ABSTRACT

Information retrieval from XML documents offers an opportunity to go below the document level in search of relevant information, making any element of an XML document a retrievable unit. We consider two dimensions along which we compare this element retrieval task with the traditional document retrieval task. We investigate how different topic representations and language model smoothing approaches affect the performance of the two tasks. We evaluate our ideas against the INEX 2002 XML retrieval test-suite.

Categories and Subject Descriptors
H.3 [Information Storage and Retrieval]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms
Experimentation

Keywords
XML retrieval, language models, smoothing, topic representation

1. INTRODUCTION

XML documents differ from plain text documents. The latter contain only plain text and they themselves are the natural unit of retrieval. XML documents, in contrast, are divided into a hierarchy of text objects, each of which could in principle be returned in response to a query. It is thus tempting to try to go below the document level and focus on retrieving document fragments that provide exhaustive yet concise answers to the users’ information need.

In this paper we report on ongoing work aimed at comparing two XML retrieval tasks: XML document retrieval (return whole XML documents in response to an information need) and XML element retrieval (return focused elements only). Thus, our main question in this paper is the following:

Aim 1 How is XML element retrieval different from XML document retrieval?

In our comparison we focus on two aspects, closely related to the fact that XML elements and XML documents may vary widely in length: topic field selection and language model smoothing.

It is known that the use of additional topic fields from a test collection may affect retrieval effectiveness (see the related work section below). In particular, for adhoc retrieval the use of additional (longer) topic fields tends to increase performance, whereas for retrieval tasks that aim to retrieve sentences or other very small units, the use of longer topic representations tends to hurt performance. In principle, XML elements can range in length from very short (e.g., a single word) to the whole document. This, then, gives rise to the second of our main aims in this paper:

Aim 2 How does topic field selection affect the two tasks?

In recent years, language modeling approaches to information retrieval have attracted a lot of attention [20, 10, 16]. Language models are attractive because of their foundations in statistical theory, the great deal of complementary work on language modeling in speech recognition and natural language processing, and the fact that very simple language modeling retrieval methods have performed quite well empirically. The basic idea of these approaches is to estimate a language model for each document, and then rank documents by the likelihood of the query according to the estimated language model. Since document language models may suffer from inaccuracy due to data sparseness, a core issue in language modeling is smoothing. Smoothing refers to adjusting the maximum likelihood estimator for the document language model by, for example, combining it with a collection language model. The retrieval performance is generally sensitive to the smoothing parameters. Earlier studies in adhoc retrieval have found that for shorter queries the Jelinek-Mercer method works well with less smoothing (i.e., more weight is given to the document language model), while long queries require more smoothing (i.e., more weight is given to the collection language model) [23]. How do these findings carry over to the setting of XML document or element retrieval, and how are they influenced by the choice of topic fields? More generally, we have our third aim:

Aim 3 How does smoothing affect the two XML retrieval tasks?

To answer the questions raised above, we use the INEX test collection. The INitiative for the Evaluation of XML retrieval (INEX) was launched in 2002 to assess the effectiveness of retrieval methods for XML document and element retrieval [11]. The collection contains two kinds of topics. Content-only topics (CO) are traditional IR topics written in natural language. Content-and-structure
Smoothing is also task dependent. Language models for adhoc retrieval, and other tasks that are assessed in terms of mean average precision scores, tend to perform better if much smoothing is done \[13, 10\]. On the other hand, language models for high precision tasks such as web retrieval tasks seem to perform better if very little smoothing is applied \[14\]. With XML element retrieval we seem to be in a mixed situation: while it is assessed in terms of mean average precision, it can be thought of as a high precision retrieval task.

Much attention has been given to passage retrieval in the information retrieval community. The work has mainly focused on the use of passages to improve document retrieval \[21, 12, 15\]. Assessments have traditionally been performed on the document level, but not at the level of passages. Hence the evaluation of the passage retrieval is actually done at the document level. In \[18\] this approach has been adopted to XML retrieval; scores for individual XML elements are used to improve document retrieval in an SGML collection. These tasks are different from the XML element retrieval task discussed in this paper: the INEX collection provides assessments done directly on the element level. Hence the retrieval of XML elements proper is evaluated directly.

### 3. EXPERIMENTAL SETUP

We evaluate our ideas against the INEX 2002 XML information retrieval test-suite \[7\]. The INEX 2002 collection contains over 12,000 articles (consisting of nearly 7,000,000 elements) from 21 IEEE Computer Society journals, with layout marked up with XML tags. The collection contains around 170 different tag-names, representing units as diverse as complete articles \texttt{<article>}, sections \texttt{<sec>}, paragraphs \texttt{<p>} and italics font \texttt{<it>}.

To evaluate the two XML retrieval tasks, document and element retrieval, we need two types of indexes.

![Simplified figure of how an XML document is split up into overlapping indexing units.](image)

**Element index** Here, each element of an XML document is an indexing unit. For each element, all text nested within the element (including its descendants) is indexed (See Figure 1). This results in an overlapping element index, since the text nested at depth \( n \) is indexed as part of \( n \) different units.

**Document index** A fraction of the element index where only elements with a location path of depth 1 are considered (Such as the element with path \texttt{/article[1]} in Figure 1).
No stemming was applied to the indexes but we did lower-casing and stop-words were removed.

In our experiments we used the 23 CO topics that come with the INEX collection. INEX topics are divided into four fields: title, a short 2-3 word version of the topic statement; description, a one sentence definition of an information need; narrative, an explanation of the topic statement in more detail; and keywords, synonyms or terms that are broader/narrower than those listed in the title and description [6] (p.179). We used these fields, independently or in combinations, to create 5 different topic representations.

**T** The terms from the title field.

**TD** The terms from the title and description fields.

**TDN** The terms from the title, description and narrative fields.

**TDK** The terms from the title, description and keywords fields.

**TDNK** The terms from the title, description, narrative and keywords fields.

As with the collection, we did not stem the topics but lower-cased and removed stop-words.

At INEX 2002, relevance was assessed at the element level. Elements were assessed on a two dimensional graded relevance scale, one for topic relevance and another for element coverage [9] (p.184). From the official relevance assessments we derived two assessment sets, one for each of the tasks we want to evaluate.

**Document retrieval task** For evaluating the document retrieval we considered a document relevant if it contains an element judged highly relevant with exact coverage.

**Element retrieval task** For evaluating the element retrieval task we considered an element relevant if it was judged highly relevant with exact coverage.

We used version 1.8 of the INEX 2002 relevance assessments. Evaluation was done using the trec_eval program. Our evaluation method for element retrieval is similar to the strict evaluation used at INEX 2002 [6].

All our retrieval runs used a multinomial language model, with single length prior and Jelinek-Mercer smoothing [10]. Our scoring formula for an indexing unit \( d \) is thus

\[
s(d) = \log \left( \sum t f(t,d) \right) + \sum_{i=1}^{n} \log \left( 1 + \frac{\lambda \cdot t f(t,d) \cdot (\sum d f(t))}{(1-\lambda) \cdot d f(t) \cdot (\sum t f(t,d))} \right)
\]

where \( t f(t,d) \) is the frequency of term \( t \) in document \( d \) and \( d f(t) \) is the count of document in which term \( t \) occurs. We experimented with a range of \( \lambda \) in the interval \([0.05, 0.95]\). In this paper we devote special attention to two values of the smoothing parameter used frequently in the literature. First, the default value of \( \lambda \) in adhoc retrieval: 0.15 [13]. Second, for high precision tasks such as web retrieval a high value of \( \lambda \) is normally used, such as 0.90 [14].

4. EXPERIMENTAL RESULTS

In this section we report on the results of our experiments for the two XML retrieval tasks. In Section 4.1 we look at the XML document retrieval task, in Section 4.2 we look at the XML element retrieval task and in Section 4.3 we compare the results for the two tasks. Since the combination of title and description fields is the most common topic representation in adhoc retrieval tasks, we use it as the baseline in our numeric comparisons. For determining whether a difference between retrieval runs is statistically significant, we use the bootstrapping method [5, 22]. We take 100,000 re-samples and look for improvement at significance levels 0.95 (*); 0.99 (**); and 0.999 (***)

### 4.1 XML document retrieval task

Figure 2 shows MAP scores of document runs for different values of the smoothing parameter \( \lambda \). The first thing to notice is that the inclusion of keywords in the topic representation has the biggest positive impact on scoring (see TDN vs TDK, and TD vs TDK). In Table 1 we compare the MAP scores of the topic representations for two values of the smoothing parameter \( \lambda \). As the table shows, the runs using the keywords field are the only ones to improve significantly over the baseline. We can also see that the topic representations containing the narrative field are the most sensitive to smoothing. This is not surprising since there may be various terms in the narrative that are not informative for the particular topic at hand. We see that the title-only queries do contain only good retrieval terms for the topic at hand. The T and TDK topic representations are the most stable over the range of values for the smoothing parameter. As before, the informativeness of all terms in the title and keywords fields is the most plausible explanation.

### 4.2 XML element retrieval task

Figure 3 shows MAP scores of element runs for different values of the smoothing parameter \( \lambda \). For the element retrieval task, increased smoothing seems to hurt all topic representations, except for the title-only run. We again see that the queries including the
Finding the relevant elements seems to be a genuine needle-in-a-haystack problem.

There are some similarities between the two tasks with respect to the impact of topic field selection. Adding terms from the keywords field leads to the biggest improvements. Longer topic representations will generally improve recall, but at the same time may hurt precision. Since the keyword field contains only terms that are informative for the topic at hand, we may expect little loss of precision. For terms in the other fields this need not be the case: both the description and narrative may contain terms that are not specific for the topic at hand. This is illustrated by the plots in Figure 4. The differences between the two tasks with respect to topic field selection largely depend on the used smoothing parameter.

The two tasks respond totally different to changes in the smoothing parameter \( \lambda \). Much smoothing, i.e., a low value for \( \lambda \), is the appropriate choice for the document retrieval task. This is in line with other adhoc retrieval experiments [23, 13, 10]. Little smoothing, i.e., a high value for \( \lambda \), is the appropriate choice for the element retrieval task. We believe there are two factors working together toward providing the highest scoring for the element retrieval task. One is the high initial precision of the \( \lambda = 0.9 \) run; see Figure 4. Since it is extremely difficult to get high recall for this task, early precision is very important. Another factor is the size of the retrieved element. There is a serious variance in the length of elements in the collection and the average element length is low. However, assessors seem to have a strong bias toward larger elements [12]. Since we approach coordination level matching as \( \lambda \) approaches 1 (10 Appendix B), combining the long TDNK topic with little smoothing gives us a retrieval run that prefers elements which contain all the query terms, independent of whether they are informative or not. This may result in retrieval that has a similar bias toward larger elements as is present in the assessments.

Figure 5(c) shows the average length of retrieved elements for each of the values of the smoothing parameter. We can see a clear connection between the smoothing parameter and the average length of retrieved elements. For the longer topics (TD, TDN, TDK and TDNK), a higher value for \( \lambda \) causes larger elements to be retrieved on average. The opposite effect for the title only run is probably due to the fact that the length prior dominates in the scoring formula, as there are so few query terms. If we restrict our attention to the relevant elements retrieved we see the same tendency, but on a smaller scale (Figure 5(d)). Corresponding graphs for the document runs are shown in Figure 5(a) and (b). For comparison with the actual collection and assessment statistics see Table 3.

### Table 3: The count, average length, minimum length and maximum length of the set of documents, set of relevant documents, set of elements and set of relevant elements

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Avg. len</th>
<th>Min len</th>
<th>Max. len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>12,107</td>
<td>3,234</td>
<td>24</td>
<td>21,335</td>
</tr>
<tr>
<td>Relevant</td>
<td>627</td>
<td>3,902</td>
<td>95</td>
<td>18,109</td>
</tr>
<tr>
<td>Element</td>
<td>6,779,686</td>
<td>29</td>
<td>1</td>
<td>21,335</td>
</tr>
<tr>
<td>Relevant</td>
<td>1,394</td>
<td>1,484</td>
<td>1</td>
<td>18,109</td>
</tr>
</tbody>
</table>

### Figure 3: Mean average precision for element runs using different values for the smoothing parameter \( \lambda \).

### Table 2: MAP of element-runs using different query formats and different smoothing parameters.

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>MAP</th>
<th>% change</th>
<th>MAP</th>
<th>% change</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD (baseline)</td>
<td>0.0396</td>
<td></td>
<td>0.0396</td>
<td></td>
</tr>
<tr>
<td>TDN</td>
<td>0.0624</td>
<td>+59.6%</td>
<td>0.0581</td>
<td>-2.8%</td>
</tr>
<tr>
<td>TDK</td>
<td>0.0326</td>
<td>-16.6%</td>
<td>0.0668</td>
<td>+11.7%</td>
</tr>
<tr>
<td>TDNK</td>
<td>0.0493</td>
<td>+26.1%</td>
<td>0.0857</td>
<td>+43.3%***</td>
</tr>
<tr>
<td>T</td>
<td>0.0481</td>
<td>+23.0%</td>
<td>0.0893</td>
<td>+49.3%***</td>
</tr>
</tbody>
</table>

4.3 Documents vs. Elements

Looking at the results for the document and element retrieval tasks (Figures 2 and 3), there is a striking difference between the performance of XML document retrieval and XML element retrieval. Document retrieval performs much better than element retrieval. There is no surprise here: we can view the XML element retrieval task as a non-trivial extension of the XML document retrieval task. For the XML element retrieval task, given the set of relevant XML documents, we need to dive into each of the documents and retrieve the exact unit that made the document relevant.

We can also look at the ratio between the number of relevant documents and the number of documents in the collection and compare it to the ratio between the number of relevant elements in the collection and the total number of indexed elements in the collection (over all topics). (See Table 3)

\[
\frac{\text{rel.articles}}{\text{articles}} = \frac{627}{12,107} \approx 0.0517
\]

\[
\frac{\text{rel.elements}}{\text{elements}} = \frac{1,394}{6,779,686} \approx 0.000206
\]

5. CONCLUSIONS

In Section 1, we introduced a number of research questions that motivated the experiments on which we reported in this paper. As for the second aim (how does topic field selection affect the two XML retrieval tasks?), we have seen that topic representations including keywords give the best MAP score for both tasks. Those
Figure 4: Precision-Recall curves for the different retrieval tasks, smoothing parameters and query formats.

Figure 5: Average unit length for the different retrieval tasks, smoothing parameters and query formats.
7. REFERENCES


