Investigating the Use of Summarisation for Interactive XML Retrieval

Zoltán Szlávik
Queen Mary University of London
London, E1 4NS
UK
zolley@dcs.qmul.ac.uk

Anastasios Tombros
Queen Mary University of London
London, E1 4NS
UK
tassos@dcs.qmul.ac.uk

Mounia Lalmas
Queen Mary University of London
London, E1 4NS
UK
mounia@dcs.qmul.ac.uk

ABSTRACT

As the number of components in XML documents is much larger than that of 'flat' documents, we believe it is essential to provide users of XML information retrieval systems with overviews of the content of retrieved elements. In this paper, we investigate the use of summarisation in XML retrieval as a means of helping users in their searching process.

1. INTRODUCTION

As the *eXtensible Markup Language (XML)* is becoming increasingly widespread, retrieval engines that allow search within collections of XML documents are being developed. XML documents contain not only textual information, like in 'flat' documents, but also information about the logical structure of the documents. The logical structure is a tree-like structure encoded by XML tags. For example, an article can be seen as corresponding to the root of the tree, and sections, subsections and paragraphs can be arranged in branches and leaves of the tree. The logical units, called elements, provide document portions that may be better to retrieve than the whole XML document itself, i.e., some elements can themselves be answers to an information need while the rest of the document may contain non-, or partially, relevant information. Thus, in XML retrieval, document components, rather than whole documents, are retrieved. This content-based retrieval of XML documents has received interest over the last few years, mainly through the INEX initiative [4].

As the number of XML components is typically large (much larger than that of documents), we believe it is essential to provide users of XML information retrieval systems with overviews of the contents of the retrieved elements. One approach is to use summarisation, which has been shown useful in interactive information retrieval (IIR) [6, 5, 10].

In this paper, we investigate the use of summarisation in XML retrieval in an interactive environment. In standard IIR, a summary is usually associated with each document returned by the retrieval system; in interactive XML retrieval, a summary can be associated with each document component returned by the XML retrieval system. Because of the nature of XML documents, users can, in addition to accessing any retrieved element, browse within the XML document containing that element. One method to allow browsing XML documents is to display the logical structure of the document containing the retrieved elements. This has the additional benefit of providing (sometimes necessary) context to users when reading a component. Therefore, summaries can also be associated with the other elements forming the document, in addition to the returned components themselves.

The aims of our investigation are twofold: 1) regarding summarisation, we examine whether summarisation is useful when browsing within XML documents; 2) regarding structural information and summarisation, we want to know what structural levels should summaries be applied to, and how closely the structural display and the use of summaries are related to each other in an interactive search process. To answer the questions above, an interactive information retrieval system was developed and examined using human searchers.

The paper is organised as follows. In Section 2, we describe the experimental system that was used and, in Section 3, the experimental design. We show the results in Section 4, followed by discussion. We finish with future work.

2. EXPERIMENTAL SYSTEM

In this section we describe the system that was used in our study: the user interface with XML specific features, the summarisation method and the XML search engine.

2.1 User Interface

The user interface is a web based system which passes the query to the retrieval module, processes and displays the retrieved result list and shows the result elements. The system allows users to enter a search query and start the retrieval process by clicking on the search button. The result list display is similar to standard web search interfaces to minimise searchers' frustration which may be caused by learning how to use a new system. For each result element, the following are shown: rank, retrieval score, query-biased summary, title and path of the XML document that contains the result

SAC'06 April 23-27, 2006, Dijon, France

element, size of the element, a mini map that shows the path of the element within the XML document and gives an idea about how deep the element is nested in the structure, and a link to display the result element. The result page also includes the possibility of running a new query, and shows the number of results, retrieval time and the query terms. Query terms in the titles and summaries are highlighted using a yellow background. Once searchers follow the link to the element, the element is displayed in a new window (Figure 1). The frame on the right shows the content of the target element with query words highlighted. On the left, the structural view of the whole document is displayed, where the position of the currently shown element is also highlighted.

The structural display is based on the XML structure of the whole document, i.e. the root element is shown at the top level, while the descendants are displayed at lower levels (indented, with bullets). Each *structural item* is also a hyperlink that will show the corresponding XML element on the right window when clicked. As an XML document may contain element types that are for formatting purposes only (e.g. it corresponds to italic), selection of element types to be displayed in the structural view is necessary. Based on the analysis of the document corpus and the relevance assessments on this collection by INEX participants, 9 element types were selected for structural display, including article, abstract, section and paragraph types. These correspond to the types that were usually assessed highly relevant for a number of topics.

The label of an item in the hierarchical structure is the title of the corresponding XML element when such information is available. If no title is available, the name of the element type is displayed with its sequential number (e.g. Paragraph 2, Section 1, etc.).

For this user study, four levels of structural items were displayed; the number of levels could be changed by searchers.

For each structural item shown in the hierarchical structure on the left, automatic summaries are generated for that particular item. The algorithm for creating summaries is the same as the one used in the result list display and is described in the next section. Summaries are being displayed as 'tool tips' when the mouse pointer is over a structural item. The aim of summary display is to reduce the number of searchers' clicks and let searchers find relevant document portions more quickly. Query terms in the summaries are also highlighted.

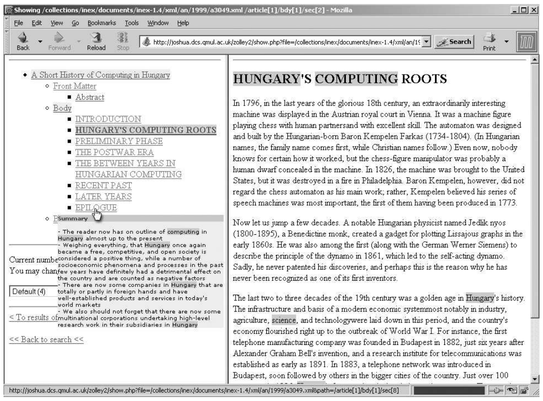
2.2 Summarisation

Since we investigate whether summarisation is useful in interactive XML retrieval and there has not been much research on how to generate summaries from nested XML elements, we implemented a summarisation method that is query-biased and easy to implement.

For summary generation, we used a sentence extraction approach that has been widely used [8]. First, both the sentence terms and query terms are stemmed using the Porter stemming algorithm [7], and stop-words are removed from both term sets. The summarisation method for selecting extract-worthy sentences of an XML element is as follows. The score of each sentence in a given element is calculated according to Equation 1.

$$S_i = \sum_{j \in Q} occ(j, T_i) \cdot occ(j, Q) \qquad (1)$$

where S_i is the score of sentence number i, Q is the set of unique query terms, T_i is the set of unique terms in sentence i, and $occ(j, A)$ is the number of times term j occurs in the term set denoted by A. If two sentences have the same score, the one occurring first in the element is given higher rank among candidate summary sentences. If the source of the summary does not contain sentences that include query terms, which can happen as summaries are generated for all elements of XML documents, the first four sentences of the element are being shown as the summary. This approach is based on the location method [3], which assumes that the first couple of sentences in a document, or paragraph, are more indicative of its content. A maximum of four sentences with the highest ranks are presented as extracts of the source XML elements, in order of appearance in the source element.



Showing /collections/inex/documents/inex-1.4/xml/an/1999/a3049.xml /article[1]/bdy[1]/sec[2] - Mozilla

http://joshua.dcs.qmul.ac.uk/zolley2/show.php?file=/collections/inex/documents/inex-1.4/xml/an/19

- A Short History of Computing in Hungary
  - Front Matter
    - Abstract
  - Body
    - INTRODUCTION
    - HUNGARY'S COMPUTING ROOTS
    - PRELIMINARY PHASE
    - THE POSTWAR ERA
    - THE BETWEEN YEARS IN HUNGARIAN COMPUTING
    - RECENT PAST
    - LATER YEARS
    - EPILOGUE
  - P

Summary

- The reader now has on outline of computing in Hungary almost up to the present
- Weighing everything, that Hungary once again became a free, competitive, and open society is considered a positive thing, while a number of socioeconomic phenomena and processes in the past few years have definitely had a detrimental effect on the country and are counted as negative factors
- There are now some companies in Hungary that are totally or partly in foreign hands and have well-established products and services in today's world markets
- We also should not forget that there are now some multinational corporations undertaking high-level research work in their subsidiaries in Hungary

Current numbe
You may chan

Default (4)

< To results of

<< Back to search <<

HUNGARY'S COMPUTING ROOTS

In 1796, in the last years of the glorious 18th century, an extraordinarily interesting machine was displayed in the Austrian royal court in Vienna. It was a machine figure playing chess with human partnersand with excellent skill. The automaton was designed and built by the Hungarian-born Baron Kempelen Farkas (1734-1804). (In Hungarian names, the family name comes first, while Christian names follow.) Even now, nobody knows for certain how it worked, but the chess-figure manipulator was probably a human dwarf concealed in the machine. In 1826, the machine was brought to the United States, but it was destroyed in a fire in Philadelphia. Baron Kempelen, however, did not regard the chess automaton as his main work; rather, Kempelen believed his series of speech machines was most important, the first of them having been produced in 1773.

Now let us jump a few decades. A notable Hungarian physicist named Jedlik nyos (1800-1895), a Benedictine monk, created a gadget for plotting Lissajous graphs in the early 1860s. He was also among the first (along with the German Werner Siemens) to describe the principle of the dynamo in 1861, which led to the self-acting dynamo. Sadly, he never patented his discoveries, and perhaps this is the reason why he has never been recognized as one of its first inventors.

The last two to three decades of the 19th century was a golden age in Hungary's history. The infrastructure and basis of a modern economic systemmost notably in industry, agriculture, science, and technologywere laid down in this period, and the country's economy flourished right up to the outbreak of World War I. For instance, the first telephone manufacturing company was founded in Budapest in 1882, just six years after Alexander Graham Bell's invention, and a research institute for telecommunications was established as early as 1891. In 1883, a telephone network was introduced in Budapest, soon followed by others in the bigger cities of the country. Just over 100

http://joshua.dcs.qmul.ac.uk/zolley2/show.php?file=/collections/inex/documents/inex-1.4/xml/an/1999/a3049.xml&path=/article[1]/bdy[1]/sec[8]

Figure 1: On the left, the structure of the XML document with a summary; on the right, the content of a section element displayed.

2.3 XML Retrieval Engine

The retrieval was based on the HySpirit retrieval framework [9]. HySpirit is capable of indexing and retrieving XML documents and elements based on a probabilistic framework that allows defining various retrieval strategies. To be able to examine the relation between the structural display and the use of summaries, only paragraphs were returned as retrieval results (the used document collection comprises scientific articles and is further described in Subsection 3.1). This strategy ensured that elements deeply nested in a document logical structure were returned, so that to "force" searchers to browse through the structural display on the left panel (instead of simply scrolling down the right window).

3. EXPERIMENTAL DESIGN

This section describes the document collection and experimental methodology we used in our study.

3.1 Document Collection

The document collection we used was the IEEE collection which contains 12,107 articles, marked up in XML, of the IEEE Computer Society's publications from 12 magazines and 6 transactions, covering the period of 1995-2002. On average, an article contains 1532 XML nodes and the average depth of a node is 6.9. These properties provided us a suitably large collection with articles of varying depth of logical structure.

3.2 Searchers

Twelve searchers were recruited for this study. All of them had computer science background as the collection used contained articles from the field of computer science. Nine of them were male, three female. Participants had five different languages as their first language.

3.3 Experimental and Control Systems

Two versions of the developed system were used in this study. The control system was the one described above (we refer to this as system S_c), the experimental system differed in the display mode of summaries (system S_e). While the control system was 'fully functional', the experimental system displayed summaries only at high levels in the hierarchical structure, i.e. the upper three levels had associated summaries, the fourth level did not. The rationale behind this is that we wanted to see whether searchers realise the difference and act differently.

From the observed difference, the usefulness of showing the structure of a document and summaries can be examined. To avoid bias towards the use of the hierarchical structure and summarisation, we employed a blind study, i.e. searchers were not told what the purpose of the study was.

3.4 Tasks

Four tasks were created for the experiments. The aim of making tasks was to create simulated work task situations [2]. As a first step of the task generation, four topics were chosen from the INEX 2005 ad hoc track topics. Choosing such topics ensured that relevant XML portions were present in the collection. Selected topics were then modified by adding introductory sentences that would serve as a background for searchers who would perform searches using the task generated from the given topic. Task description generation also included combining sentences from the various parts of the original INEX topics, e.g. title, initial topic statement, narrative.

Topics were also chosen in a way that allowed us to create two types of tasks, each containing two tasks. Background type tasks instructed searchers to look for information about a certain topic (e.g. concerns about the CIA and FBI's monitoring the public) while List type tasks asked searchers to create a list of products that are connected to the topic of their tasks (e.g. a list of found speech recognition software). We refer to these task types as B and L, respectively. The reason for creating more types of tasks is to avoid the effect of testing our hypotheses in only one type of task condition. From each group of tasks, searchers could freely choose the one that was more interesting to them. Searchers had a maximum of 20 minutes for each task. This period is defined as a *search session.* Search sessions of the same searcher (i.e. one searcher had two search sessions) are defined and used in this paper as a *user session.*

3.5 Search Design

To rule out the fatigue and learning effects that could affect the results, we adopted Latin square design. Participants were randomly assigned into groups of four. Within groups, the system order and the task order were permuted, i.e. each searcher performed two tasks on different systems which involved two different task types. We made an effort to keep situational variables constant, e.g. the same computer settings were used for each subject, the same (and only) experimenter was present, and the place of the experiments was the same.

3.6 Data Collected

Information was collected in three ways: searchers filled in questionnaires, searching logs were recorded and interviews took place at the end of each user session. In this paper, we report the results based on the questionnaires and interviews.

Questionnaires were filled in by searchers before and after each search session, and before and after each user session. Within the entry questionnaires, general information was collected about users (age, first language, education, etc.), and their computer and searching experience. Information about users' perception of the search tasks and systems was also collected in before- and after-each-task questionnaires. Questionnaires included questions about specific features of the user interface, i.e. the use of summaries and the displayed XML structure. Comments and suggestions were also recorded in the after-each-task and exit questionnaires.

After each user session, searchers were interviewed. In the interviews, their use of the system, summaries and XML structure was discussed in detail to understand how these had affected their searching behaviour and user satisfaction.

4. RESULTS

In this section, results collected by questionnaires and interviews are presented and discussed. We first describe our users' familiarity with computer science and searching, and examine their search task and system understanding. This is followed by results with respect to the different task types (L and B). We also describe the differences between the two systems (S_e and S_c). Finally, the outcome of the interviews is described. Apart from comments written by searchers, we used 7 point Likert scales to measure users' perceptions, where 7 corresponds to the strongest and 1 to the weakest agreement with regards to the question asked. In this section, we refer to such a scale unless otherwise stated.

According to the entry questionnaires, most of the users claimed to be experts in both working with computers and searching (average of 5.9 and 6 points, respectively, where 1 indicated low level, 7 indicated high level of expertise, Table 1). This indicates that they were able to understand the search tasks as well as the search results as both the document collection and the search tasks were computer science oriented. Searchers also indicated their task understanding as 5.75 in average.

Information about users' perception of the given tasks was also collected. Searchers found that the difficulty of tasks in general was average both before starting the search (mean of 4.54 where 7 corresponds to easy) and after finishing it (mean 4.33). As they also understood how to use the two systems easily (6.17 average), we believe the system did not cause frustration to them which would have had an unwanted effect on their searching behaviour. Our second set of results are with respect to task types summarised in Table 2. L tasks appeared to be easier to start with, but users did not find difference in terms of task difficulty later in their search process. According to a t-test (α=0.05), there is no statistically significant difference in these results either (as values are equal or close to 1). Both task types appeared

Table 2: Questionnaire Data by Task Types.

	Task B		Task L		
	Average	St. dev.	Average	St. dev.	t-test
Was it easy to get started on this search?	4.50	1.51	5.25	1.48	0.21
Was it easy to do the search on this topic?	4.33	1.61	4.33	1.78	1.00
Did you have enough time[...]?	4.58	1.24	4.00	2.30	0.36
Are you confident in your results?	4.50	1.45	4.42	2.31	0.92
Do you feel your results are complete?	3.58	1.24	3.33	1.72	0.73
Did your previous knowledge[...]help[...]?	3.17	1.95	3.58	1.78	0.45
Have you learned much new about the topic[...]?	4.08	1.56	4.17	1.64	0.89
Was the search task interesting?	5.08	1.08	5.08	1.08	1.00
Was the search task realistic?	5.58	1.00	5.25	1.29	0.47
Did you use the hierarchy[...]?	4.67	1.61	5.08	1.16	0.32
Did you read the summaries[...]in the hierarchy?	4.92	1.68	4.92	1.56	1.00

Table 1: Questionnaire Data about searchers' expertise, task and system difficulty.

	Average	St. dev.
Level of expertise with computers	5.92	0.90
Level of expertise with searching	6.00	0.60
Understanding the nature of searching task?	5.75	0.87
How easy do you think the task is?	4.54	1.10
Was it easy to do the search on this topic?	4.33	1.66
Understanding how to use the system?	6.17	0.83
How easy was it to learn to use the system?	5.83	0.83
How easy was it to use the system?	5.67	0.89

to be quite interesting (on average 5.08 for both task types) and realistic (5.58 and 5.25), where realism is explained to searchers as how likely it is to have such search task in real life. Searchers read about the same amount of summaries regardless of task types, and there was a small difference in the use of the hierarchical structure. Regarding the use of the two system versions (our third set of results), as expected, there is a considerable difference in reading summaries and preferring them at all levels of hierarchy (Table 3). Results show that using the complete system (S_c), searchers read more summaries as, by the design of the two systems, more summaries were displayed to them. The display of summaries at all levels was preferred by searchers.

In the corresponding questionnaires, users indicated problems with summaries at low (i.e. paragraph) level in 8 out of 24 search sessions. 5 cases concerned system S_e where searchers reported missing summaries, whereas 3 were with system S_c where searchers did not want to see those summaries (at paragraph level).

During the interviews some of the searchers who did not comment on the use of summaries at low level of the structure within their questionnaires, stated that, in fact, they would have liked to have summaries for paragraphs when using system S_e. They said that they did not comment on this issue because they did not realise that some summaries were actually missing (especially when they performed search on system S_e first) as they were focusing on the retrieval performance of the systems, not the use of structural information and summaries[1]. Regardless of systems and tasks, searchers were satisfied with summaries as tool tips (5.54 average).

The last results are based on users' comments obtained during the interviews after the user sessions. In addition to provide answers to our investigation, these comments helped identifying possibly typical problems and requirements of an interactive XML retrieval system (this is currently investigated as part of the interactive track in INEX).

Several searchers expressed that they did not like the four level display of the structure but they did not change the number of levels although it was possible. People also said fewer levels would not have been informative enough for them. This indicates that searchers expect automatic determination of what structural elements should be displayed and a general 'number of displayed levels is x' approach is not suitable.

Some of the searchers, who had problems with the number of structural levels displayed, indicated that elements without title should be displayed using labels instead of their types and numbers (e.g. Section 3). This shows that information that is visible without being explicitly requested through some user action is very important.

Some searchers complained about summaries having no query terms. This was the case with summaries of non retrieved elements contained in documents for which the structure was displayed (these were the documents composed of elements that were retrieved). The aim of an XML retrieval is to identify relevant elements, and to return only those. The same aim should be adopted when displaying the logical structure of a document: relevant elements (as estimated by the retrieval system) should be made more prominent compared to non relevant elements. How to do achieve this effectively is a research question in itself.

Searchers indicated that when an element could fit in the right window (Figure 1), showing the structure of that element on the left window (its descendants) was not necessary. This is because they can easily find the relevant information in a window as long as they do not have to scroll down.

Searchers who realised the difference between the two systems, and also remembered this during the interviews, claimed that missing summaries were disturbing; they also claimed that summaries shown at low levels, although they could be unnecessary, were rarely bothering. We interpret this as a strong connection between the structural display and summary display, i.e. when structural items are shown, corresponding summaries should always be displayed.

5. DISCUSSION

Based on the questionnaire data and interviews with par-

[1] We think this is because most users associate information retrieval with web search, where links are returned and target documents are displayed by the browser according to their HTML content, and no further processing is done by the IR system.

Table 3: Questionnaire Data by System Types.

	System S_c		System S_e		
	Average	St. dev.	Average	St. dev.	t-test
Did you have enough time[...]?	3.92	1.73	4.67	1.92	0.23
Are you confident in your results?	4.00	2.09	4.92	1.62	0.22
Do you feel your results are complete?	2.92	1.51	4.00	1.28	0.12
Did you use the hierarchy[...]?	4.83	1.53	4.92	1.31	0.85
Did you read the summaries[...]in the hierarchy?	5.42	1.56	4.42	1.51	0.05
Did you like summaries[...]?	5.17	1.75	5.25	1.48	0.75
Did you like summaries at *all levels*[...]?	5.08	1.68	4.17	1.85	0.16

ticipants of this study we believe that summarisation can be helpful in interactive XML retrieval. However, in order to be able to investigate particular summarisation algorithms in a retrieval system such as that described in this paper, display of document structure has to be well controlled. We believe that the structural document display and summarisation for XML elements is strongly connected. If the display is not well designed, development including evaluation of summarisation strategies is not reliable in an interactive environment as users are distracted by the effect of the display method.

To control the effect described above, the following design guidelines are proposed as a result of this study.

- Structured items should be displayed based on the (estimated) relevance of the corresponding element. This is because users do not want to waste their time being directed to irrelevant information.

- Structural items should be displayed according to target element size, and this independently to their content, i.e. whether they composed of figures, tables, textual and non textual content. Indeed, users indicated a relation between the need to display the structure of an element and the element length.

- Labels of the structural items should be as informative as possible. Titles, when available are the ideal choice. If not present, keywords of the corresponding elements should be displayed as labels. Types of elements, e.g. section, figure, etc. are not satisfactory.

- Summaries should be displayed for each item in the structural display. Alternatively, summaries should be completely avoided as selective summary presence can disturb users.

6. FUTURE WORK

The work presented here is a part of a wider work that aims at developing and evaluating summarisation methods for structured information retrieval. Based on the findings of this paper together with the analysis of the system logs of this user study, an improved interactive XML retrieval system will be developed and evaluated. We are also aiming at developing summarisation methods that consider the structural position of the element to be summarised (some initial work is done in [1]) and the fact that within an XML document, some elements will be estimated relevant and some not. This has to be taken into account when deciding which element to return and how to display in the logical structure. We will also take into account structural IR related search task types, e.g. focussed, fetch and browse, that is currently being investigated at INEX. The aim of the fetch and browse retrieval strategy is to first identify relevant documents (the fetching phase), and then to identify the most relevant elements within the fetched articles (the browsing phase). We believe that summarisation, the use of which was introduced in this paper, can be particularly helpful in the browsing phase, where finding relevant elements within a document is required.

7. ACKNOWLEDGMENTS

This work was partly funded by the DELOS Network of Excellence in Digital Libraries, to which we are very grateful.

8. REFERENCES

[1] M. Amini, A. Tombros, N. Usunier, M. Lalmas, and P. Gallinari. Learning to summarise XML documents by combining content and structure features (poster). In *CIKM'05*, Bremen, Germany, October 2005.

[2] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003.

[3] H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2):264–285, 1969.

[4] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors. *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004*, volume 3493. Springer-Verlag GmbH, may 2005.

[5] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *SIGIR'99*, pages 121–128. ACM Press, 1999.

[6] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *SIGIR'95*, pages 68–73. ACM Press, 1995.

[7] M. Porter. Porter stemming algorithm. http://tartarus.org/~martin/PorterStemmer.

[8] G. Rath, A. Resnick, and R. Savage. The Formation of Abstracts by the Selection of Sentences: Part 1: Sentence Selection by Man and Machines. *American Documentation*, 12(2):139–141, 1961.

[9] T. Rölleke, R. Lübeck, and G. Kazai. The hyspirit retrieval platform. In *SIGIR'01*, page 454, New York, NY, USA, 2001. ACM Press.

[10] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *SIGIR'98*, pages 2–10. ACM Press, 1998.