XML Retrieval: What to Retrieve?

Jaap Kamps Maarten Marx Maarten de Rijke Börkur Sigurbjörnsson
Language & Inference Technology Group
ILLC, University of Amsterdam
{kamps,marx,mdr,borkur}@science.uva.nl
http://lit.science.uva.nl/

ABSTRACT

The fundamental difference between standard information retrieval and XML retrieval is the unit of retrieval. In traditional IR, the unit of retrieval is fixed: it is the complete document. In XML retrieval, every XML element in a document is a retrievable unit. This makes XML retrieval more difficult: besides being relevant, a retrieved unit should be neither too large nor too small. The research presented here, a comparative analysis of two approaches to XML retrieval, aims to shed light on which XML elements should be retrieved. The experimental evaluation uses data from the Initiative for the Evaluation of XML retrieval (INEX 2002).

Categories and Subject Descriptors
H.2 [Database Management]: H.2.8 Database Applications; H.3 [Information Storage and Retrieval]: H.3.3 Information Search and Retrieval; H.3.4 Systems and Software
General Terms
Measurement, Performance, Experimentation

1. INTRODUCTION

Because the unit of retrieval is not fixed in XML retrieval there is, besides relevance, a second dimension for scoring retrieval results, called *coverage*, which indicates how much of the retrieved element is relevant. Units can be equally relevant but may differ on their coverage: the ideal is exact coverage, but obviously a relevant unit can be either too large or too small as well.

We compare two approaches to XML retrieval. As a baseline we created a system that does traditional IR: the index is built at the level of documents (an IEEE journal article in the INEX collection) and the unit of retrieval is fixed: complete articles. Earlier experiments found clear evidence that extracting XML elements from a ranked list of documents is a poor strategy [9]. This motivated our second approach: the index is built at the level of individual XML elements and every XML element is a potential unit of retrieval.

2. INEX TEST COLLECTION

The used data come from the Initiative for the Evaluation of XML retrieval (INEX) [1]. The collection to be queried contains over 12,000 articles from 21 IEEE Computer Society journals which are stored as XML documents. There are 169 different XML tags in the collection, such as complete articles, ⟨article⟩, abstracts, ⟨abs⟩, sections, ⟨sec⟩, and paragraphs, ⟨p⟩. Both the topic development and the assessments were done by the participants in INEX. Two types of topics were created. Content-only (CO) topics ignore the structure of the documents and, hence, are nothing but traditional IR topics. Content-and-structure (CAS) topics are aware of the documents' structure. Since for CAS topics the unit of retrieval is typically fixed and known, we focus on the CO topics in this paper. For each retrieved element, both the relevance and the coverage were assessed. One can define several measures out of these raw scores. Here we use the so-called *strict measure*. With this measure, an element is accepted if it is highly relevant and has exact coverage. Otherwise it is not accepted.[1]

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007.

3. COMPARATIVE ANALYSIS

For the runs in this paper, we experimented with two basic approaches to XML retrieval:

Full Document Retrieval System A baseline is formed by using a standard document index in which only whole documents are considered as a retrievable unit.

XML Element Retrieval System We created an XML element index in which each XML element is considered as a retrievable unit.

We evaluated our runs with version 0.006 of the `inex_eval` program, using version 1.8 of the relevance assessments. All runs contain up to 1000 results per topic. We did not apply stemming, nor feedback. Our retrieval model is a language model with various length priors [4].

Table 1: MAP of CO topics, using strict measure.

Run	Document Index	Element Index
No prior	0.0512	0.0227
Length prior	0.0562	0.0383
Cubic length	0.0572	0.0734

Table 1 shows the results of runs for both approaches. Two observations are immediate. First, in comparison with MAP scores for unstructured document retrieval, the scores are much lower. To put the scores in perspective, at INEX 2002, the best official run for strict CO scored 0.088 (using

[1]An alternative *generalized* measure has the undesirable property that a perfect run cannot obtain the perfect score of 1.0. This is due to the definition of generalized recall [5, p.1123]. For example, if there are two relevant documents for a topic with relevance scores 1 and 0.5, respectively, then the generalized precision at generalized recall level 1 is only 0.75.

Table 2: XML elements retrieved for CO topics.

No prior		Cubic length prior	
Tag name	Freq.	Tag name	Freq.
⟨p⟩	7265	⟨article⟩	12224
⟨atl⟩	3293	⟨bdy⟩	8012
⟨ti⟩	2218	⟨sec⟩	3505
⟨it⟩	2115	⟨p⟩	1532
⟨ip1⟩	1724	⟨ss1⟩	1025
⟨st⟩	1553	⟨bm⟩	788
⟨sec⟩	1468	⟨ip1⟩	371
⟨bb⟩	1164	⟨bibl⟩	281
⟨ss1⟩	792	⟨bib⟩	281
⟨b⟩	738	⟨atl⟩	224

100 results per topic) [2]. Second, although the evaluation is against highly relevant XML elements with exact coverage, the runs using the document index perform remarkably well. In fact, when we use no prior or a standard document length prior, the document index runs clearly outperform the XML element index runs. However, note that the effect of the length prior is much greater on the XML element index. This motivated experiments with even greater priors on document length. As it turns out, the XML element runs improve by using a cubic length prior—the prior probability of a document is proportional to the cube of its length. With this prior, the element index run outperforms the document index runs. In sum, the XML element index seems to be working better, but only if the retrieval is extremely biased toward longer elements.

Let us analyze this in more detail. In Table 2, we show the XML elements returned by two of the element-based runs. Without using a length prior, we return fairly small XML elements such as individual paragraphs, ⟨p⟩, and titles, ⟨atl⟩. With the cubic prior, we return mostly large elements such as whole articles, ⟨article⟩, and article-bodies, ⟨bdy⟩. Thus, the length-prior we had to use to boost our performance on the XML element index is effectively causing the retrieval of whole articles again. Are we using the wrong prior here? We have been experimenting with various priors, and ended up with the cubic length prior as the best performing one.

Let us look at the relevance assessments. Table 3 shows how often XML elements occur in the (strict) assessments and how often they occur in the collection. Without a length prior, the most frequently retrieved unit is paragraph, and indeed there are more relevant paragraphs than there are relevant articles. However, the number of paragraphs in the collection is so high, that the prior probability of relevance is extremely low. Finding relevant paragraphs amounts to a real needle-in-a-haystack problem. Articles, in contrast, have by far the highest prior probability of relevance. This makes them an attractive unit of retrieval, even though the task at hand is to retrieve highly relevant XML elements with exact coverage.

4. DISCUSSION AND CONCLUSIONS

Our official runs at INEX 2002 were based on a traditional document retrieval system, treating complete articles as the unit of retrieval [6]. For the CAS topics, we invested additional efforts in returning the required XML element from the initially retrieved documents. Based on the literature [9], we expected the traditional document-retrieval system's performance to be just a baseline for 'proper' XML retrieval systems, i.e., for systems that return smaller units than articles. Related work in passage retrieval showed that the combination of document-level and passage-level improves scoring [8], esp. for *document* retrieval (see also [7]). Much to our surprise, our runs based on a document index turned out to be among the top scoring submissions on INEX 2002.

Table 3: Prior probability of relevance of tags for the strict CO assessments (using implied scores).

Tag Name	Assessment Freq.	Collection Freq.	Prob. of Relevance
⟨article⟩	309	12107	0.0255
⟨bdy⟩	90	12107	0.0074
⟨sec⟩	291	69733	0.0042
⟨abs⟩	22	7359	0.0030
⟨ss1⟩	115	61490	0.0019
⟨ss2⟩	25	16288	0.0015
⟨fm⟩	13	12107	0.0011
⟨p⟩	383	747002	0.0005
⟨ip1⟩	61	183539	0.0003

In this paper, we experimented with two basic approaches to XML retrieval: (1) indexing whole documents; and (2) indexing individual XML elements in the collection. The aim of these experiments was to shed light on the unit of retrieval. This is related to [3], where an R-score biasing search toward medium sized XML elements is introduced. Our results indicate that a bias toward large sized XML elements is needed—size matters for XML retrieval!

How should we interpret this? On the one hand, the results show that a system returning entire articles is competitive to systems returning smaller units of text. This can be viewed as a positive result for it implies that the effectiveness of standard document retrieval techniques will carry over to XML retrieval. On the other hand, the results suggest that we do not yet fully understand how users and assessors perceive the coverage dimension of relevance. Our results show that users and assessors still regard whole articles as the meaningful unit of retrieval for XML collections.

5. REFERENCES

[1] N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors. *INEX 2002 Workshop Proceedings*, 2002.

[2] N. Gövert, M. Abolhassani, N. Fuhr, and K. Großjohann. Content oriented XML retrieval with HyREX. In Fuhr et al. [1], pages 13–17.

[3] K. Hatano, H. Kinutani, and M. Watanabe. An appropriate unit of retrieval results for XML document retrieval. In Fuhr et al. [1], pages 66–71.

[4] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[5] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *JASIST*, 53:1120–1129, 2002.

[6] M. Marx, J. Kamps, and M. de Rijke. The University of Amsterdam at INEX-2002. In Fuhr et al. [1], pages 24–28.

[7] S. H. Myaeng, D.-H. Jang, M.-S. Kim, and Z.-C. Zhoo. A flexible model for retrieval of SGML documents. In *SIGIR 1998*, pages 138–145, 1998.

[8] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *SIGIR 1993*, pages 49–58, 1993.

[9] R. Wilkinson. Effective retrieval of structured documents. In *SIGIR 1994*, pages 311–317, 1994.