Using Structural Relationships for Focused XML Retrieval

Georgina Ramírez, Thijs Westerveld, and Arjen P. de Vries

Centre for Mathematics and Computer Science,
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
{georgina, thijs, arjen}@cwi.nl

Abstract. In *focused* XML retrieval, information retrieval systems have to find out which are the most appropriate retrieval units and return only these to the user, avoiding overlapping elements in the result lists. This paper studies structural relationships between elements and explains how they can be used to produce a better ranking for a focused task. We analise relevance judgements to find the most useful links between elements and show how a retrieval model can be adapted to incorporate this information. Experiments on the INEX 2005 test collection show that the structural relationships improve retrieval effectiveness considerably.

1 Introduction

Structured document retrieval systems use conventional information retrieval techniques to determine the order in which to best present results to the user, but, as opposed to traditional IR systems, they *also* choose the *retrieval unit*. Given an XML document collection, a structured document retrieval system could present to the user any of the marked up elements. The possible retrieval units vary widely in size and type; from small elements like italics and section titles, to large elements like document bodies, articles or even complete journals. It is the task of the system to decide which units are the most sensible to retrieve.

In *focused* XML retrieval (introduced at INEX 2005, see Section 2), this problem is even more apparent. The goal is to avoid returning overlapping elements in the result set, to make sure the user does not get to see the same information twice. For example, when both a paragraph and its containing section contain relevant information, a system is not allowed to return both elements. Instead, it should decide which of the two candidate results is the more useful. Thus, it should reason that when the section contains only one relevant paragraph, this paragraph may be more useful on its own than the section, while the user may prefer the full section if it contains multiple relevant paragraphs.

The typical approach to focused XML retrieval produces an initial ranking of all elements using an XML retrieval model, and then applies an algorithm to remove overlapping elements from the result set. Two main types of strategies have been proposed for overlap removal. The ones that use a simple method such as keeping the highest ranked element of each path and the ones that apply clever algorithms that take into account the relations in the tree hierarchy between the highly ranked elements.

This paper studies the effect of adapting the initial ranking before using the simple overlap removing technique. After introducing our laboratory setting, INEX (Section 2), and the overlap removal problem (Section 3), we analise relevance judgements to find useful dependencies (links) between elements in the XML tree and adapt the retrieval model to incorporate this information (Section 4). Section 5 demonstrates experimentally that this leads to better retrieval effectiveness. Finally, Section 6 summarises and discusses the main findings.

2 INEX and the focused retrieval task

The *Initiative for the Evaluation of XML retrieval* (INEX) [2] is a benchmark for the evaluation of XML retrieval. The collection provided to the participants is a subset of IEEE Computer Society publications, consisting of 16.819 scientific articles from 24 different journals.

Topics and relevance judgements. The participants are responsible for creating a set of topics (queries) and for assessing the relevant XML elements for each of these topics. The relevance judgements are given by two different dimensions: exhaustivity (E) and specificity (S). The exhaustivity dimension reflects the degree to which an element covers a topic and the specificity dimension reflects how focused the element is on that topic. Thus, to assess an XML element, participants are asked to highlight the relevant parts of that elements (specificity) and to use a three-level scale $[0, 1, 2]$ to define how much of the topic that element covers (exhaustivity). For later usage in the evaluation metrics, the specificity dimension is automatically translated to a value in a continuous scale $[0 \ldots 1]$, by calculating the fraction of highlighted (relevant) information contained by that element. The combination of the two dimensions is used to quantify the relevance of the XML elements. Thus, a highly relevant element is one that is both, highly exhaustive and highly specific to the topic of request.

The focused task. INEX has defined various XML retrieval scenarios, each corresponding to a specific task. This paper addresses the **focused** task, where the goal is to find the *most* exhaustive and specific elements on a *path*. Once the element is identified and returned, none of the remaining elements in the path should be returned. In other words, the result list should not contain overlapping elements. We choose to evaluate our results for *content-oriented XML retrieval using content-only conditions* (CO). Content-only requests are free text queries that contain only content conditions (without structural constraints). The retrieval system may retrieve relevant XML elements of varying granularity.

The evaluation metrics. INEX 2005 has evaluated retrieval results using the *Extended Cumulated Gain* (XCG) metrics. We realise that these measures are not yet widely spread in the general IR community, but we prefer to report our results using these measures for comparison to other groups participating at INEX. Here, we briefly outline their main characteristics, and refer to [3] for a more detailed description. The XCG metrics are an extension of the cumulated gain (CG) metrics [5] that consider dependency between XML elements (e.g., overlap and near-misses). The XCG metrics include a user-oriented measure called normalised extended cumulated gain (nxCG) and a system-oriented measure called effort-precision/gain-recall (ep/gr). In comparison to the common IR evaluation measures, *nxCG* corresponds to a precision measure at a fixed cut-off, and *ep/gr* provides a summary measure related to mean average precision (MAP). To model different user preferences, two different quantisation functions are used. The *strict* one models a user who only wants to see highly relevant elements ($E = 2$, $S = 1$) and the *generalised* one allows different degrees of relevance.

3 Removing Overlap

In an XML retrieval setting, to identify the most appropriate elements to return to the user is not an easy problem. IR systems have the difficult task to find out which are the most exhaustive and specific elements in the tree, and return only these to the user, producing result lists without overlapping elements. Current retrieval systems produce an initial ranking of all elements with their 'standard' XML retrieval model, and then remove overlapping elements from the result set. A fairly trivial approach keeps just the highest ranked element, and removes the other elements from the result list (e.g. [10]). More advanced techniques (e.g., [1], [7], [8]) exploit the XML tree structure to decide which elements should be removed or pushed down the ranked list.

In the first approach, the information retrieval systems rely completely on the underlying retrieval models to produce the best ranking. Thus, the assumption is that the most appropriate element in a path has been assigned a higher score than the rest. This could indeed be the case, if the retrieval model would consider, when ranking, not only the estimated relevance of the XML element itself but also its *usefulness* compared to other elements in the same path. However, since most retrieval models rank elements independently, the highest scored element may not be the best one for a focused retrieval task. We argue that retrieval models should take into account the dependency between elements to produce a good ranking for focused XML retrieval.

As an example, consider the following baseline models:

1) A retrieval model ($base_{LM}$) based on simple statistical language models [9, 4]. The estimated probability of relevance for an XML element E_j is calculated as follows:

$$P_{LM}(E_j) = \prod_{i=1}^{n} (\lambda P(T_i|E_j) + (1 - \lambda) P_{cf}(T_i)), \tag{1}$$

where:

$$P(T_i|E_j) = \frac{tf_{i,j}}{\sum_t tf_{t,j}} \text{ and } P_{cf}(T_i) = \frac{cf_i}{\sum_t cf_t},$$

2) The same retrieval model (LM) applying a length prior for length normalisation ($base_{LP}$). The probability of relevance is then estimated as:

$$P(E_j) = size(E_j) \;\; P_{LM}(E_j) \tag{2}$$

3) The same retrieval model (LM) but removing all the small elements (shorter than 30 terms) for length normalisation($base_{RM}$).

Using a $\lambda = 0.5$ and removing overlap with the simple algorithm of keeping the highest scored element in a path, we obtain the results shown in Table 1 for the three models described.

Table 1. Results for the different baselines runs in the focused task with strict (^{S}) and generalised (^{G}) quantisations

	nxCG[10]	**nxCG[25]**	**nxCG[50]**	**Maep**
$base^G_{LM}$	0.1621	0.1507	0.1557	0.0569
$base^G_{RM}$	**0.2189**	**0.2206**	**0.2100**	**0.0817**
$base^G_{LP}$	0.2128	0.1855	0.1855	0.0717
$base^S_{LM}$	**0.1016**	0.0855	**0.1207**	**0.0536**
$base^S_{RM}$	0.0610	**0.0974**	0.1176	0.0197
$base^S_{LP}$	0.0940	0.0910	0.1075	0.0503

In the generalised case, the length normalisation approaches help to improve the effectiveness of the system. This is because the original ranking contains many small elements that are ranked high but are not appropriate retrieval units. When applying length normalisation, other more lengthy units are pushed up the ranked list. These units tend to be more appropriate than the small ones, not only because longer elements contain more information but also due to the cumulative nature of the exhaustivity dimension. Since exhaustivity propagates up the tree, ancestors of a relevant element have an exhaustivity equal or greater than their descendants. These ancestors are relevant to some degree, even though their specificity may be low, i.e., even if they contain only a marginal portion of relevant text. Because far less large elements exist in a collection than small elements, researchers have found that it is worthwhile to return larger elements first [6], especially for INEX's *thorough* retrieval task (where systems are asked to identify all relevant elements, regardless their overlap). In the language modelling framework, this is achieved by introducing a length prior that rewards elements for their size.

In the focused task however, Table 1 shows that re-ranking the elements based on a length prior never results in the best retrieval results. This is explained by the contribution of the *specificity* dimension to the final relevance, which is captured best by the original language models. The elements pushed up the list by the length prior tend to be less specific, as they often cover more than one topic. In the generalised setting, where near-misses are allowed, removing the smallest elements is beneficial for retrieval effectiveness. In the strict case however, where near misses are not considered in the evaluation, the original ranking (without removing small elements) is the one that performs best. None of the three baseline models is satisfactory in all settings. Moreover, each of these models treat XML elements independently. We argue that in an XML retrieval setting, retrieval models should be aware of the structural relationships between elements in the tree structure. This is even more important when elements that are related through the tree hierarchy cannot all be presented to the user, as is the case in focused XML retrieval. In such a setting, the element's expected relevance of an element should depend on the expected relevance of its structurally related elements. If this structural information is already in the model, presenting a non-overlapping result list to the user becomes a presentation issue, performed by a simple post-filtering algorithm (which would work for any retrieval model). To achieve this goal, we analyse which are the relationships between retrieved elements and the most appropriate (highly relevant) elements and use this extra information to improve the initial ranking towards the task of focused retrieval.

4 Using structural relationships to improve focused retrieval

For the reasons described in previous section, we want to extend the retrieval model in a way that the appropriate units are rewarded and therefore they get a higher score than the rest. For that, we analyse the INEX 2005 relevance assessments (version 7.0) to find out which are the relationships between retrieved elements and the most appropriate (highly relevant) elements. Once these relationships (links) are created, we use this extra information to reinforce the estimated relevance of (hopefully) the most appropriate units.

4.1 Discovering links

To help the retrieval system to produce a proper ranking for the focused task, we need to learn what are the relationships between the retrieved XML elements in a baseline run and the elements identified in the INEX assessments as highly relevant for that topic. We consider that the most appropriate units to return to the user are those that are highly exhaustive ($E = 2$) and highly specific ($S >= 0.75$).

Our analysis is based on the top 1000 retrieved elements in a basic content-only run that uses a simple language modelling approach, which treats each element as a separate 'document' (described as $base_{LM}$ in Section 3). We study the occurrence of highly relevant elements in the direct vicinity of each retrieved element in the XML document. Since elements of different types (i.e., different tag names) are likely to show different patterns, we differentiate according to element type. In addition, we expect to observe different behaviour in front matter, body, and back matter. Figure 1 and 2 show results of an analysis of the ancestors of retrieved elements. The figure shows the probability for each level of finding the first highly relevant element when going up the tree from

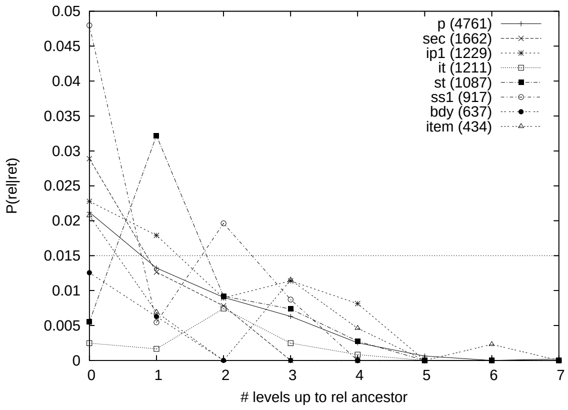

Fig. 1. Probability of finding the first highly relevant ancestor N levels up for the elements retrieved in the body

a retrieved element.[1] These graphs show for example that in the body part, retrieved `st` elements (section titles) are rarely relevant themselves, but their containing element, one level up, often is. The same holds for `fig` (figures) or `b` (bold) elements (not shown in the graph), while retrieved sections and subsections (`sec` and `ss1`) are mostly relevant themselves. For the retrieved elements in front and back matter (see Figure 2) it is generally needed to go more levels up to find the highly relevant elements, although elements such as `abs` (abstracts) and `p` (paragraphs) are mostly relevant themselves.

4.2 Using links

To use the possible relationships between retrieved and highly relevant elements, we need to define a propagation method that uses this *link* information to reward highly relevant elements from the information of the retrieved ones. We propose the following approach:

For each of the element types, we create a link from that element type to the two levels where the probability of finding a highly relevant element is higher (the two highest peaks for each type in the graphs in Figure 1 and 2). For instance, in the body part of an article, a `st` (section title) will point to the containing element (level 1) but also to the parent of this element (level 2). In a similar way, a `ss1` (subsection) element will have a link to the element located two levels up in the tree structure and another one to itself (level 0).

[1] To avoid very dense graphs, we only show the most frequently retrieved element types; also the lines with very low probabilities are left out.

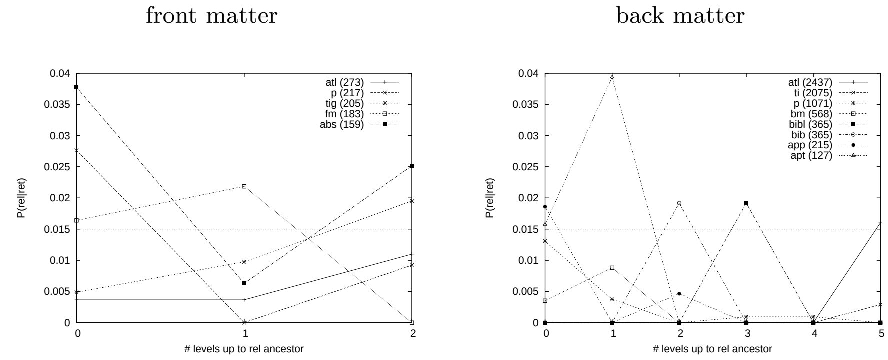

Fig. 2. Probability of finding the first highly relevant ancestor N levels up for elements retrieved in front matter and back matter

Since to use the two highest peaks of a distribution does not mean that the probability they represent is high, we define two types of links: the *strong* ones, where the probability that the element pointed at is highly relevant exceeds a threshold, and the *weak* ones, where this probability (even being the highest for that element) is lower than the threshold. For the analysis and experiments of this first paper, we set this threshold to 0.015 (shown in all figures of Section 4.1). As an example, a subset of the INEX collection with the discovered relations is shown in Figure 3.

Once the links and their types are defined, we need to define a model where the link information is used to propagate elements scores with the aim to reinforce the relevance of the most appropriate ones. We believe that the new score for any element in the XML document should be determined by a combination of the original score (estimated by the retrieval model) and the scores of the elements that point to it. Formally, we estimate the probability of relevance for an element in the following way:

$$P(E_j) = \alpha\ P_{LM}(E_j) + \beta\ aggr_{i \in sl(E_j)}(P_{LM}(i)) + \gamma\ aggr_{i \in wl(E_j)}(P_{LM}(i)), \quad (3)$$

where $P_{LM}(\cdot)$ is the score given to an element by the (baseline) language model; $sl(E_j)$ and $wl(E_j)$ are the sets of strong and weak links pointing to E_j; and $aggr$ is an aggregation function that combines the scores of all the elements that point to E_j. Note that some of the links discovered are self references (e.g., Section to Section). This means that for these nodes, the original retrieval model score contributes to the final score in two ways, once as the original score, with weight α and once in an aggregate of strong or weak links with weight β or γ.

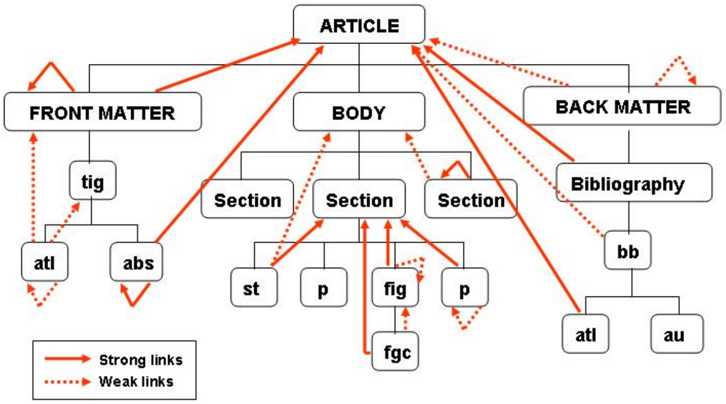

Fig. 3. Subset of article's structure with added links

5 Experiments

To evaluate the performance of our approach, we experimented with: (1) The values for the parameters of our new model (Equation 3), (2) the usage of two different aggregation functions (average and max), and (3) the individual and combined contribution of the different divisions of an article: front matter (FM), back matter(BM) and body (BDY). For each of the experiments, we report results with the official INEX metrics: nxCG at three different cut-off points (10, 25 and 50) and MAep, the uninterpolated mean average effort-precision. Our baseline model is the one described in Section 3 as $base_{LM}$.

5.1 Parametrisation values

The results of using different values for α, β, and γ in Equation 3 are shown in Table 2. For these experiments we use the link information from all the divisions of the article (FM, BM and BDY) and use the average as aggregation function.

Although we experimented with many more parameter combinations, only the most promising ones are shown. Performance tends to go down with higher α values. That means our model performs best when ranking the XML elements using only the *link* information. Combining this information with the original scores of the elements hurts the performance in both quantisations. The results show that most of the parameter combinations outperform the baseline run. The run that performs better is the one that uses only the *strong* links information (run2). Note that this run already outperforms all the models described in Section 3. The most surprising result of these experiments is the big improvement obtained under the strict quantisation for nxCG at 25 and 50. That indicates that the use of the link information helps indeed in finding the most highly relevant elements.

Table 2. Parametrisation values. Use of all *link* information (FM, BM and BDY) and average as aggregation function. Results for strict (^{S}) and generalised (^{G}) quantisations

	α	β	γ	**nxCG[10]**	**nxCG[25]**	**nxCG[50]**	**Maep**
$base^G_{LM}$	1	0	0	0.1621	0.1507	0.1557	0.0569
$run1^G$	0	0	1	0.1760	0.1370	0.1224	0.0375
$run2^G$	0	1	0	**0.2206**	**0.2213**	0.2235	**0.0798**
$run3^G$	0	0.9	0.1	0.2117	0.2170	0.2244	0.0783
$run4^G$	0	0.8	0.2	0.2106	0.2182	**0.2275**	0.0765
$run5^G$	0	0.7	0.3	0.2192	0.2182	0.2198	0.0746
$run6^G$	0	0.6	0.4	0.2100	0.1950	0.1998	0.0701
$run7^G$	0	0.5	0.5	0.2108	0.1926	0.2001	0.0690
$run8^G$	0.1	0.8	0.1	0.1980	0.1955	0.2055	0.0719
$run9^G$	0.1	0.7	0.2	0.2029	0.1909	0.1989	0.0700
$run10^G$	0.1	0.6	0.3	0.1999	0.1756	0.1837	0.0670
$base^S_{LM}$	1	0	0	0.1016	0.0885	0.1207	0.0536
$run1^S$	0	0	1	0.0577	0.0723	0.0796	0.0428
$run2^S$	0	1	0	**0.1154**	0.1561	**0.1696**	**0.0670**
$run3^S$	0	0.9	0.1	0.1077	**0.1577**	0.1585	0.0662
$run4^S$	0	0.8	0.2	0.0923	0.1500	0.1523	0.0563
$run5^S$	0	0.7	0.3	0.0962	0.1500	0.1515	0.0556
$run6^S$	0	0.6	0.4	0.0885	0.1207	0.1381	0.0544
$run7^S$	0	0.5	0.5	0.0846	0.1191	0.1396	0.0531
$run8^S$	0.1	0.8	0.1	0.1055	0.1570	0.1650	0.0597
$run9^S$	0.1	0.7	0.2	0.1093	0.1570	0.1650	0.0590
$run10^S$	0.1	0.6	0.3	0.1093	0.1292	0.1474	0.0582

5.2 Aggregation functions

We experimented with two different aggregation functions: the average and the max. The average rewards the elements that have all of their inlinks relevant and punishes the ones that are pointed to also by irrelevant elements, while the max rewards the elements if they contain at least, one relevant element pointing to them, regardless of the other inlinks. We would expect that the average works well for links such as a paragraph to section, since, intuitively, a section is relevant if most of its paragraphs are. The max would work better for other types of links such as section title to section, where having only one of the inlinks relevant might already be a good indicator that the element is relevant. For these experiments we use the link information from the body part of the article (BDY) and several parametrization values. Results are shown in Table 3.

For most cases under the generalised quantisation the max operator outperforms the average. This means that the links pointing to an element are good indicators of relevance, regardless the number or relevance of other links pointing to that element. However, we can see that for nxCG at 25 and 50 under the strict quantisation, the average performs much better than the max. Which might indicate that the highly relevant elements are those that have many relevant links pointing to them.

Table 3. Aggregation functions. Use of *link* information in the body part of the articles and MAX or AVG as aggregation functions. Evaluation with strict (^{S}) and generalised (^{G}) quantisations

	α	β	γ	**nxCG[10]**		**nxCG[25]**		**nxCG[50]**		**Maep**	
				MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG
$run2^G$	0	1	0	0.2301	**0.2247**	0.2219	**0.2247**	0.2213	**0.2248**	**0.0805**	**0.0825**
$run4^G$	0	0.8	0.2	0.2350	0.2180	0.2246	0.2218	**0.2235**	0.2216	0.0801	0.0780
$run6^G$	0	0.6	0.4	**0.2404**	0.2059	**0.2267**	0.2076	0.2206	0.2093	0.0802	0.0746
$run9^G$	0.1	0.7	0.2	0.2138	0.2013	0.1964	0.1856	0.2078	0.1960	0.0720	0.0700
$run2^S$	0	1	0	**0.1173**	**0.1288**	0.1531	**0.1854**	0.1718	**0.1993**	**0.0761**	**0.0712**
$run4^S$	0	0.8	0.2	0.1000	0.0923	0.1469	0.1732	0.1676	0.1901	0.0680	0.0592
$run6^S$	0	0.6	0.4	0.0962	0.0846	0.1469	0.1716	0.1555	0.1884	0.0648	0.0587
$run9^S$	0.1	0.7	0.2	0.1170	0.1055	**0.1616**	0.1555	**0.1735**	0.1666	0.0694	0.0600

5.3 Article's divisions contribution

We also analysed which of the divisions of an article contributes more to the gain of performance obtained by our approach. For that, we use the link information from each of the divisions independently and also combined. We use the max as aggregation function and two of the parametrisation value combinations. The results of these runs are shown in Table 4.

As expected, the only division that performs well on its own is the body part of the articles. To use only the links of the front and back matter hurts considerably the performance of the model. That is because with the parametrisation used, the original scores of the elements are cancelled out. Effectively, this means that elements without inlinks are removed from the result lists. When using a higher α the scores would be much better but probably not reaching the baseline model ones. For both runs and in both quantisations, the best combination is to use the front matter and the body divisions. This means that, while on its own body is the only effective document part, the links contained in the front matter are valuable. Either they point to relevant elements in the front matter itself (such as abstracts) or they help the retrieval model to give higher scores to the relevant articles. The back matter information hurts the performance of the system. A possible cause for this is that the information contained in the back matter should not be propagated up the tree but to the elements that refer to it. Further experimentation needs to be done to test this hypothesis.

6 Discussion

We presented an analysis of links between retrieved and relevant elements and used the findings to improve retrieval effectiveness in the focused retrieval task of INEX 2005. Our approach outperforms the baselines presented in all settings. Under the strict quantisation, this improvement is considerably big, indicating that the links discovered are very good pointers to highly relevant information. Perhaps our most striking finding is that the original score of an element is not

Table 4. Main article divisions: FM (front matter), BM (back matter) and BDY (body). Results using MAX as aggregation function. Evaluation with strict (^{S}) and generalised (^{G}) quantisations

	Divisions used	α	β	γ	**nxCG[10]**	**nxCG[25]**	**nxCG[50]**	**Maep**
$base^G_{LM}$		1	0	0	0.1621	0.1507	0.1557	0.0569
$run2^G$	BM	0	1	0	0.0881	0.0662	0.0619	0.0150
$run2^G$	FM	0	1	0	0.1124	0.0826	0.0653	0.0197
$run2^G$	BDY	0	1	0	0.2301	0.2219	0.2213	0.0805
$run2^G$	BM+BDY	0	1	0	0.2123	0.1950	0.2018	0.0722
$run2^G$	FM+BDY	0	1	0	**0.2498**	**0.2413**	**0.2406**	**0.0898**
$run2^G$	BM+FM+BDY	0	1	0	0.2123	0.2003	0.2145	0.0745
$run4^G$	BM	0	0.8	0.2	0.1068	0.0760	0.0701	0.0177
$run4^G$	FM	1	0.8	0.2	0.0882	0.0713	0.0527	0.0144
$run4^G$	BDY	1	0.8	0.2	0.2350	0.2246	0.2235	0.0801
$run4^G$	BM+BDY	0	0.8	0.2	0.2229	0.1966	0.2003	0.0709
$run4^G$	FM+BDY	0	0.8	0.2	**0.2421**	**0.2366**	**0.2321**	**0.0835**
$run4^G$	BM+FM+BDY	0	0.8	0.2	0.2286	0.1981	0.2076	0.0724
$base^S_{LM}$		1	0	0	0.1016	0.0885	0.1207	0.0536
$run2^S$	BM	0	1	0	0.0000	0.0015	0.0010	0.0002
$run2^S$	FM	0	1	0	0.0170	0.0147	0.0303	0.0113
$run2^S$	BDY	0	1	0	0.1173	0.1531	0.1718	0.0761
$run2^S$	BM+BDY	0	1	0	0.1000	0.1268	0.1363	0.0649
$run2^S$	FM+BDY	0	1	0	**0.1327**	**0.1902**	**0.2134**	**0.0810**
$run2^S$	BM+FM+BDY	0	1	0	0.1077	0.1378	0.1566	0.0688
$run4^S$	BM	0	0.8	0.2	0.0077	0.0031	0.0026	0.0010
$run4^S$	FM	0	0.8	0.2	0.0115	0.0146	0.0180	0.0086
$run4^S$	BDY	0	0.8	0.2	0.1000	0.1469	0.1676	**0.0680**
$run4^S$	BM+BDY	0	0.8	0.2	0.0962	0.1191	0.1178	0.0613
$run4^S$	FM+BDY	0	0.8	0.2	**0.1038**	**0.1563**	**0.1743**	0.0674
$run4^S$	BM+FM+BDY	0	0.8	0.2	0.0962	0.1246	0.1271	0.0609

a good indicator for the element's relevance. Ignoring that score and replacing it with a score based on the retrieval model scores of the elements that link to the element improves results significantly. Note that some of these links may be self links, which means that for some element types the original score *is* taken into account. We also showed that using the maximum score of all linked elements gives better results than taking an average. This indicates that a single good linked element can already indicates the merit of the element at hand. Furthermore, links from the document body turned out to be most valuable, but also front matter links contribute to the results. Back matter links are less valuable. Perhaps information from the back matter should be propagated to the elements that refer to it rather than up the tree, but this is subject to further study. Since, in the current paper, we performed the analysis and experiments in the same collection and topic set (INEX 2005), there is a big risk of overfitting. Still, we believe most of the links discovered are intuitive (e.g., section title to section or abstract to article) and therefore likely to be topic independent and recurring in other collections. If relevance assessments are not available, the discovered relationship information could be obtained from a person familiar with the XML structure of the collection (e.g. publisher) or probably by analysing clickthrough data. In any case, we showed that the structure in an XML tree can contain valuable information and therefore XML elements should not be treated independently.

References

1. Charles L. A. Clarke. Controlling Overlap in Content-Oriented XML Retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 314–321, New York, NY, USA, 2005. ACM Press.
2. Norbert Fuhr, Norbert Gövert, Gabriella Kazai, and Mounia Lalmas. INEX: INitiative for the Evaluation of XML Retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, 2002.
3. Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Gabriella Kazai, editors. *Advances in XML Information Retrieval. Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, volume 3977 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.
4. Djoerd Hiemstra. A Linguistically Motivated Probabilistic Model of Information Retrieval. In C. Nicolaou and C. Stephanidis, editors, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 513 of *Lecture Notes in Computer Science*, pages 569–584. Springer-Verlag, 1998.
5. Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.
6. Jaap Kamps, Maarten de Rijke, and Börkur Sigurbjörnsson. Length Normalization in XML Retrieval. In *SIGIR '04: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, pages 80–87. ACM Press, 2004.
7. Yosi Mass and Matan Mandelbrod. Experimenting Various User Models for XML Retrieval. In N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors, *INEX 2005 Workshop Proceedings*, Dagstuhl, Germany, 2005.
8. Vojkan Mihajlović, Georgina Ramírez, Thijs Westerveld, Djoerd Hiemstra, Henk Ernst Blok, and Arjen P. de Vries. TIJAH Scratches INEX 2005: Vague Element Selection, Image Search, Overlap and Relevance Feedback. In N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors, *INEX 2005 Workshop Proceedings*, Dagstuhl, Germany, 2005.
9. Jay M. Ponte and W. Bruce Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press.
10. Karen Sauvagnat, Lobna Hlaoua, and Mohand Boughanem. XFIRM at INEX 2005: ad-hoc, heterogenous and relevance feedback tracks. In N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors, *INEX 2005 Workshop Proceedings*, Dagstuhl, Germany, 2005.