

Feature- and query-based table of contents generation for XML documents

Zoltán Szlávik, Anastasios Tombros, and Mounia Lalmas

Department of Computer Science,
Queen Mary University of London

Abstract. The availability of a document’s logical structure in XML retrieval allows retrieval systems to return document portions (elements) instead of whole documents. This helps searchers focusing their attention to the relevant content within a document. However, other, e.g. sibling or parent, elements of retrieved elements may also be important as they provide context to the retrieved elements. The use of table of contents (TOC) offers an overview of a document and shows the most important elements and their relations to each other. In this paper, we investigate what searchers think is important in automatic TOC generation. We ask searchers to indicate their preferences for element features (depth, length, relevance) in order to generate TOCs that help them complete information seeking tasks. We investigate what these preferences are, and what are the characteristics of the TOCs generated by searchers’ settings. The results have implications for the design of intelligent TOC generation approaches for XML retrieval.

1 Introduction

As the *eXtensible Markup Language (XML)* is becoming increasingly used, retrieval engines that allow search within collections of XML documents are being developed. In addition to textual information, XML documents provide a markup that allows the representation of the logical structure of XML documents in content-oriented retrieval. The logical units, called elements, are encoded in a tree-like structure by XML tags. The logical structure allows retrieval systems to return document portions that may provide better searcher satisfaction, as the retrieved information will be more focussed.

XML retrieval has received large interest over the last few years, mainly through the INEX initiative [5, 4]. The interactive aspect of XML IR has recently been investigated through the interactive track at INEX (iTrack) [16, 10]. One of the findings of the iTrack was the importance of the hierarchically structured presentation of documents and elements in the result list, and the table of contents of the documents [9, 8]. The table of contents (TOC), in particular, provided context to the retrieved elements. The use of TOCs offered an overview of the document structure and showed the relevant elements and their relations to each other. This was appreciated by many searchers [11].

In the systems used in the iTrack, TOCs had two main limitations: i) they were static, i.e. the same TOCs for a given document were displayed for all searcher queries, and ii) they were manually defined, i.e. before the documents were used in the systems, they had to be analysed and several (types of) elements selected to be included in TOCs, and this selection was not automatic.

These limitations were verified in [13, 14], where it was also investigated what a useful TOC should be like according to searchers of an interactive IR system. It was found that a TOC should reflect which elements are possibly relevant to the searchers' queries, i.e. TOC items of relevant elements are more important to be shown in the TOC than those of non relevant ones. It was also found that the display of a TOC should depend on the length of the elements, e.g. longer sections are more important to include in the TOC. Another finding was that it is important to consider how deeply an element is in the document structure, i.e. for most of the XML documents currently being used in XML retrieval, a one or two-level deep TOC is probably too shallow, while four or five level deep TOCs may sometimes be too deep to help searchers with their information seeking task.

Based on the above, in this work we follow an approach to automatically generate "dynamic" TOCs by considering the characteristics of XML documents and the searcher's query. The TOC generation algorithm is inspired by early text summarisation (sentence extraction) systems (e.g. [3, 17]). We consider features of elements such as their length, depth and relevance and combine these in order to determine whether an element should have a reference in the TOC. Our approach allows us to create a TOC for any XML document, and the TOC will be biased towards the searcher's query. In this paper, we investigate how this approach can be used in a user-oriented system, and what features are more important in TOC generation than others according to searchers. We also examine what the size of a TOC should be so that searchers can find the relevant information effectively.

To answer these questions, we created a system. Searchers were asked to consider information seeking tasks and to find relevant information within XML documents. They were allowed to adjust the importance of element features (i.e. length, depth, relevance). By adjusting these, searchers were able to alter the characteristics of the current TOC and the aim was to generate an appropriate TOC for documents in the context of the current query. We recorded these searcher preferences along with questionnaires and analysed them.

The system and the methodology followed are described in Section 2, followed by the detailed description of the TOC generation algorithm (Section 3). Section 4 presents and analyses the results, and we close with discussion and conclusions in Section 5.

2 Experimental setup

We followed a methodology that is based on simulated work task situations [1]. Searchers were given work task descriptions so they could search for relevant

information. During their search, they were asked to identify the best TOC of the current document with respect to the current work task by adjusting the importance of element features.

Searchers were asked to read the work task descriptions, proceed to the document view of as many documents as they wish, and adjust their preferences for three element features (length, relevance, depth) and a threshold by moving sliders on the interface. By adjusting the sliders, searchers were able to alter the characteristics of the current TOC. When they felt that the displayed TOC was helpful enough to assist them in finding relevant information, they could move on to the next document or topic.

Participants were asked to fill in questionnaires before and after the experiment, they were given detailed introduction to what their task was and no time restrictions to finish the experiment were imposed. After filling in the entry questionnaire and having read the introduction, searchers were presented with the first topic description and links to the corresponding relevant documents (Figure 1). The order of the displayed topics was randomised to avoid any effect caused by one particular order.

Let us suppose that you work as teaching assistant for operating systems. This inspired your interest in microkernel operating systems, and you are looking for documents that talk about operating systems with microkernel. Documents or document segments talking only about operating systems or microkernel are not of your interest. (A Microkernel is a highly modular collection of powerful OS-neutral abstractions, upon which can be built operating system servers.)

The following documents contain possibly relevant information. Use the sliders to get the best table of contents and explore the document through clicking on the items in the left window.
[CLICK](#), [CLICK](#) or [CLICK](#).

Fig. 1. A topic description and links to its documents.

After choosing a document, the document view was shown (Figure 2). This consisted of four main parts:

- on the left, sliders associated with element features were shown. These needed to be adjusted by searchers to generate TOCs;
- on the bottom left, the generated TOC was shown;
- on the right hand side, the contents of the document or element was presented. These changed when searchers clicked on an item in the TOC;
- on the top left corner, links to the topic description, next topic and final page were displayed.

By clicking on the ‘finish’ link, the exit questionnaire was shown, where we recorded information about the searchers’ perception of the system and TOC generation, e.g. the strategies searchers used when adjusting the sliders on the main screen.

Documents from two XML document collections - IEEE and Wikipedia [2] - were used¹. Ten topics that were available for these collections were selected

¹ These are the test collections used in INEX.

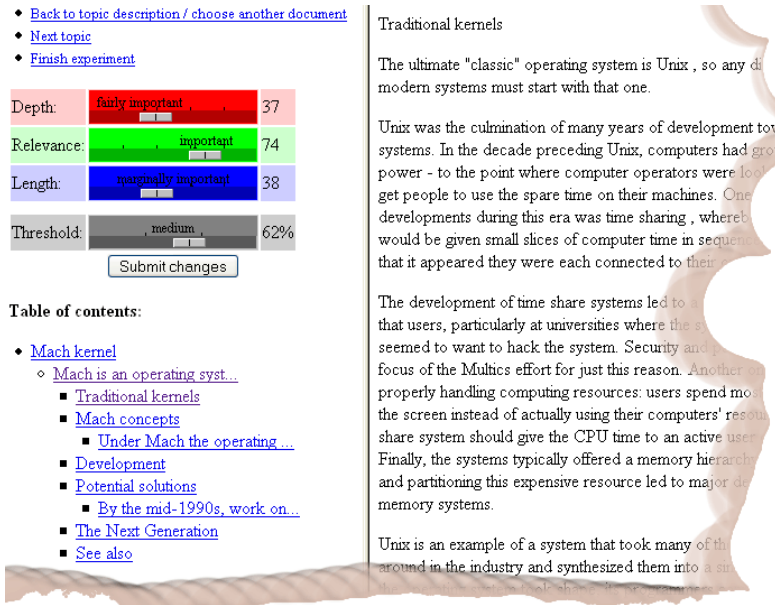


Fig. 2. Screen shot of the main screen with sliders, TOC and element display.

by random and converted into work task descriptions. This gave us five work tasks for each collection. For each topic, three to five relevant documents were selected. Relevant documents were obtained by formulating queries from the topic descriptions, running these queries in the TopX system² [15] and selecting the most relevant documents from the result list. The retrieval status values of elements were saved to be used in the TOC generation (Section 3). Documents of various sizes were selected, the shortest document contained 334 bytes of text while the longest 49KB. We made an effort to select documents from both collections for a particular topic; there were only two topics where relevant documents were found in both collections.

As a result of the topic and document selection, 33 documents and 10 topics were selected from the two collections. These provide an appropriate level of diversity thus ensuring that our results are not biased by topic and document selection.

3 TOC generation

In this section, the algorithm to generate TOCs is described. The algorithm aims to identify among a set of XML elements, those that will form the TOC. It makes use of an element score that is calculated for every element in consideration. If the score of an element is higher than a certain threshold value (described below), the element is considered as a TOC element. Ancestors of such elements, i.e.

² TopX is the system used in INEX for the collection exploration phase.

elements higher in the XML hierarchy, are also used to place the TOC elements into context. For example, a section reference in a TOC without the chapter it is in would be just ‘floating’ in the TOC. These selected elements’ titles are displayed as TOC items. If no title is available, the first 25 characters of the text are shown. The ancestor-descendant relation of elements is reflected, as in a standard TOC, by indentation (Figure 2).

The score of an element is computed using three features of the element: its depth, length and relevance. The first two are element-based features, whereas the third is query-based. These features have been shown to be important characteristics in various XML retrieval tasks [4], although other features can be also taken into account.

Depth score Each element receives a depth score between zero and one, based on where it is in the structure of the document. In our case, an *article* element is always at depth level one (i.e. it is the root element in the tree structure). Descendants of a depth level one element are at depth level two (e.g. /article[1]/section[4]), etc. [13] found that elements at depth level three of a TOC were the most important to access the relevant content whereas the adjacent levels (two and four) were found less important, and so on. Sigurbjörnsson ([12], Ch. 8) also found, using the IEEE collection, that searchers mostly visited level 2-3 elements while looking for relevant information. Hammer-Aebi et al.[6] confirmed that searchers found the highest number of relevant elements at levels two to four. Since the latter work used a different XML collection (Lonely Planet [18]), the importance of these levels seems to be general for XML collections. To reflect these findings, the following scoring function was used to calculate an element’s depth score (Equation 1):

$$S_{depth}(e) = \begin{cases} 1 & \text{if } depth(e) = 3, \\ 0.66 & \text{if } depth(e) \in \{2, 4\}, \\ 0.33 & \text{if } depth(e) \in \{1, 5\}, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $S_{depth}(e)$ denotes the depth score of element e .

Length score Each element receives a length score, which is also normalised to one. The normalisation is done on a logarithmic scale [7], where the longest element of the document, i.e. the root element, receives the maximum score of one (Equation 2):

$$S_{length}(e) = \frac{\log(TextLength(e))}{\log(TextLength(root))} \quad (2)$$

where $S_{length}(e)$ is the length score of element e , *root* is the root element of the document structure and *TextLength* denotes the number of characters of the element.

Relevance score A score between zero and one is used to reflect how relevant an element is to the current search topic. The scores were those given by the

search engine used in INEX for the collection exploration [15] (i.e. a normalised retrieval status value). The RSVs are obtained in the document selection phase (described in Section 2).

Feature weighting The scores of the above three features are combined so that we can emphasize the importance of a feature over another. This is done by using a weighted linear combination of the feature scores (Equation 3). Searchers are allowed to set the weights themselves. This allows us to investigate what searchers find important in TOC generation, and also, to determine what weights should be used to generate TOCs based on such features.

$$S(e) = \sum_{f \in F} W(f) \cdot S_f(e) \quad (3)$$

where $S(e)$ denotes the overall score of element e , F is the set of the three features, $W(f)$ is the weight of feature f and $S_f(e)$ denotes the score that is given to element e based on its feature f .

Threshold To determine the lowest score an element must achieve in order to be included in the TOC, we use a threshold value. As well as the feature weights described above, this value is set by the searchers of the system. This allows us to determine what the desirable size of a TOC should be. In our algorithm, if the threshold is set to 100% only elements with the maximum depth, relevance and length scores will be included in the TOC (i.e. $\sum S_f(e) = 1$). If the threshold is set to zero, every element with greater than zero score will be in the TOC. We use a default value of 50%.

4 Results and analysis

In this section we describe and analyse the results of our experiment. We start with results regarding participation (Section 4.1) followed by a detailed analysis of the collected data. We investigate what slider values were set and what were the main characteristics of the generated TOCs when searchers were finished with a document (Section 4.2), whether there were differences in terms of preferences among searchers (Section 4.3) and whether differences could be found among documents of the two collections and among the topics used (Section 4.4).

4.1 Participation and questionnaires

50 searchers, mainly with a computer science background, responded to our call for participation. To record information only from searchers who spent a significant amount of time participating in the experiment (thus providing usable data), we filtered the log data so that only those involving at least three different documents were kept and analysed. As a result, 31 searcher sessions were analysed where participants used an average of 7.74 (out of the the maximum ten) topics and 15.58 (out of the maximum 33) documents. This gave us 483

different settings. The slider values after a searcher finished with a document were considered as a single setting.

In the post experiment questionnaires, searchers indicated how easy they had found to learn and understand the usage of the system. On a seven point scale, they indicated an average of 4.85 and 4.64 respectively, where 1 meant ‘was not easy at all’ and 7 meant ‘it was extremely easy’. This shows that the use of the system was not an issue in our study. In the same scale, searchers indicated an average value of 3.71 for the question ‘How easy was it to set up the sliders to get a good table of contents?’ which shows that getting the desired TOC took some effort. In their comments, searchers indicated various understandings of the sliders, but most of them had a strategy how to set them up and, according to the logs, all of the searchers used the sliders extensively.

4.2 Sliders and TOC characteristics

On average, the depth, length and threshold slider values were around the default value (50), while the relevance slider was in a higher average position (75.79). The standard deviation of the depth, relevance and length sliders was the same (25), and the threshold slider had a lower (15) standard deviation. This indicates that searchers found the relevance feature to select TOC elements more important than the other two. Indeed, the importance of relevance to access the relevant content is the most intuitive of the three features, but results indicate that other features also needed to be considered if one wanted to place the relevant elements into context, i.e. to show related contents.

The average threshold slider value of 50% with standard deviation of 15 shows that the agreement among searchers regarding the threshold was higher than those of the three features. In addition, we did not find a tendency of different slider values for documents of various sizes, thus the average values seem to be appropriate for XML documents of any size in our setting.

An average TOC consisted of 19 items which is, on average, 8.16% of all the elements in the original XML document. We examined how the size of the TOCs is related to the size of the documents. Since there was high correlation between the length of the documents and the number of elements they contained, we use these two measures as ‘size’ interchangeably. We found that the more elements a document contained, the smaller the proportion of the number of TOC elements to the number of document elements was (Figure 3). This means that a long document does not necessarily need a long TOC. This is because a too long TOC does not help searchers gain an overview of the contents of the document because they need to gain an overview of the contents of the TOC first. This clearly indicates that a TOC generation algorithm has to perform particularly well for longer documents, as the TOC algorithm is much more selective in element items in longer documents.

To also examine the distribution of the length of elements in the TOCs and in the documents, we created five size categories based on the length of text in elements (Figure 4.a). We found that the distribution of element lengths of documents (light gray in Figure 4.a) follows a bell curve where most of the elements

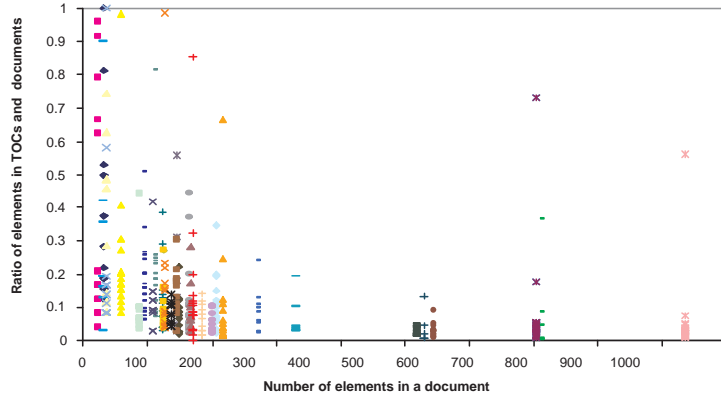


Fig. 3. Number of elements in the documents and the TOC elements' ratio to it.

cover 10-100 characters of text, i.e. there are many elements with the size of a short sentence. Elements longer than 10KB are very rare in our documents, such elements can be e.g., the root (i.e. *article*) element of a long document.

The distribution of the length of elements included in the TOCs (dark gray in Figure 4.a) is also a bell curve having its peak at the category of 100-1000 character long elements. Slightly shorter and longer elements are less frequent in the TOCs while very long and very short elements do not occur frequently in the TOCs. This shows that the length distribution of the elements in the documents and TOCs is of the same nature, only the parameters are different. The implication of this is that a TOC has to be constructed to reflect the original structure of the document.

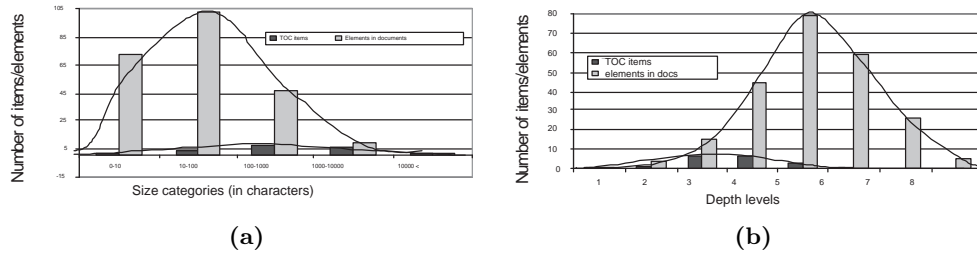


Fig. 4. TOC items and document elements at various size categories (a) and depth levels (b).

We also examined the distribution of depth levels in the TOCs and document elements. (Figure 4.b). We considered eight depth levels because deeper than the eighth level there were very few, if any, elements in a document, and none of the displayed TOCs were deeper than seven levels. The distribution of element

depths of documents follows, similar to the length distribution, a bell curve: most of the elements are between the fourth and sixth level, there is only one element at level one (which is the root element of a document's tree structure) and very few elements are below the seventh level. The depth distribution of the TOC elements is still a bell curve with the peak at level three and four, which is consistent with a finding in [14]. In other words, the depth distribution found in the TOCs and documents are the same, only the parameters are different.

Considering both the length and depth distributions, the TOCs reflect the main characteristics of the documents. The TOCs can therefore be viewed as extracts of the document structure, so not only is the algorithm based on summarisation but the output of it is also a summary (i.e. that of the document structure).

4.3 Searchers

Searchers used different slider setting strategies to generate the best TOC. Nonetheless, the majority of them did set the relevance slider high. This high view of relevance was confirmed in the post experiment questionnaires. The questionnaires show that the relevance slider was usually set first, which also shows its importance. Some of the searchers set the length, depth and relevance sliders first and changed the threshold slider slightly document after document. In this process, setting the most appropriate threshold was found difficult for most searchers, especially because they had an 'ideal' TOC in their minds that was to be reached by adjusting the sliders. According to the questionnaires, an ideal TOC contained those, and only those, elements that had been found useful when searchers had been experimenting with the settings for the TOC of the current document. Although not all of them followed these strategies, the TOCs generated did not differ very much for different searchers: apart from a few searchers, the TOCs were not longer than twenty items, and searchers also seemed to agree in terms of length categories and depth levels (Figure 5). Based on the above, it seems that the size of a TOC does not significantly depend on individuals. Indeed, a TOC should rarely contain more element references than some fixed value, in our case twenty. Our study shows that it is important to select the best (maximum) twenty elements, appropriately.

4.4 Collections and topics

We investigated the possible differences between TOCs generated for documents of the two collections. There were slightly more documents used from the Wikipedia collection (20) than from the IEEE (13), and therefore 158 slider values were examined for the IEEE collection and 325 for Wikipedia documents. We did not find differences in the average slider values with respect to the two collections; settings for one collection also seemed satisfactory for the other.

The generated TOCs' characteristics were not significantly different either. Although the structure of the two collections' documents are similar to each

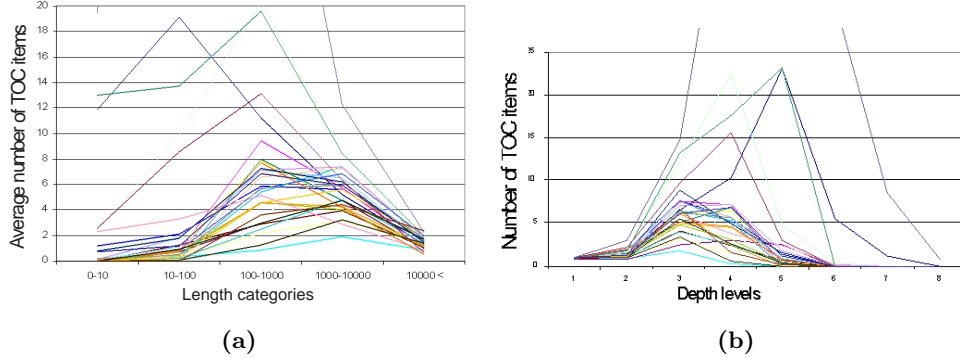


Fig. 5. Number of TOC items at (a) various length categories and (b) depth levels for searchers.

other, we do not expect that these results will be different for other XML documents. This is because documents of XML collections must have exactly one root element, which has several child elements etc., so we expect that the algorithm described in this paper can be used for any XML documents.

We also investigated whether there were differences in settings and TOCs among the ten topics. The distribution of the slider values did not reveal great differences: the relevance values were always higher than that of other sliders and the depth-length-threshold triplet’s order was slightly different for some topics but these were always closely around the default value of 50. This shows that for the ten topics we used, searchers did not need to use very different settings to obtain an acceptable TOC. However, the number of topics does not guarantee that we covered a wide enough range of topics and task types. To compare the results for different task types (e.g. finding background information vs. answering a question), more topics are needed from different task types.

The number of TOC elements were between 14 and 27 for 8 of the 10 topics, which is around 9% of the number of elements in the documents. There was one topic (*w2*) for which the average number of TOC elements was as low as 7, and for another topic (*w5*), this number was 32. These two extreme values are closely related to the number of elements the topics’ documents had, i.e. documents for (*w2*) were very short while documents for (*w5*) contained the longest document. This shows that longer documents may require longer TOCs, but since the ratio of the number of elements in TOCs and documents is different for shorter and longer documents (see Figure 3), size differences in the TOCs should not be linearly proportional to document sizes.

5 Discussion and conclusions

We have studied searchers’ preferences of element features in automatic TOC generation for XML retrieval. The features and searchers’ preferences (as weights) can be used to select those elements that will form the TOC. We have considered

three features: depth, length of the elements and their relevance to the current query. We have conducted a user study to investigate which of these features are important in TOC generation, and what are the characteristics of TOCs that were generated by searchers' feature preferences.

Our algorithm offers a mapping from the set of elements in the document to that of the TOC, where the most important elements of the documents are selected as TOC elements and the distribution of length and depth of elements remains very similar. The result of the mapping is an extract of the document structure, which, by organising it into a table of contents, can be used to help searchers find relevant content inside an XML document.

In this study, we found that a TOC generation algorithm like the one introduced in this paper has to consider the relevance of an element, i.e. it should be highly query-biased. The other two element features used, i.e. length and depth of an element, should also be considered and the weight of these two should be lower than that of the relevance feature. It is also understood that TOCs should not be large in size, i.e. longer documents should still have a relatively small TOC. This also shows that automatic TOC generation has to be more carefully designed when dealing with longer documents. To ensure better results, the TOC algorithm can be extended to include other element features such as e.g. tag names, titles of elements.

Our data also suggest that the size of a TOC does not significantly depend on individual searchers. The selection of the most important elements is much more important, and here may be worth considering searchers' individual preferences. We suggest that if a TOC algorithm selects more than a certain number (e.g. 20) of TOC elements (e.g. 30), the top scored elements (i.e. top 20) should be kept regardless of what threshold value the algorithm uses. If the number of TOC elements is lower than this number (e.g. 10), these elements should be used to construct the TOC.

During the analysis of the collected data we learned that searchers had several strategies to get the best TOCs. Although not every one of the searchers understood the concepts of the features completely, they actively used the sliders and created TOCs that, according to questionnaires, were suitably good to access the relevant content more easily.

To conclude, we have developed an algorithm to automatically generate tables of contents for XML documents. The algorithm uses features of elements to select those that will form the TOC. Different TOCs are generated for different queries which we think may help searchers access the relevant content more quickly. In our experiment, we investigated which features are important in TOC generation, and what are the characteristics of TOCs that are generated by searchers' feature preferences.

The work presented here is part of a wider work that aims at developing and evaluating methods that summarise the content and the structure of documents for structured document retrieval. We believe that the effective combination of the two types of summarisation can help searchers focus on only the useful

contents of the documents, decrease the time searchers spend on finding relevant elements, and thus, increase user satisfaction.

6 Acknowledgments

The authors wish to acknowledge the participants of the study. This work was partly funded by the Nuffield Foundation (grant NAL/01081/G) and the DELOS, Network of Excellence in Digital Libraries.

References

1. P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003.
2. L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40(1):64–69, 2006.
3. H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2):264–285, 1969.
4. N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Proceedings of INEX 2005*, volume 3977, 2006.
5. N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Proceedings of INEX 2004*, volume 3493, 2005.
6. B. Hammer-Aebi, K. W. Christensen, H. Lund, and B. Larsen. Users, structured documents and overlap: interactive searching of elements and the influence of context on search behaviour. In *Proceedings of IiX*, pages 46–55, 2006.
7. J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In *Proceedings of ACM SIGIR*, pages 80–87, 2004.
8. J. Kamps and B. Sigurbjörnsson. What do users think of an XML element retrieval system? In Fuhr et al. [4], pages 411–421.
9. H. Kim and H. Son. Users interaction with the hierarchically structured presentation in XML document retrieval. In Fuhr et al. [4], pages 422–431.
10. B. Larsen, S. Malik, and A. Tombros. The interactive track at INEX 2005. In Fuhr et al. [4], pages 398–410.
11. S., C.-P. Klas, N. Fuhr, B. Larsen, and A. Tombros. Designing a user interface for interactive retrieval of structured documents - lessons learned from the INEX interactive track. In *Proceedings of ECDL 2006*, pages 291–302, 2006.
12. B. Sigurbjörnsson. *Focused Information Access using XML Element Retrieval*. PhD thesis, Faculty of Science, University of Amsterdam, 2006.
13. Z. Szlávik, A. Tombros, and M. Lalmas. Investigating the use of summarisation for interactive XML retrieval. In F. Crestani and G. Pasi, editors, *Proceedings of ACM SAC-IARS'06*, pages 1068–1072, 2006.
14. Z. Szlávik, A. Tombros, and M. Lalmas. The use of summaries in XML retrieval. In *Proceedings of ECDL 2006*, pages 75–86, 2006.
15. M. Theobald, R. Schenkel, and G. Weikum. An efficient and versatile query engine for TopX search. In *Proceedings of VLDB*, pages 625–636, 2005.
16. A. Tombros, B. Larsen, and S. Malik. The interactive track at INEX 2004. In Fuhr et al. [5], pages 422–435.
17. A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of ACM SIGIR*, pages 2–10, 1998.
18. R. van Zwol, G. Kazai, and M. Lalmas. INEX 2005 multimedia track. In Fuhr et al. [4], pages 497–510.