year> y2 </year>
</article>
...
<inproceedings>
  <author> a5 </author>
  <author> a6 </author>
  <year> y3 </year>
</inproceedings>
...
.</dblp>
```



Region algebra encoding

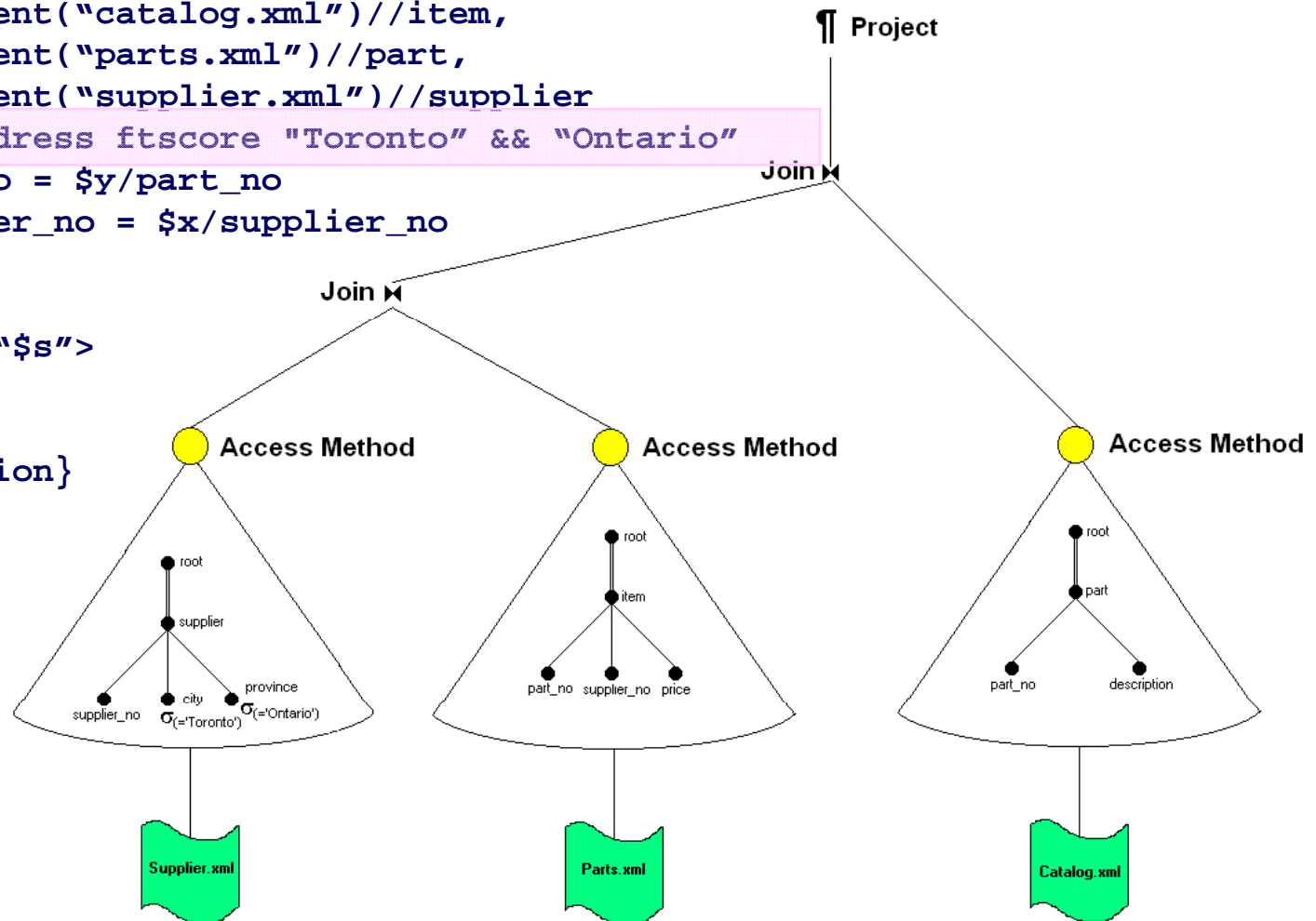
- Elements [DocID, Element, Start, End, LevelNum]
- Values [DocID, Value, Start, LevelNum]

Structural Summaries

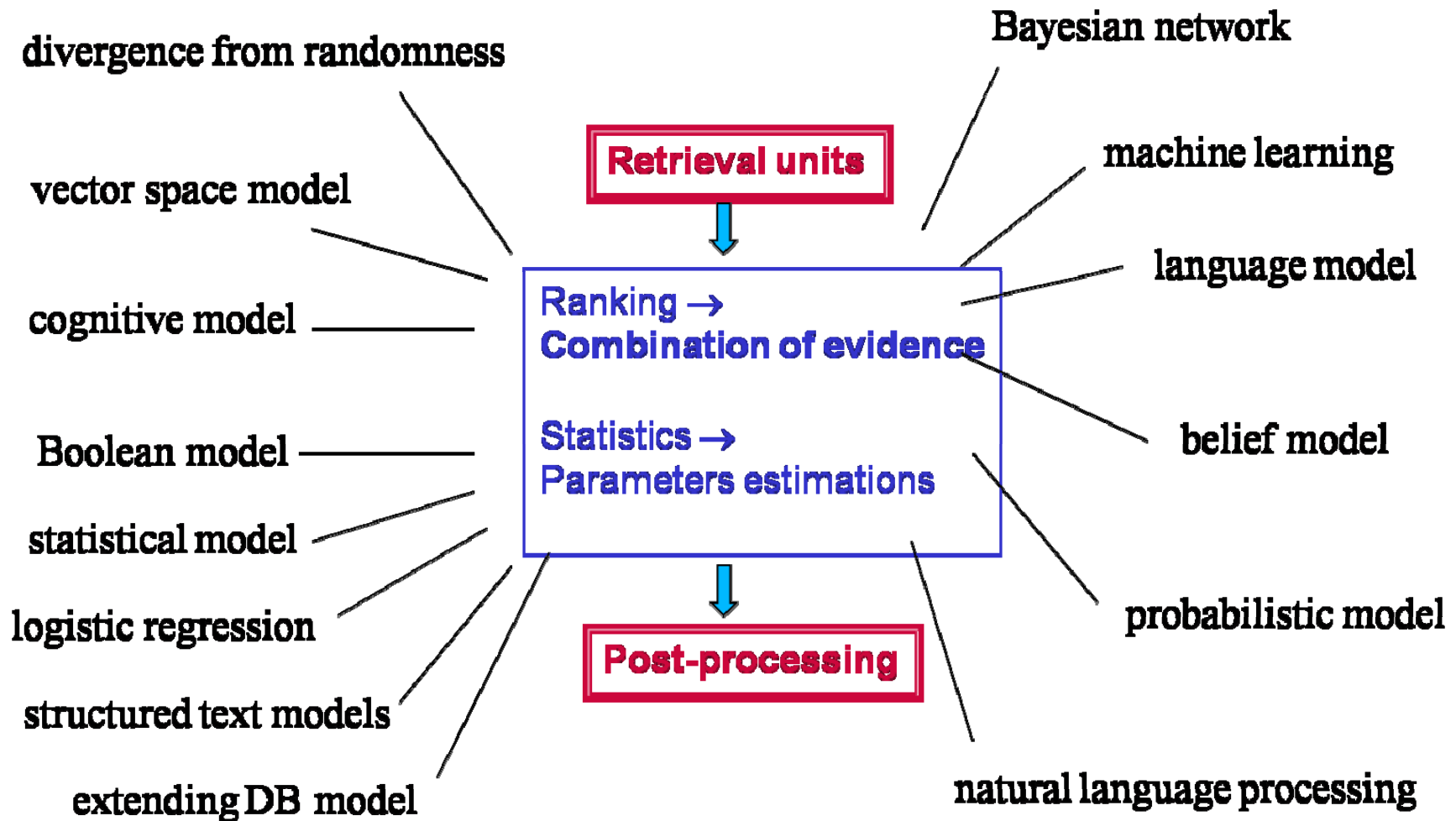
- XML structural summaries are graphs representing **relationships** between sets in a **partition** of XML elements.
 - Many proposals
 - Region inclusion graphs (RIGs) [CM94], representative objects (ROs) [NUWC97], **dataguides** [GW97], 1-index, 2-index and T-index [MS99], ToXin [RM01], XSKETCH [PG02], APEX [CMS02], A(k)-index [KSBG02], F+B-Index and F&B-Index [KBNK02], D(k)-index [QLO03], M(k)-index [HY04], **Skeleton** [BCFH+05], XCLUSTER [PG06]
 - AxPRE (axis path r.e.) Summaries answer
 - How are all these summaries related?
 - Can they be constructed together?
 - Can they be used [for **query evaluation**] together?
-

Query Processing

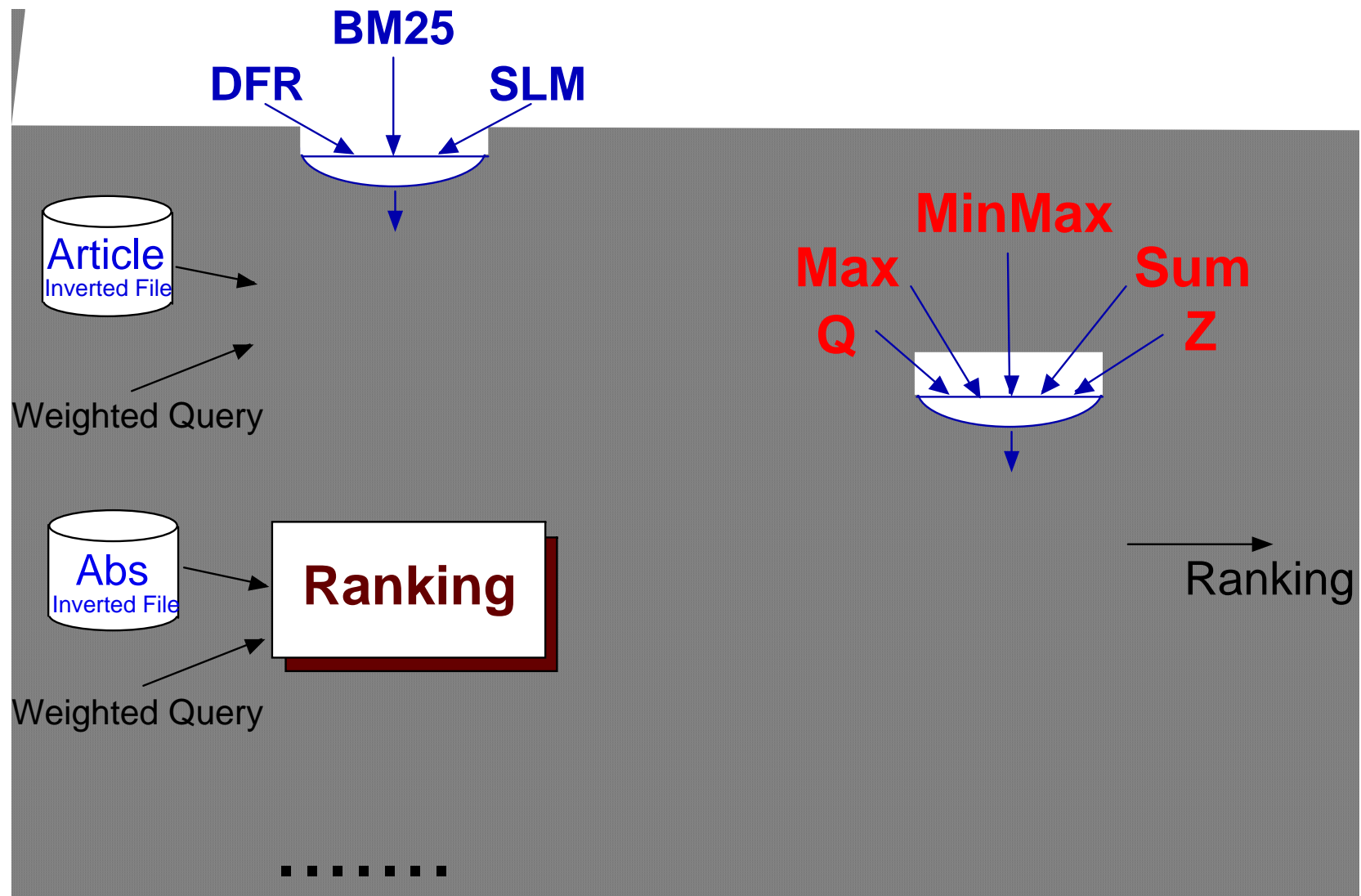
```
for $x in document("catalog.xml")//item,  
    $y in document("parts.xml")//part,  
    $z in document("supplier.xml")//supplier  
let $s := $z/address ftscore "Toronto" && "Ontario"  
where $x/part_no = $y/part_no  
    and $z/supplier_no = $x/supplier_no  
order by $s  
return  
  <result score="$s">  
    {$x/part_no}  
    {$x/price}  
    {$y/description}  
  </result>
```



Retrieval models



Score Combination



Preliminaries for Top-k Retrieval

- Each object is **scored** using different criteria
 - Score (or grade) is a value, usually $[0,1]$
- **Criterion** (e.g., a keyword) refers attributes or keywords specified in the query
- Each criterion has a **sorted list** of $R(\text{objects}, \text{score})$
- The combined score is computed using an **Aggregation function** $t(x_1, x_2, \dots, x_m)$
 - If $x_i \leq x'_i$ for every i , then $t(x_1, x_2, \dots, x_m) \leq t(x'_1, x'_2, \dots, x'_m)$
 - Examples: average, weighted sum, min, max, etc.
- **Goal**
 - Merge ranked results to find the **best top-k** answers

Threshold Algorithm (TA) [FLN'01]

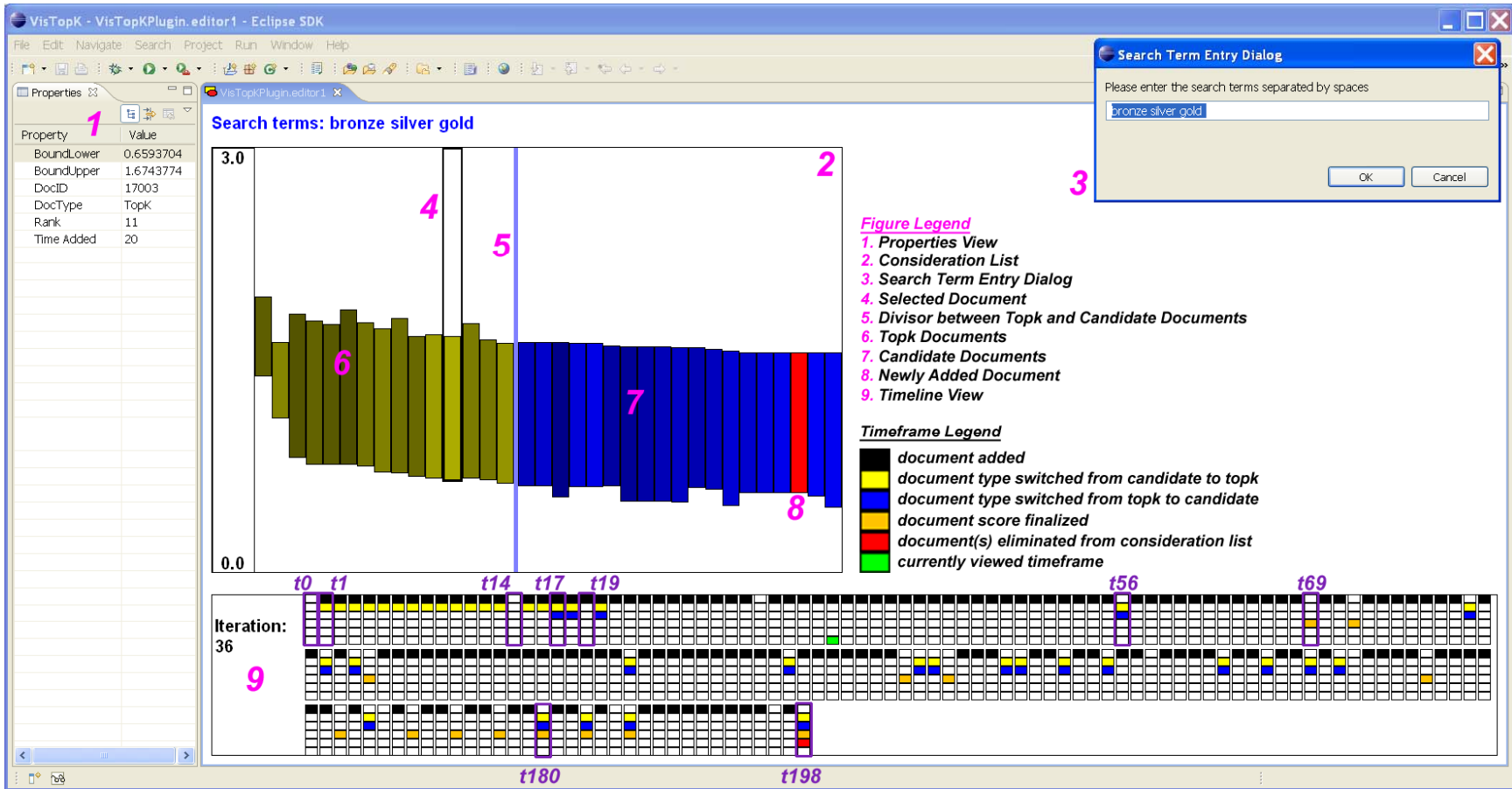
- **Sorted** access in parallel to each of the m lists
- **Random** access for every new object seen in every other list to find i -th field x_i of R .
- Use **aggregation function** $t(R) = t(x_1, x_2, \dots, x_m)$ to calculate grade and store it in set Y **only** if it belongs to current **top-k** objects.
- Calculate **threshold** value $T = t(\underline{x}_1, \underline{x}_2, \dots, \underline{x}_m)$ of aggregate function after every **sorted** access, stop when k objects have grade at least T
- Return set Y which has **top-k values**

- **Analysis: TA Optimal over every instance**
 - but... big O , and don't forget assumptions

Variations of TA

- **NRA**: When no random access (RA) is possible
 - Example: Web search engines, which typically do not allow you to enter a URL and get its ranking
- **TA_z**: When no sorted access (SA) is possible for some predicates
 - Example: Find good restaurants near location x (sorted and random access for restaurant ratings, random access only for distances from a mapping site)
- **CA**: When the relative costs of random and sorted accesses matter (TA+NRA).
- **TA_θ**: Only when approximate answers are needed
 - Example: Web search, with lots of good quality answers
- SA/RA scheduling problem, **IO-Top-K** [BMSTW'06]

VisTopK Demo



Overview

- Preliminaries
- Web in Practice
- DB/IR in Theory
 - Query Languages
 - Retrieval Semantics
 - Evaluation *à la* DB (Query Processing)
 - Evaluation *à la* DB (Relevance Assessments)
- Challenges

Evaluation of XML retrieval: INEX

- Evaluating the effectiveness of **content-oriented XML** retrieval approaches like TREC
- **Collaborative** effort \Rightarrow participants contribute to the development of the collection (IEEE and Wikipedia)
 - queries
 - relevance assessments
 - methodology
- **Content-only (CO) topics**
 - Ignore document structure
- **Content-and-structure (CAS) topics**
 - Contain conditions referring both to content and structure of the sought elements
 - Conditions may or may not be strict



CAS topics 2003-2004

<title>

```
//article[(./fm//yr = '2000' OR ./fm//yr = '1999') AND about(.,  
  "intelligent transportation system")]//sec[about(., 'automation  
+vehicle')]
```

</title>

<description>

Automated vehicle applications in articles from 1999 or 2000 about intelligent transportation systems.

</description>

<narrative>

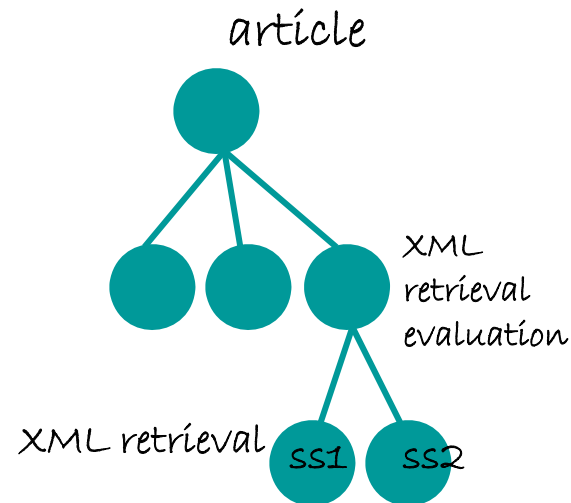
To be relevant, the target component must be from an article on intelligent transportation systems published in 1999 or 2000 and must include a section which discusses automated vehicle applications, proposed or implemented, in an intelligent transportation system.

</narrative>

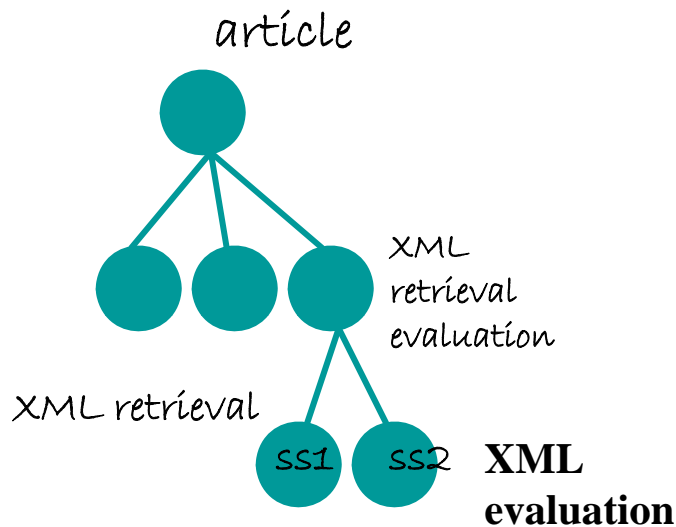
Relevance in XML retrieval

- 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**

(Borlund, JASIST 2003)
(Goevert et al., JIR 2006)



Relevance in XML retrieval: INEX 2003 - 2004



- 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, ...**

Specificity Dimension 2005

continuous scale defined as ratio (in characters) of the highlighted text to element size

User unichile | Links | Pool | X-Rai > Demo pool > ieee > dt > dt/1999 >
File dt/1999/d1053

The earlier type of integration uses a DRAM macro embedded on an application specific IC (ASIC). For this purpose, designers have developed reconfigurable DRAM macros for many applications, providing a different configuration for each application. Although the many applications require a wide variety of configurations, the macro-testing methodology must be unified to reduce product-testing costs. This article describes circuitry that helps simplify testing the embedded-DRAM macro on an ASIC.>

>

<<TESTING DILEMMA>

<The dilemma in testing embedded DRAM arises from differences in character between ASICs and commodity DRAMs. In the case of commodity DRAMs, despite huge amounts of production, manufacturers produce only a few different products at the same time. { As a result, they can optimize the testing methodology for each product. In contrast, companies produce a large variety of ASIC products, but the production volume of each product is small. Also, ASICs require a very short turnaround time. Therefore, customizing the test methodology for each product is difficult. ASICs require a common test environment that covers all product variations. }>

<Furthermore, since the commodity DRAM is a general-purpose product, we cannot specify its application during testing. Thus, testing must cover various kinds of applications and provide very a result, the commodity DRAM's test time is longer than the ASIC's.

1

Measuring effectiveness: Metrics

- Inex_eval (also known as inex2002) (Goevert & Kazai, INEX 2002)
official INEX metric 2002-2004
- Inex_eval_ng (also known as inex2003) (Goevert et al, JIR 2006)
- ERR (expected ratio of relevant units) (Piwowarski & Gallinari, INEX 2003)
- xCG (XML cumulative gain) (Kazai & Lalmas, TOIS 2006)
official INEX metric 2005-
- t2i (tolerance to irrelevance) (de Vries *et al*, RIAO 2004)
- EPRUM (Expected Precision Recall with User Modelling) (Piwowarski & Dupret, SIGIR 2006)
- HiXEval (Highlighting XML Retrieval Evaluation) (Pehcevski & Thom, INEX 2005)
official INEX metric 2007
- Structural Relevance (Ali & Consens & Lalmas, SIGIR Element Retrieval Workshop 2007)

Overview

- Preliminaries
- Web in Practice
- DB/IR in Theory
- Challenges

Challenges

- In practice, user interfaces are key
 - Combine sources of information
 - Provide feedback on retrieval results
- Interaction between traditional DB query optimization and ranking/top-k
- What are the useful extensions to keyword querying that incorporate structural information?

Challenges

- Indexing, Searching, Ranking
 - Efficient (and Effective) algorithms
- INEX-like test collection and effectiveness
 - Too complex?
 - What constitutes a retrieval baseline?
 - What is a good measure?
 - Generalisation of the results on other data sets
- Quality evaluation (Web, XML)
 - Who are the users?
 - What are their information needs?
 - What are the requirements?

Challenges Ahead

- Lots of opportunities
 - To understand the structure of data
 - To exploit structure in searches
 - To measure and improve search quality
- Can search remain a **joy to use** when users are allowed to
 - Contribute content? (Wikipedia)
 - Share it? (Flickr)
 - rate it? (YouTube)