Focussed Structured Document Retrieval

Gabrialla Kazai, Mounia Lalmas and Thomas Roelleke
Department of Computer Science, Queen Mary University of London,
London E1 4NS, England
{gabs,mounia,thor}@dcs.qmul.ac.uk,
http://qmir.dcs.qmul.ac.uk

Abstract. Focussed structured document retrieval aims at retrieving best entry points from where users can browse to access relevant document components in the document structure. In this paper, we report on the development, implementation and evaluation of best entry point retrieval strategies derived from user studies designed to elicit what constitutes a best entry point.

1 Introduction

With the rapid adoption of the XML markup language on the Web and in digital libraries there is more scope and need to exploit the structural knowledge of documents for the purpose of their retrieval. Numerous studies (e.g. [3,5,7]) have highlighted that indexing structured documents based on combined structure and content knowledge can improve retrieval effectiveness. In addition, this combination makes it possible to retrieve relevant document components of varying granularity, for example, a document component when only that component is relevant, a group of components, when all the components in the group are relevant, or the document itself, when the entire document is relevant. The retrieval result of structured document retrieval is then a ranked list of pointers to relevant document components, which may be "related" to each other, e.g. a component and its sub-components. According to the ranking algorithm these related components may be displayed at distant locations in the result. This can waste user time and lead to user disorientation [1]. By exploiting structural knowledge these relationships can be made explicit to the user.

In this paper, we develop an approach that allows to *focus retrieval* to the so-called *best entry points* (BEPs), which correspond to document components from which users can browse to access relevant document components. Returning BEPs, and not merely relevant document components, is a means to capture relationships between retrieved document components. The approach defines the representation of a document component as the *aggregated* representation of its own content and the content of its structurally related components. Using the aggregated representation of document components, BEPs are selected based on criteria elicited from user studies.

The rest of this paper is organised as follows. In Section 2, we describe our approach for obtaining aggregated representations of document components and selecting BEPs. In Section 3, we describe a test collection of structured documents based on XML that was built to evaluate our approach. Section 4 describes a number of experiments that were carried out to evaluate the effectiveness of both the aggregation strategies and BEP selection algorithms. We conclude in Section 5.

2 The Approach

We are concerned with document structure that can be viewed as a tree whose nodes are components of the document and whose edges represent the relationships between the connected nodes. A document component can be either a leaf or a composite node. Leaf nodes are document components that correspond to the last elements of hierarchical relationship chains. All other nodes are composite nodes. Both leaf and composite nodes can contain raw data (text), referred to as the node's own content (in this paper we use the terms "node" and "component" interchangeably).

Best entry points. For the focussed retrieval of structured documents, we first determine what constitutes a BEP. Based on two user studies [2,4], we adopt the following two criteria: (C1) a super-node is retrieved instead of its sub-nodes if *many* of the sub-nodes have been estimated relevant to the query; otherwise the sub-nodes are retrieved individually. (C2) Only the *first* node in a linear sequence of closely related nodes that have been retrieved according to C1 is returned as a result. To implement C1, the estimation of relevance, i.e. the retrieval status value (RSV) of a super-node, is based on a content description that is derived from its own content and the content of its sub-nodes. For this purpose, we aggregate the content of a super-node with that of its sub-nodes taking into account their so-called *accessibility*, which reflects the extent to which a sub-node's content should contribute to the super-node's aggregated representation. [6]. The aggregation process is applied to the whole document structure, starting with the leaf nodes, where no aggregation is performed.

Aggregated representations. Consider a super-node p composed of m sub-nodes $c_1,\ldots,c_m$. Let $P(term(t,p))$ be the probability of the event $term(t,p)$ that term t describes the own content of p, $P(term(t,c_k))$ be the probability of the event $term(t,c_k)$ that term t describes the own content of the k-th sub-node, and $P(term^*(t,c_k))$ the probability of the event $term^*(t,c_k)$ that term t describes the aggregated representation of the k-th sub-node, for k=1…m. For a leaf node c_k, $P(term^*(t,c_k)) = P(term(t,c_k))$. Let $P(acc(p,c_k))$ be the probability of the event $acc(p,c_k)$ that the k-th sub-node is accessed from p. We define the aggregated representation of term t in p, $P(term^*(t,p))$, as a probabilistic disjunction of the events $term(t,p)$, $term^*(t,c_1) \wedge acc(p,c_1)$, …, $term^*(t,c_m) \wedge acc(p,c_m)$. To reduce complexity, we assume independence over the events, and therefore calculate $P(term^*(t,p))$ as:

$$P(term^*(t,p)) = 1 - (1 - P(term(t,p))) \prod_{k=1}^{m} (1 - P(acc(p,c_k))) P(term^*(t,c_k))$$

Estimating the probabilities. $P(term(t,p))$ and $P(term(t,c_k))$ can be estimated using standard term weighing schemes. The probability $P(acc(p,c_k))$, referred to as the *accessibility function*, is estimated in the following ways:

(a) $\frac{1}{m}$ (b) $\frac{1}{m^2}$ (c) $\frac{1}{m^{0.5}}$ (d) $\frac{1}{d_{ck}}$ (e) fixed values of 0.2, 0.4, 0.6 and 0.8

In (a), $P(acc(p,c_k))$ is viewed as the probability of a user, randomly browsing the structure of a document, arriving in node c_k from node p. (b) and (c) are based on the findings in [6] and allow to emphasise or de-emphasise the effect of the number of sub-nodes in the aggregation. (d) is motivated by the findings of the user study in [4], which suggests that users generally prefer to retrieve more specific, smaller contexts to general, larger contexts. The "depth" d_{ck} is measured as the distance between the sub-node c_k and the last element of the longest structural chain that c_k is an element of. Finally, (e) uses fixed values of 0.2, 0.4, 0.6, and 0.8.

Retrieval strategies. During retrieval, a query is matched against the aggregated representation of the document components, where the probability of relevance of a document component *a_node* to the query *q* is calculated as follows.

$$P(a\_node \text{ is relevant to } q) = 1 - \prod_{t \in q} (1 - P(term^*(t, a\_node))$$

We refer to this strategy as our baseline retrieval (BR). We also investigated several recall-enhancing retrieval strategies exploiting the structural nature of the XML data in order to increase retrieval performance and allowing us to observe the behavior of the BEP selection algorithms. These strategies are specific to the collection used to evaluate our approach, but can be adapted for other collections of structured documents. The strategies are:

- Cascaded Retrieval (CR): retrieves the sibling nodes of a relevant Speaker elements. This strategy aims to compensate for the Speech-Speaker semantic relationship being expressed as parent-child elements in the collection instead of Speaker being an attribute of Speech.
- Enforced Retrieval (ER): enforces the retrieval of all descendant elements of relevant Speech elements.
- Cascaded Enforced Retrieval (CER): applies first the CR strategy then the ER strategy.
- United (UNT): performs retrieval using the CR and ER strategies independently and then joins their result sets.

With CER sub-nodes of Speech elements containing relevant Speaker elements are elevated to higher ranks, whereas UNT will only increase the RSV of those sub-nodes of a Speech that contain both relevant Line and Speaker elements.

Focussed retrieval algorithms. The results of the above retrieval strategies are ordered lists of document components that may be structurally related. The ranking of the document components varies depending on the applied aggregation strategy. To return only the BEPs to the user, we remove *redundant* nodes from the result list using the two criteria C1 and C2. With respect to C1, redundant nodes are those components in the ranking, which are hierarchically related to BEPs with a *higher* RSV. Two nodes, *n1* and *n2*, are hierarchically related if *n1* lies on *n2*'s path and vice versa. This is implemented in an algorithm referred to as R1. With respect to C2, every node, but the first, in a linearly related chain is considered redundant. To implement this, the algorithm R2 selects the first element of a linearly connected chain of nodes, where two nodes are linearly connected if they are no further away from each other than a preset threshold value (derived from our user study [4]).

3 Structured test collection

To evaluate the effectiveness of our approach, we constructed a test collection, which consists of a set of structured documents, queries, "standard" relevance assessments (the document components that are relevant to a given query), and "best entry point" relevance assessments (the BEPs for each query). The test collection was based on the Shakespeare plays marked up in XML by Jon Bosak (http://www.ibiblio.org/bosak/). The queries, relevance assessments and BEPs were obtained from 11 English and Drama students who were experts in Shakespeare's works. The test collection consists of 37 plays, with an average of 30,600 words per play, a total 180,000 retrievable elements, 25 queries, with an average of 4 words per query, and an average of 124 relevant elements and 12 BEPs per query. The relevance assessments were obtained at leaf node level. The relevance of higher structural levels was then computed automatically as follows: a composite element was judged relevant if at least one of its child elements was

marked relevant. The maximum depth of nested XML tags is 6 (Play, Act, Scene, Speech, Line, StageDir (stage direction)). These correspond to retrieval elements of varying complexity. The test collection is available at http://qmir.dcs.qmul.ac.uk/Focus/resources.htm.

Analysis of the collected BEPs [4] showed that users preferred to retrieve smaller, more specific contexts to larger, more general components. Another result of the study was that users preferred to see the context within which relevant elements occurred, but priority was given to having their attention directed at the relevant components. The algorithm R1 applied to the aggregated representations of document components reflects this finding. Regarding the linear relationship between relevant nodes, the first node of a linear chain was in most cases selected as BEP. The BEP selection algorithm R2 relates directly to this finding.

4 Experiments and Results

We designed our experiments with the aim to identify which combinations of aggregation and BEP algorithms produce the best retrieval performance. We indexed the document components' content using *idf* weighting (here, inverse node frequency) after stopword removal and stemming. We implemented our aggregation method using the eight different accessibility functions (denoted *acc*). We performed retrieval on the aggregated representations using the five retrieval strategies. We then implement the algorithms R1 and R2 to remove redundant nodes from the obtained ranked lists. We evaluated our retrieval results with respect to the retrieval effectiveness of relevant nodes and BEPs.

Retrieval of relevant document components. Table 1 shows the precision values (in percentage) for retrieving relevant nodes using the eight *acc* functions, averaged over the 11 recall points and the 25 queries. The best performances for each strategy are highlighted in bold. The results show that all recall-enhancing strategies outperform the baseline retrieval method. The best retrieval performance is achieved by the CER strategy, which produces the highest precision values for seven of the eight *acc* functions, including the highest overall precision for $P(acc(p,c_k))$=0.2. One finding is the performance difference between the *acc* functions of $1/m$, $1/m^{0.5}$ and $1/m^2$ using the baseline strategy. Another conclusion is that although all eight of the *acc* functions produce similar precision values, those employing fixed *acc* functions tend to outperform those based on structure-dependant *acc* functions.

acc	BR	CR	ER	CER	UNT
$1/m$	9.64	20.90	26.96	28.91	29.75
$1/m^{0.5}$	13.36	21.51	25.63	31.53	28.95
$1/m^2$	9.81	21.55	25.81	29.12	28.77
$1/d_{ck}$	16.56	24.97	26.21	28.67	27.47
0.2	12.46	24.50	**29.55**	**31.67**	29.98
0.4	15.19	25.21	29.27	31.05	**29.99**
0.6	15.72	25.51	28.84	31.41	29.20
0.8	**17.08**	**26.31**	27.20	31.52	27.61

Table 1. Average precision of retrieval of relevant nodes

Retrieval of best entry points. Table 2 lists the average precision values calculated against the BEP relevance set before removing redundant nodes from the result set. With the baseline retrieval the highest average precision is only 10%. The recall percentage of retrieved relevant BEPs were, however, comparable for all *acc* methods (75%), so the low precision values are

due to BEPs being retrieved at lower ranks. This indicates that the applied *acc* functions do not push BEPs high enough in the result set.

Looking at the effect of the recall-enhancing strategies, the improved effectiveness achieved when retrieving relevant nodes is not reflected in the retrieval of BEPs (average improvement over the baseline is 19% for CR, 32% for ER, -4% for CER and 23% for UNT). It can be inferred that although these particular strategies proved to be effective for the retrieval of relevant components, they are less suitable for the retrieval of BEPs. Although they have an overall positive effect, they tend to retrieve too many non-BEP elements, particularly at higher ranks. This could be explained by the fact that as recall-enhancing strategies, CR and ER retrieve additional elements (with equal RSV), which "dilute" the result set, and lead to BEPs being retrieved at lower ranks. Furthermore, the CER strategy, which proved the most successful for the retrieval of relevant components, produces the worst performance values for the retrieval of BEPs. The best overall performance is achieved by the *acc* functions using $P(acc(p,c_k))=1/d_{ck}$, $P(acc(p,c_k))=0.4$ and $P(acc(p,c_k))=1/m^{0.5}$.

acc	BR	CR	ER	CER	UNT
$1/m$	5.17	5.07	4.45	2.06	3.76
$1/m^{0.5}$	7.21	9.78	10.77	9.67	11.52
$1/m^2$	4.82	3.92	3.81	1.90	3.06
$1/d_{ck}$	**10.09**	11.46	11.88	8.97	**12.99**
0.2	6.90	11.78	11.44	7.77	11.16
0.4	9.05	**12.10**	13.28	**10.62**	11.50
0.6	8.48	9.48	10.49	9.70	10.82
0.8	7.19	7.63	**13.52**	9.00	10.16

Table 2. Average precision of retrieving BEPs

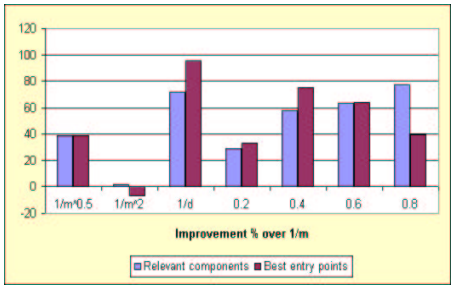

Figure 1. Relative baseline performance of *acc* functions to $1/m$

Figure 1 shows the relative performance of the different *acc* functions in retrieving relevant document components and BEPs compared against $P(acc(p,c_k))=1/m$. It can be seen that the relative performance of most *acc* functions for retrieving BEPs is similar to the relative performance of retrieving relevant document components when compared against $P(acc(p,c_k))=1/m$. The main difference is the relative improved performance of $P(acc(p,c_k))=1/d_{ck}$. Also notable, is the relative performance drop of $P(acc(p,c_k))=0.8$, which produced the highest improvement of 77% for the retrieval of relevant components, but exhibits only 39% improvement in retrieving BEPs over $P(acc(p,c_k))=1/m$ using baseline retrieval. Meanwhile $P(acc(p,c_k))=1/d_{ck}$ shows 95% improvement when retrieving BEPs, compared with 72% when retrieving relevant components.

Effects of R1 and R2. The difference in retrieval effectiveness between the results obtained in the previous section, and the results after applying R1 alone and R1 followed by R2 are shown in Figures 2 and 3, respectively. Algorithm R1 leads to a performance decrease, which suggests that BEPs are being removed by R1. The function $P(acc(p,c_k))=0.8$ suffers the worst performance drop across all retrieval strategies and $P(acc(p,c_k))=0.6$ and $P(acc(p,c_k))=0.4$ are also greatly affected. The functions least affected, and even improved, by R1 are $P(acc(p,c_k))=1/m^2$ and $P(acc(p,c_k))=1/m^{0.5}$, which assign higher RSV to BEPs. $P(acc(p,c_k))=0.2$ and $P(acc(p,c_k))=1/d_{ck}$, also show tendencies of retrieving BEPs with higher RSV. The overall best performing *acc* function is $P(acc(p,c_k))=1/m^{0.5}$. With respect to R2, we can observe that there is an improvement of up to 35% for structure-dependant *acc* functions, whereas the performance

of fixed-valued *acc* functions reduces by up to 74%. This indicates that R2 seems a promising BEP selection strategy.

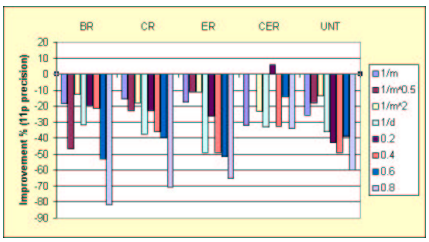

Figure 2. Effect of R1 on average precision

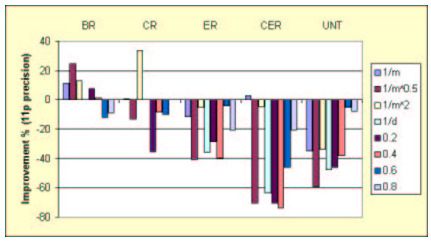

Figure 3. Effect of R1+R2 on average precision

5 Conclusions

This paper reported on an approach for the focussed retrieval of structured documents. Following from various theoretical and empirical studies regarding structured document retrieval, we developed, implemented and evaluated document representations based on the aggregated representation of their components, retrieval strategies, and BEP selection algorithms. The evaluation was done using the Shakespeare test collection that was specifically built for the purpose. Our experiments showed that the aggregation of document component representations as well as retrieval strategies reflecting user conceptions of relevance in our test collection led to effective retrieval of relevant nodes. However, these same user conceptions led to poor performance with respect to the selection of BEPs. We also observed that recall-enhancing strategies, although successful in improving the effectiveness of retrieving relevant nodes, are harmful for retrieving BEPs. We believe that the nature of the data set used in our experiments, with its specific characteristics, also had strong implications. We are currently investigating the evaluation of our approach with other test collections of structured documents.

Acknowledgements. This work has been carried out in the framework of the EPSRC Project GR/N37612 and the British Council ARC Project 1162.

References

1. Y Chiaramella. Browsing and querying: two complementary approaches for multimedia information retrieval. *Hypermedia - Information Retrieval - Multimedia*, pp 9-26, 1997.
2. M Hertzum, M Lalmas and E Frokjer. How Are Searching and Reading Intertwined during Retrieval from Hierarchically Structured Documents?, *INTERACT 2001*, pp 537-544, 2001.
3. E Kotsakis. Structured Information Retrieval in XML documents. *17th ACM Symposium on Applied Computing*, pp 663-667, 2002.
4. M Lalmas, J Reid and M Hertzum. Information-seeking Behaviour in the Context of Structured Documents, 2002 (Submitted for Publication).
5. S Myaeng, DH Jang, MS Kim and ZC Zhoo. A flexible model for retrieval of SGML documents. *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp 138-145, 1998.
6. T Roelleke, M Lalmas, G Kazai, I Ruthven and S Quicker. The Accessibility Dimension for Structured Document Retrieval, *European Colloquium on Information Retrieval Research*, pp 284-302, 2002.
7. R Wilkinson. Effective Retrieval of Structured Documents. *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp 311-317, 1994.