

XML Retrieval: From elements to relevant elements

Börkur Sigurbjörnsson

Jaap Kamps

Maarten de Rijke

Language & Inference Technology Group

University of Amsterdam

December 15th 2003

Strategic lines

- ▶ Finding the appropriate unit of retrieval
 - ◇ Lessons from last year's evaluation
 - The average **element** is **small**
 - average length 29; median length 2
 - The average **relevant element** is **bigger**
 - average length 1469; median length 220
 - ◇ How to go from an average **element** to an average **relevant element**?
- ▶ Mixing **multiple evidence**
 - ◇ Ad-hoc retrieval:
 - mix two levels, **documents** and the **collection**
 - ◇ XML retrieval:
 - also mix evidence from **various** levels of the XML hierarchy

Indexing

- ▶ First, we need to index the data
 - ◇ Article index – whole articles are the indexing unit
 - ◇ Element index – each element is an indexing unit

simple.xml

```
<article>
  <au> Tom Waits </au>
  <sec> Champagne for my real friends </sec>
  <sec> Real pain for my sham friends </sec>
</article>
```



- ▶ No stemming, but case-folding and stopword removal

Query processing

- ▶ Only `<title>` and `<description>`
- ▶ No support for `+`, `-` or `phrases`
 - ◇ Removed words and phrases bounded by `-`
 - ◇ Removed the tokens `+` and `"`
- ▶ No stemming, but `case-folding` and `stopword` removal
- ▶ Blind feedback
 - ◇ Top `10` articles considered relevant
 - ◇ Terms appearing `more often than expected` are added to the query
 - ◇ We added up to `20` terms to the original queries

Retrieval model

- ▶ Language model
 - ◇ Estimate a **language model** for each element
 - ◇ Rank the elements by the **likelihood of the query**
 - ◇ The chance of getting the query by **random sampling** from element
- ▶ Smoothing
 - ◇ Estimation of element model adjusted by collection model
- ▶ Length prior
 - ◇ Element is assigned prior probability proportional to its length
- ▶ Index cut-off
 - ◇ Restriction of the element index to elements larger than n

Content Only Task

▶ How to determine the appropriate unit of retrieval?

- ◇ Using length prior [SIGIR 2003][INEX 2003]
- ◇ Tuning the smoothing parameter [DIR 2003][INEX 2003]
- ◇ Using the index cut-off [INEX 2003]
- ◇ Using blind feedback [INEX 2003]

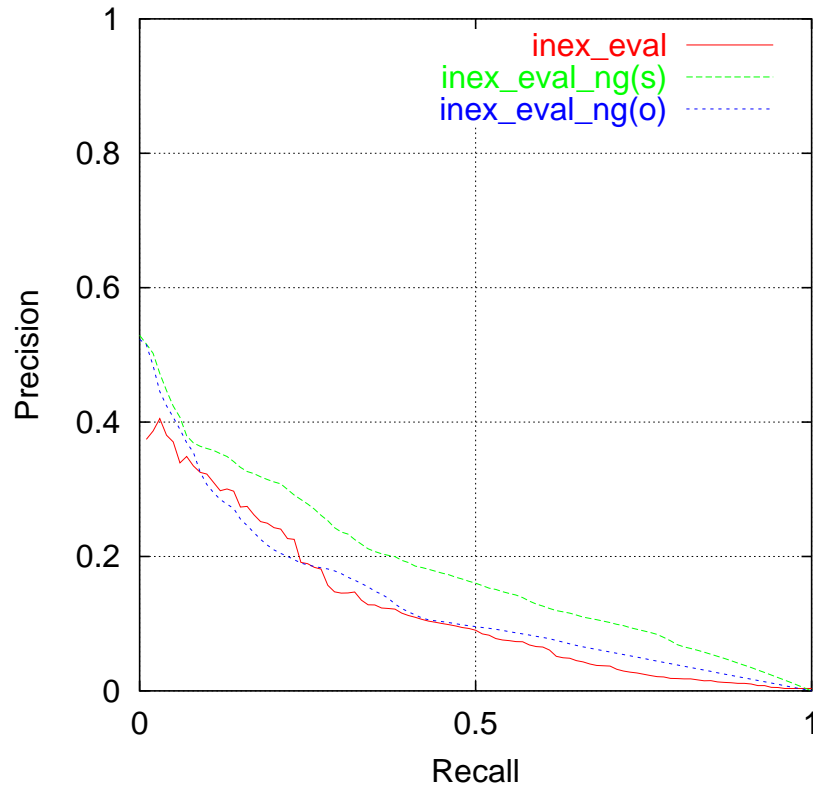
▶ How to mix evidence?

- ◇ Smooth our language models using the collection
- ◇ Mix element scores and article scores

▶ Our runs differ in value of the smoothing parameter

- ◇ Little difference in performance between runs

Results Content-Only Task



Evaluation metric	MAP
inex_eval	0.1214
inex_eval_ng(s)	0.1857
inex_eval_ng(o)	0.1367

- ▶ **Strict** evaluation of our **best scoring** run only
 - ◇ We seem to be retrieving **large** elements
 - ◇ We seem to be retrieving **overlapping** elements
 - ◇ The overlapping elements have **little effect** in the **low-recall** area

Strict Content-And-Structure Topics

- ▶ Original topic 90

Title //article[about(./sec,'trust authentication electronic commerce e-commerce e-business marketplace')]/abs[about(., 'trust authentication')]

Description Find abstracts of articles that discuss automated tools for establishing trust between parties on the Internet. The article should discuss applications of trust for authenticating parties in e-commerce.

- ▶ Our versions of topic 90

90TD trust authentication electronic commerce e commerce e business ...
discuss applications of trust for authenticating parties in e commerce

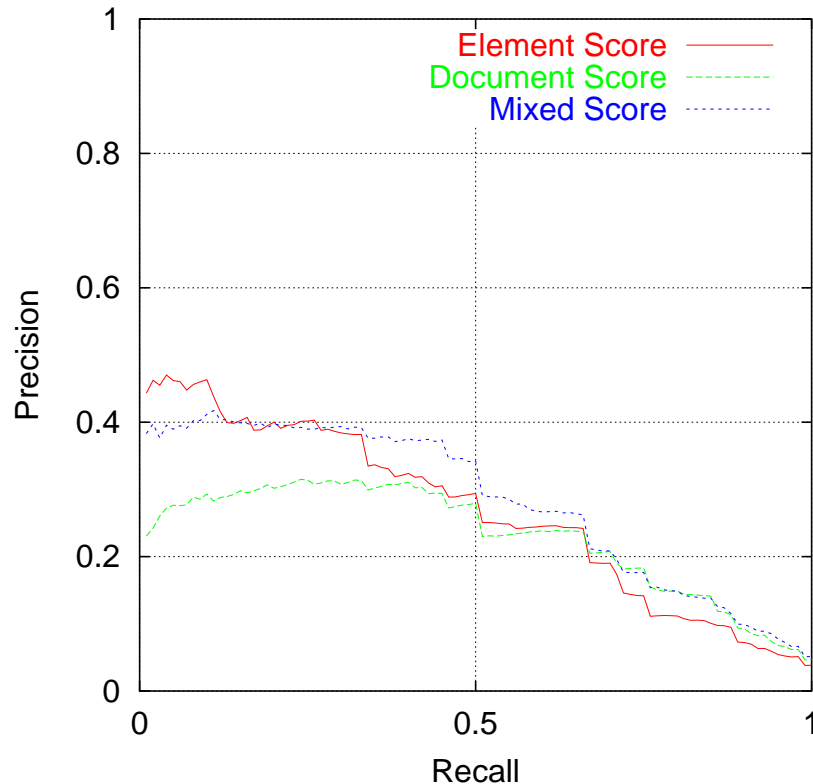
90Ta trust authentication electronic commerce e commerce e business
marketplace

90Tb trust authentication

Strict Content-And-Structure Task

- ▶ Assign **score** to elements satisfying an **XPath**-like expression
- ▶ We compute ...
 - ◇ scores for **articles**
 - ◇ scores for target **elements**
 - ◇ scores for each **about-filter**
- ▶ Elements were assigned ...
 - ◇ their own **element** score
 - ◇ the score of the **article** containing them
 - ◇ a **mixed** score, using ...
 - the score of the **article**
 - their own **element** score
 - the score of all elements matching **about-filters**

Results Strict Content-And-Structure Task



Run	MAP
ElementScore	0.2650
DocumentScore	0.2289
MixedScore	0.2815

- ▶ Strict evaluation of our SCAS runs
 - ◇ Using element scores helps
 - ◇ Mixing scores seems to help even more

Conclusions

- ▶ Content Only task
 - ◇ Finding the **appropriate unit of retrieval** is important
 - ◇ Tuning the **smoothing parameter** does not have a big impact here
 - The **cut-off** goes a long way to bridge the gap
- ▶ Strict Content-and-structure task
 - ◇ Using **element** scores is important
 - ◇ Mixing **multiple** scores is even more important
- ▶ Future work
 - ◇ Choosing the **appropriate statistics**
 - ◇ Further explore **size** and **mixing** issues
 - Integrate into a language model framework
 - ◇ Reduce **redundancy** in results