

# XML Information Retrieval Considering Physical Page Layout of Logical Elements

Toshiyuki Shimizu<sup>\*</sup>  
Graduate School of Informatics  
Kyoto University  
shimizu@soc.i.kyoto-u.ac.jp

Masatoshi Yoshikawa  
Graduate School of Informatics  
Kyoto University  
yoshikawa@i.kyoto-u.ac.jp

## ABSTRACT

XML information retrieval (XML-IR) systems utilize the logical structure of XML documents for retrieving relevant elements. From a practical point of view, displaying the search results of XML-IR systems is important to achieve. When we search XML documents that are constructed by marking up documents originally composed of pages, such as scholarly articles or books, we would like result elements to be overlaid on the physical layout of pages in the user interfaces. We propose such a displaying method for keyword searches on XML documents of scholarly articles and ranking methods based on page units. We also need a new ranking method different from those used in simple element ranking because multiple result elements may be in the same page. We propose a ranking method considering the *benefit* that we obtain from the result elements and the *reading effort* that needs to be spent in reading the result elements and nearby elements to understand the content of the result elements.

## 1. INTRODUCTION

XML information retrieval (XML-IR) systems retrieve elements relevant to a certain topic from a collection of XML documents. For example, target XML documents are scholarly articles, XML-IR systems retrieve sections, subsections, paragraphs and so on.

Many research groups in INEX [1] are evaluating the effectiveness of their XML-IR systems. The retrieved elements from XML-IR systems are ranked in order of relevance scores. The INEX 2005 project defined three strategies for element retrieval by keyword searches. A system with the *Thorough* strategy simply retrieves relevant elements from all elements and ranks them in order of relevance. The retrieved elements using the *Thorough* strategy may overlap due to nestings. A system with the *Focussed* strategy retrieves only focussed elements (i.e., non-overlapping ele-

ments) and also ranks them in order of relevance. A system with the *FetchBrowse* strategy first identifies relevant documents (the fetching phase) and then identifies relevant elements within a fetched document (the browsing phase). The three retrieval strategies of INEX 2005 were designed to evaluate XML-IR systems and were not necessarily intended to be used in designing user interfaces.

From 2002 to 2005, the INEX project had made a test collection for XML-IR using scholarly articles of the IEEE Computer Society's publications marked up in XML. The physical layouts are fixed for such XML documents created by marking up original PDF files. Hence, it is natural to show search result elements mapped on a physical page image. We believe that XML-IR systems for such XML documents would be more useful if they provided a user interface that can treat physical pages as a basic granularity for display. Because result elements with low scores are noise, we first retrieve high-scored elements using the top-k search with the *Thorough* strategy. The system provides thumbnails of the physical layouts in the user interfaces, and users can intuitively understand the search results.

We propose a new concept called *PAG* (Page Aggregation Granularity), which is a granularity of search results based on page units. In addition, we also propose *ARA* (Augmented Reading Area) which is an area a user would need to read to understand the content of the search result, and consider *RS* (Recommendation Score) for ARA. *PAG* and *ARA* enable more practical ranking that matches user behavior.

## 2. PAGE AGGREGATION GRANULARITY

When the system outputs the search results on the physical layout of pages, multiple result elements may be in the same page. If this happens, users can browse these elements at the same time even if the score of some of the elements is not very high. For example, the search results of the *Thorough* strategy in Figure 1(a) can be mapped on the pages in Figure 1(b). Elements  $e_5$  and  $e_8$  in document  $d_2$  are on the same page,  $p_2$ . In Figure 1(a),  $e_{12}$  within  $d_1$  is the top ranked element; however, in Figure 1(b), the page within  $d_2$  is the top ranked page. For the sake of simplicity, in Figure 1, we considered the score of the page is the sum of the scores of the elements within the page. In addition, the pages can be aggregated and shown by a document like the *FetchBrowse* in INEX 2005. In our example,  $d_1$  is the top ranked document as shown in Figure 1(c).

*PAG*, as stated, is a granularity of search results based on page units. In this paper, we consider only aggregation

---

<sup>\*</sup>JSPS Research Fellow

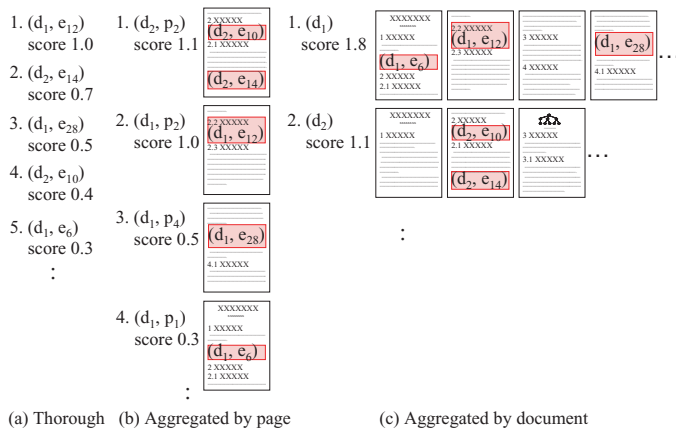


Figure 1: Aggregation of result elements.

by page and aggregation by document, as Figure 1(b)(c) depict. An arbitrary number of pages can be aggregated in general, and it is considered to be effective that the system automatically decides the appropriate PAG.

### 3. AUGMENTED READING AREA

Users typically read the preceding and following content near result elements in addition to the result elements themselves to better understand the content of the result elements. Such areas are *ARAs*, and an *RS* is assigned for each *ARA*. When users input a threshold value of *RS*, the system displays elements with an *RS* value larger than the threshold value near result elements.

In general, the elements near result elements are assigned a high *RS* value. To calculate the *RS* value, we can use the structural information, the length of the element, or other information. For example, let us consider the case when the preceding-sibling nodes of the result element are assigned a high *RS* value. Figure 2(a) (and Figure 2(b)) is an example of considering *ARA* of Figure 1(b) (and Figure 1(c)) when the preceding-sibling and self nodes of the result element are an *ARA* with an *RS* value greater than the threshold user input. The corresponding logical structure of  $d_1$  is shown in Figure 3 with result elements and the *ARA*. The *ARA* is highlighted in the user interfaces.

INEX 2006 introduced the concept of *BEP* (best entry point). *BEP* is the starting-point in the article from which users read the relevant information in the article. *BEP* is a similar concept to *ARA*. However, *BEP* provides only one starting-point in the article with no implied end-point.

The amount of the *ARA* that a user reads is the *reading effort* for obtaining the *benefit* from the content of the *ARA*. The ranking of the search results is calculated using the *benefit* and *reading effort*. For example, we can calculate the score of a result element by dividing the *benefit* by the *reading effort*.

The *benefit* is the amount of gain by reading the element. We can use the relevance score of the element against the query as the *benefit*. The relevance scoring formula in our past research [2] is one available example. The component of element length in the formula must be dropped because the element length is considered to be the *reading effort*. The *benefit* of the element is the sum of the *benefit* of the child nodes. Therefore, the *benefit* of leaf nodes is calculated

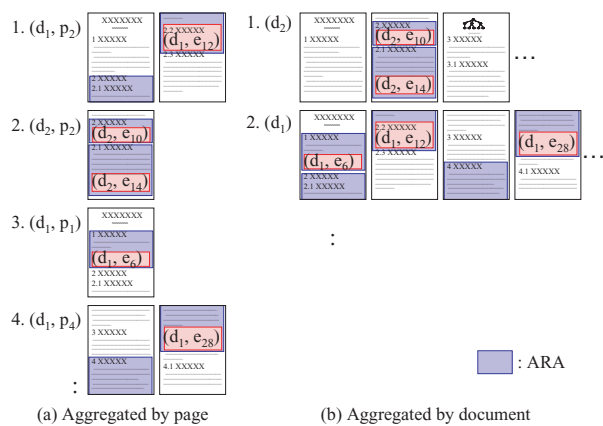


Figure 2: Ranking considering *ARA*.

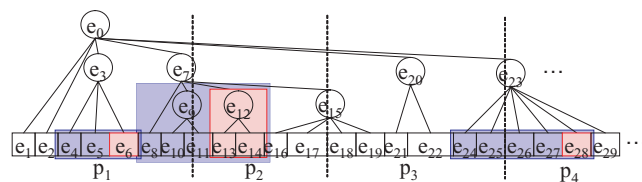


Figure 3: Tree structure of  $d_1$ .

first, and then the *benefit* of the parent node is obtained recursively.

The *reading effort* is the cost of reading elements. For example, it can be calculated using the amount of words in the element. If the *ARA* of the result is large, the amount of *reading effort* is also large, and the result is ranked low. Note that, though we first retrieve high-scored elements by top-*k* searches with the *Thorough* strategy and display top-*k* results highlighted in the user interface, the elements under rank *k* also have *benefit* that is not shown explicitly. So we need to take into account the invisible *benefit* in *ARA*.

### 4. CONCLUSIONS

We displayed result elements of XML-IR systems overlaid on the physical layout of pages in user interfaces and ranked them based on the page unit. We described new concepts *PAG* and *ARA* for a more practical ranking that matches the usage scenario.

Our future work will include studying how to calculate appropriate *PAG* automatically to enable browsing search results effectively. For example, aggregating  $p_1$  and  $p_2$  in  $d_1$  and displaying the three page sets  $(d_1, p_1-p_2)$ ,  $(d_2, p_2)$ , and  $(d_1, p_3-p_4)$  in this order may be a good solution for Figure 2.

### 5. REFERENCES

- [1] INEX. Initiative for the Evaluation of XML Retrieval. <http://inex.is.informatik.uni-duisburg.de/>.
- [2] T. Shimizu, N. Terada, and M. Yoshikawa. Kikori-KS: An effective and efficient keyword search system for digital libraries in XML. In *ICADL*, pages 390–399, 2006.