Using Topic Shifts for Focussed Access to XML Repositories

Elham Ashoori and Mounia Lalmas

Queen Mary, University of London
London, E1 4NS, UK
{elham,mounia}@dcs.qmul.ac.uk

Abstract. In focussed XML retrieval, a retrieval unit is an XML element that not only contains information relevant to a user query, but also is specific to the query. INEX defines a relevant element to be at the right level of granularity if it is exhaustive and specific to the user's request – i.e., it discusses fully the topic requested in the user's query and no other topics. The exhaustivity and specificity dimensions are both expressed in terms of the "quantity" of topics discussed within each element. We therefore propose to use the number of topic shifts in an XML element, to express the "quantity" of topics discussed in an element as a mean to capture specificity. We experimented with a number of element-specific smoothing methods within the language modelling framework. These methods enable us to adjust the amount of smoothing required for each XML element depending on its number of topic shifts, to capture specificity. Using the number of topic shifts combined with element length improves retrieval effectiveness, thus indicating that the number of topic shifts is a useful evidence in focussed XML retrieval.

1 Introduction

Content-oriented XML[1] retrieval systems aim at supporting more precise access to XML repositories by retrieving XML document components (the so-called XML elements) instead of whole documents in response to users' queries. Therefore, in principle, XML elements of any granularity (for example a paragraph or the section enclosing it) are potential answers to a query, as long as they are relevant. However, the child element (paragraph) may be more focussed on the topic than its parent element (the section), which may contain additional irrelevant content. In this case, the child element is a better element to retrieve than its parent element, because not only it is relevant to the query, but it is also specific to the query. Thus the aim of an XML retrieval system is to provide a *focussed access* to XML repositories by returning the most appropriate units of retrieval for a given query.

To identify what constitutes a most appropriate unit of retrieval, INEX, the initiative evaluation for XML retrieval (e.g., [6]) defined relevance in terms of

[1] XML stands for eXtensible Markup Language – see http://www.w3.org/

G. Amati, C. Carpineto, and G. Romano (Eds.): ECIR 2007, LNCS 4425, pp. 444–455, 2007.

two dimensions, *exhaustivity* and *specificity*, each defined on a graded scale. These two dimensions are respectively defined as "how exhaustively an element discusses the topic of request" and "how focussed an element is on the topic of request (i.e., discusses no other irrelevant topics)" [5]. The combination of these two relevance dimensions is used to identify relevant elements that are both exhaustive and specific to the user's query.

In IR, the main factors that affect the importance of a term in a text – in terms of how good it is at representing the content of the text – are the term frequency, the inverse document frequency, and the document length. To incorporate the "specificity" dimension in XML retrieval, we propose to exploit the "number of topic shifts" as another factor affecting the importance of a term in a text. We incorporate the "number of topic shifts" in the smoothing process within the language modelling framework. In the language modelling approach, smoothing refers to adjusting the maximum likelihood estimator for the element language model so as to correct inaccuracy arising from data sparseness. In the smoothing process, the probability of terms seen in an element are discounted mainly by combining the element language model with the collection language model, thus assigning a non-zero probability to the unseen terms.

We use "Dirichlet smoothing" approach [14] as the smoothing framework. Dirichlet smoothing is one of the popular document-dependent smoothing methods. Using this approach in XML retrieval enables us to adjust the amount of smoothing dynamically by the features of elements (e.g. length, topic shifts). In this work, we explore various ways to incorporate topic shifts in the smoothing process, either individually or combined with the length of XML elements. We investigate whether using topic shifts individually, or combined with length provides a better smoothing approach for the focussed access to XML documents.

The paper is organised as follows. Section 2 discusses related work. In section 3, we define the notion of topic shifts and how we calculate it. In section 4, we describe the language modelling formalism used to perform focussed retrieval. Section 5 describes the methodology used to compare the proposed topic shifts-based smoothing process, including the INEX test collection used to carry out this investigation. The experiments and results are discussed in Section 6. Section 7 concludes the paper, with some thoughts for future work.

2 Related Work

The language modeling approach to IR is a sound and flexible framework, not only in content-oriented XML retrieval (e.g., [9,13,12]), but also for IR research in general [3]. The basic idea of this approach is to estimate a language model for each document, and rank documents with respect to the likelihood that the query can be generated from the estimated language models. Retrieval performance of language modelling approaches have been shown to be sensitive to the smoothing parameters both in ad hoc IR [14] and in XML retrieval [9]. Although smoothing is essential in language modelling due to data sparseness, Zhai et al. [14] introduced another role for it. They showed that the effects of

smoothing is very sensitive to the type of queries (long, short) which results in a new role for smoothing, query modelling, to "explain the common and non-informative words in a query". Following the query modelling role of smoothing, Hiemstra [8] introduced the term-specific smoothing and used feedback documents for estimating term-specific smoothing parameters.

In the context of ad hoc XML element retrieval, Kamps et al. [9] used a multinomial language model, with Jelinek-Mercer smoothing which is a linear interpolation of the element language model and the collection model. In this approach the smoothing parameter is fixed for all elements. They showed that the smoothing parameter indirectly introduces a length bias by increasing the importance of the presence of query terms in the retrieved elements. In this approach, the optimal amount of smoothing depends on the relevance assessments. If during assessment, the assessors favor the longer elements in the collection, little smoothing is required. More precisely, it was shown that a high amount of smoothing leads to the retrieval of shorter elements. This work is different from ours as we propose an element-dependent smoothing approach, where the amount of smoothing can be dynamically adjusted according to features of elements (in this paper, length, topic shifts). This allows us to investigate the effect of topic shifts (and element length) by making them "parameters" of the language modeling framework[2].

3 Topic Shifts

In this section, we describe how we measure the number of topic shifts of the elements forming a XML document. We use the number of topic shifts in an XML element, to express the "quantity" of topics discussed in an element. For this purpose, both the logical structure and a semantic decomposition of the XML document are needed. Whereas the logical structure of XML documents is readily available through their XML markup, their semantic decomposition needs to be extracted. To achieve that, we apply a topic segmentation algorithm based on lexical cohesion, TextTiling[3] [7], which has been successfully used in several IR applications (e.g., [2]). The underlying assumption of topic segmentation algorithms based on lexical cohesion is that a change in vocabulary signifies that a topic shift occurs. This results in topic shifts being detected by examining the lexical similarity of adjacent text segments.

TextTiling is a linear segmentation algorithm that considers the discourse unit to correspond to a paragraph, and therefore subdivides the text into multi-paragraph segments. TextTiling is performed in three steps. In the first step, after performing tokenization, the text is divided into pseudo-sentences of size W, called token-sequences. Next, these token-sequences are grouped together into blocks of size K. The gap between two adjacent blocks constitutes a potential

[2] This work is also different from ours, as the former is concerned with the task of estimating the relevance of XML elements, and not the focussed access to XML elements (see Section 5).

[3] http://elib.cs.berkeley.edu/src/texttiles/

boundary for a semantic segment. To identify the actual boundaries, a depth score is computed for each potential boundary, by using the similarity scores assigned to the neighbouring gaps between blocks, and by applying a smoothing process. The algorithm determines the number of segments, referred to as tiles, by considering segment boundaries to correspond to gaps with depth scores above a certain threshold. The detected boundaries are then adjusted to correspond to the actual discourse unit breaks, i.e., the paragraph breaks.

The semantic decomposition of an XML document is used as a basis to calculate the number of topic shifts in each XML element forming that document. We consider that a topic shift occurs (i) when one segment ends and another segment starts, or (ii) when the starting (ending) point of an XML element coincides with the starting (ending) point of a semantic segment.

The *number of topic shifts* in an XML element e in document d is defined as:

$$actual\_topic\_shifts(e, d) + 1 \tag{1}$$

where $actual\_topic\_shifts(e, d)$ are the actual occurrences of topic shifts in element e of document d. We are adding 1 to avoid zero values. With the above definition, the larger the number of topic shifts, the more topics are discussed in the element, which would indicate that the content of element is less focussed with respect to the overall topic discussed in the element. By considering the number of topic shifts occurring in an element instead of the number of topics discussed (in our case modelled as the number of tiles), we are able to distinguish the cases where the topic shift occurs not within the actual content of an element, but at its boundaries.

4 Element-Specific Smoothing Using Topic Shifts

Since XML elements of any granularity are potential answers to a query, we estimate a language model for each XML element in the collection. The element language model is smoothed using a Dirichlet prior [14] with the collection language model as the reference model.

If we estimate a language model for each element, then the relevance of an element e to a given query q is computed as how likely the query can be generated from the language model for that element. We rank elements based on the likelihood for a query $q = (t_1, t_2, ..., t_n)$ to be generated from an element e as:

$$P(t_1, .., t_n|e) = \prod_{i=1}^{n} \left(\frac{c(t_i, e) + \mu P(t_i|C)}{\mu + |e|}\right) \tag{2}$$

$$= \prod_{i=1}^{n} \left(\left(1 - \frac{\mu}{\mu + |e|}\right)\frac{c(t_i, e)}{|e|} + \frac{\mu}{\mu + |e|}P(t_i|C)\right) \tag{3}$$

$$= \prod_{i=1}^{n} \left((1 - \alpha_e)P_{ml}(t_i|e) + \alpha_e P(t_i|C)\right) \tag{4}$$

where

- t_i is a query term in q,
- $c(t_i, e)$ is the number of occurrences of the query term t_i in element e,
- μ is a constant,
- $|e|$ is the number of terms in element e,
- $P_{ml}(t_i|e) = \frac{c(t_i,e)}{|e|}$ is the probability of observing term t_i in element e, estimated using the maximum likelihood estimation,
- $P(t_i|C) = \frac{ef(t_i)}{\sum_t ef(t)}$ is the probability of observing query term t_i in the collection where $ef(t)$ is the number of XML elements in which the term t occurs.
- $\alpha_e = \frac{\mu}{\mu+|e|}$ is an element-dependent constant which is related to how much probability mass will be allocated to unseen query terms, i.e., the amount of smoothing.

Since the maximum likelihood estimator will generally underestimate the probability of any term unseen in the element, the main purpose of smoothing is to improve the accuracy of the term probability estimation. If we are concerned with the exhaustivity dimension of relevance, then we may expect that most of the query terms to appear in an element for the element to be retrieved. In this case, one would expect that the term probability estimates are more reliable for long elements as they contain more terms compared to the short elements. Therefore, a shorter element needs to be more smoothed with the collection model compared to a longer element. This shows that a higher value of α_e is needed to capture exhaustivity in small elements. Smoothing with Dirichlet prior (Equation 4) satisfies this requirement as the value of α_e depends on the length of the elements.

The above smoothing process is reasonable if we are not concerned with the specificity dimension. In INEX, specificity is automatically measured by calculating the ratio of the relevant content in an XML element (see Section 5). This implies that unseen terms are less of an issue for small elements compared to the above case. Therefore a smaller amount of smoothing (a lower value of α_e) is needed to capture specificity in small elements than the amount of smoothing required to capture exhaustivity. Due to this contradictory behaviour in the required amount of smoothing – if we want to capture both exhaustivity and specificity – Equation 4 in its current version cannot be used to capture both relevance dimensions if only length is taken into account.

To accommodate for the specificity dimension, we propose to set α_e, the amount of smoothing, to be proportional to the number of topic shifts in the element. The idea of incorporating topic shifts in this manner originates from the fact that if the number of topic shifts in an element is low and an element is relevant, then it is likely to contain less non-relevant information compared to the case where a high number of topic shifts exists.

It might be argued that in general, when the length of an element increases, it is highly likely that it will discuss more topics. However, this is not always the case, as it was shown in [1], where the number of topic shifts of parent

elements was compared to that of their children. Even though the length from children to their parents increases, the number of topic shifts in the majority of cases stays the same, i.e. it does not vary when the length increases. As topic shifts and length are two distinct evidences [1], we explore several ways to compute the element-dependent constant α_e (amount of smoothing) in Equation 4 as a function of its length, its number of topic shifts and a combination of both. We replace length by the number of topic shifts in the original formula to compare how these two retrieval settings are useful to capture exhaustivity. Next, we replace the length with the inverse of length and inverse of topic shifts to capture specificity. Finally we combine length and topic shifts in a retrieval setting to capture both exhaustivity and specificity. We, thus, experiment with five different retrieval approaches:

1. $\alpha_e = \frac{\mu}{\mu+|e|}$ implies that longer elements need less smoothing. This approach is the original Bayesian smoothing with Dirichlet priors. We refer to this approach as our baseline approach (L).
2. $\alpha_e = \frac{\mu}{\mu+1/|e|}$ implies that shorter elements need less smoothing. This means that the presence of a query term in an element is rewarded if the number of terms in the element is small. We refer to this approach as (1/L).
3. $\alpha_e = \frac{\mu}{\mu+|T|}$ implies that elements with a high number of topic shifts need less smoothing. We refer to this approach as (T).
4. $\alpha_e = \frac{\mu}{\mu+1/|T|}$ implies that elements with a lower number of topic shifts need less smoothing. This means that the presence of a query term in an element is rewarded if the number of topic shifts in the element is low. We refer to this approach as (1/T).
5. $\alpha_e = \frac{\mu}{\mu+\frac{|e|}{|T|}}$ implies that elements should be smoothed based on the average number of terms per topic shifts of element. This is an approximation of the average number of terms per topic in an XML element. In this way, we return back to the normal interpretation of smoothing in Equation 4 but we consider a more refined version of length. In this case we differentiate between two elements with equal length and different numbers of topic shifts so that the presence of a query term in element with a lower number of topic shifts is rewarded. We refer to this approach as (L/T).

5 Methodology

In our work, we use the INEX-2005 test collection. The INEX collection, *Version 1.8*, contains 16,819 scientific articles from 24 IEEE Computer Society journals, marked up in XML, consisting of over 10 million elements of varying length.

We use the title field of the 29 content-only (CO) topics of *Version 2005-003* of the INEX 2005 data set[4]. CO topics are requests that ignore the document structure and contain only content related conditions, and at this stage of our work, are sufficient for our investigation. We evaluate our approaches against

[4] In INEX 2005, these topics are referred to as CO+S.

the relevance assessments *Version adhoc2005-assessments-v7*. For INEX 2005, exhaustivity is measured on a *3 + 1*-point scale: highly exhaustive (e=2), somewhat exhaustive (e=1), not exhaustive (e=0) and too small (e=?). In this work, we ignore too small elements. The specificity dimension is automatically measured on a continuous scale [0,1], by calculating the ratio of the relevant content of an XML element after the assessors highlighted text fragments containing only relevant information.

The official evaluation metrics employed in INEX 2005 are the eXtended Cumulated Gain (XCG) metrics [10], which include the user-oriented measures of extended cumulated gain ($nxCG[i]$) and the system-oriented effort-precision/gain-recall measures ($MAep$). For a given rank i, $nxCG[i]$ reflects the relative gain the user accumulated up to that rank, compared to the gain he/she could have attained if the system would have produced the optimum best ranking.

The effort-precision *ep* at a given gain-recall value *gr* is defined as the number of visited ranks required to reach a given level of gain relative to the total gain that can be obtained. The non-interpolated mean average effort-precision, $MAep$, is calculated by averaging the effort-precision values measured at natural recall-point, i.e., whenever a relevant XML element is found in the ranking.

INEX employs quantization functions to combine the two graded relevance dimensions, by providing a relative ordering of the various combinations of e-s values and a mapping of these to a single relevance scale in [0, 1], as required by the XCG metrics. In INEX 2005, two quantization functions were used:

$$f_{strict}(e,s) := \begin{cases} 1 \text{ if } e = 2 \text{ and } s = 1, \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

$$f_{generalised}(e,s) := \begin{cases} e * s \text{ if } e \in \{1,2\}, \\ 0 \quad \text{otherwise.} \end{cases} \tag{6}$$

The strict quantization function f_{strict} is used to evaluate retrieval methods with respect to their capability of retrieving highly exhaustive and highly specific elements (e=2, s=1). The generalised quantization function $f_{generalised}$ credits elements according to their degree of relevance, hence allowing modelling varying levels of user satisfaction gained from not fully specific and highly exhaustive elements, i.e., less relevant elements.

The retrieval task addressed in this work is focussed XML retrieval. INEX has defined various XML retrieval scenarios, each corresponding to a specific task. In the *focussed* task, the aim is for XML retrieval systems to return to the user a non-overlapping ranked list of the most exhaustive and specific elements on a relevant path[5]. The five approaches described in the previous section will rank elements, for example, considering the number of topic shifts. They will, however, not produce an overlap-free ranking. There are sophisticated ways to remove overlapping elements (e.g., [11]). In this work we restrict ourselves to a

[5] A relevant path is a path within the XML tree of a given XML document, whose root node is the root element and whose leaf node is a relevant element that has no or only irrelevant descendants.

post-filtering on the retrieved ranked list by selecting the highest scored element from each of the paths, as our main interest here is to investigate how the proposed smoothing approaches can help retrieval effectiveness.

To calculate the number of topic shifts in each XML element, our first step is to decompose the INEX XML documents into semantic segments through the application of TextTiling. We consider the discourse units in TextTiling to correspond to *paragraph* XML elements. We considered paragraph elements to be the lowest possible level of granularity of a retrieval unit. Although this can be viewed as collection-dependent and might change from one collection to the next, it is likely that for many XML content-oriented collections, meaningful content will occur mainly at paragraph level and above.

We set the TextTiling parameters to $W = 10$ and $K = 6$, which is based on a heuristic setting $W * K$ to be equal to the average paragraph length (in terms of the number of terms) [7]. After the application of TextTiling, we compute the number of topic shifts in elements.

6 Experiments and Results

In this section, we report on the experiments, and their results, that were carried out in order to investigate the effects of topic shifts in the smoothing process. We experiment with a wide range for μ between [0, 20000] to study the behaviour of each individual retrieval approach. To compare the five smoothing approaches (*L, 1/L, T, 1/T, L/T*), we select a best run (in terms of MAep) for each approach and then compare the behaviour of these best runs based on nxCG.

For each of the approaches, the top 1500 ranked elements are returned as answers for each of the CO topics. For the user-oriented evaluation, we report nxCG at three different early cut-off points (10, 25, 50). In addition, the nxCG graphs for both the full rank and the early rank levels are given. For the system-oriented evaluation, MAep is reported. For both evaluations, both strict and generalised quantization functions are used.

To determine whether the differences in performance between two approaches are statistically significant, we use the bootstrapping significance testing [4]. Improvements at confidence levels 95% and 99% over the baseline are respectively marked with + and ++. Similarly, decreases in performance at confidence level of 95% and 99% are marked with – and ––.

Table 1 shows a summary of the results. This table presents, for each quantization function, the results for both the user- and the system-oriented evaluation of the five retrieval approaches, with the L approach acting as the baseline.

We first consider the results in terms of mean average effort precision (MAep), shown in Figure 1 and the last column of Table 1.

Under the *generalised quantization function*, the MAep ranks L/T approach followed by L above the other approaches reflecting that on average the user needs to spend less effort when scanning the output of L/T to achieve the same level of gain. However the difference is not significant. To obtain a better understanding we look at the performance at different values for parameter μ.

Table 1. Focussed retrieval task: MAep and nxCG at different cut-off points considering L as baseline

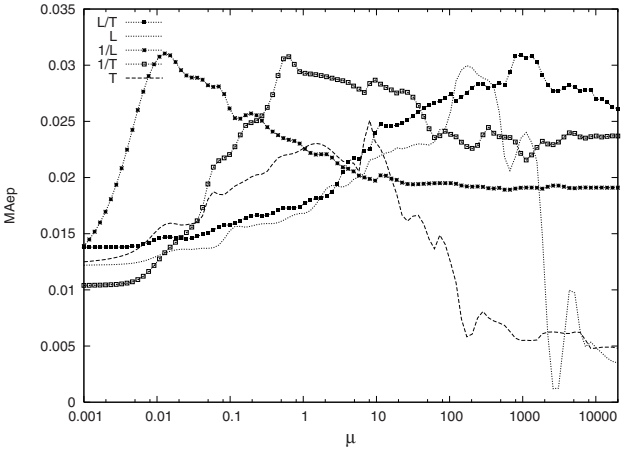
Approach	μ	nxCG@10	nxCG@25	nxCG@50	MAep
			General		
L	$\mu = 256$	0.2634	0.2387	0.2258	0.0938
$1/L$	$\mu = 0.01$	0.2333(-11%)	0.2358(-1.2%)	0.2198(-2.5%)	0.0911(-2.9%)
T	$\mu = 7$	**0.2638**(0.1%)	0.2391(1.6%)	0.2187(-3.1%)	0.0899(%-4.2)
$1/T$	$\mu = 0.15$	0.2388(-9.3%)	0.2297(-3.7%)	0.2138(-5.3%)	0.0894(-4.7%)
L/T	$\mu = 448$	0.2603(-1.1%)	**0.2506(4.9%)**	**0.2424(7.4%)**	**0.0992**(5.8%)
			Strict		
L	$\mu = 256$	0.069	0.1295	0.1351	0.029
$1/L$	$\mu = 0.01$	**0.0863**(25%)	0.15(15.8%)	0.1513(12%)	0.0305(5.2%)
T	$\mu = 8$	0.069(0%)	0.1333(2.9%)	0.1411(4.4%)	0.0251(-13.4%)
$1/T$	$\mu = 0.7$	**0.0863(25%)**	**0.1515(17%)**	0.1688(25%)	0.0304(4.8%)
L/T	$\mu = 1280$	0.0687(-0.4%)	0.1446(11.7%)	**0.1843(36.4%)(++)**	**0.0308(6.2%)**

Figure 1(b) shows the mean average effort precision for μ between [0, 20000]. We observe that L/T approach shows better performance than L regardless of the values of μ, which indicates that elements with equal length and smaller number of topic shifts require less smoothing. This is due to the fact that in the L/T approach, the presence of a query term in an element with a lower number of topic shifts is rewarded.

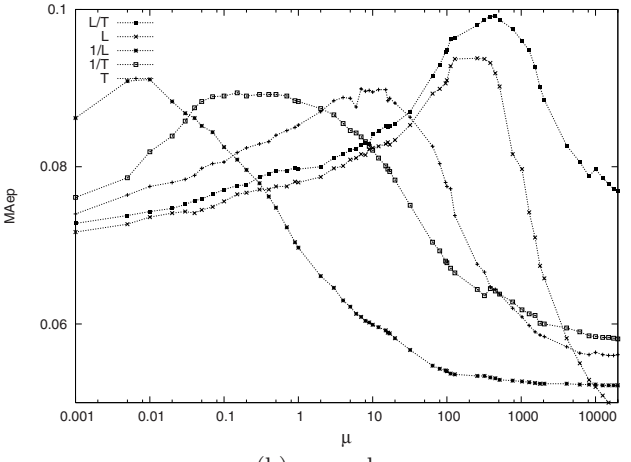
Under the *strict quantization function*, results show that L/T, $1/L$, and $1/T$ are the most effective approaches with almost the same MAep, whereas T did not perform particularly well. The $1/T$ approach considers less smoothing for elements with fewer number of topic shifts represented by the lower value for α_e. The $1/L$ approach is in the opposite direction of the standard Dirichlet smoothing and considers more smoothing for the larger elements. These results support the argument in Section 4 in which we suggested that the Dirichlet smoothing in its standard formulation is not sufficient to satisfy the "specificity" dimension of relevance. These results show that for retrieving highly specific and highly exhaustive elements, in the strict case, less smoothing is required for elements that are either small or contain fewer number of topic shifts than those that are longer or contain a higher number of topic shifts. Similar to the observed behaviour for the generalised quantization function, L/T shows better performance than L in most of the values of μ.

Overall, the L/T approach, where the number of topic shifts combined with length affects the amount of smoothing, performs better than other retrieval approaches when evaluated using the system-oriented measures.

Next, we focus on the *user-oriented evaluation* and discuss the results obtained using the nxCG measure, shown in Figure 2 and Table 1. Under the *generalised quantization function*, the baseline approach, L, shows better performance at the very early ranks (1% ranks) (see Figure 2(d)). For the other rank positions, the combination of length and topic shifts (L/T) is the most effective approach (see Figure 2(c)). These approaches are useful to satisfy a user who also gains from less relevant elements, i.e., not fully specific and highly exhaustive elements. However, for retrieving highly specific and highly exhaustive elements, the strict case, both $1/L$ and $1/T$ approaches improve the results at the early cut-



(a) strict

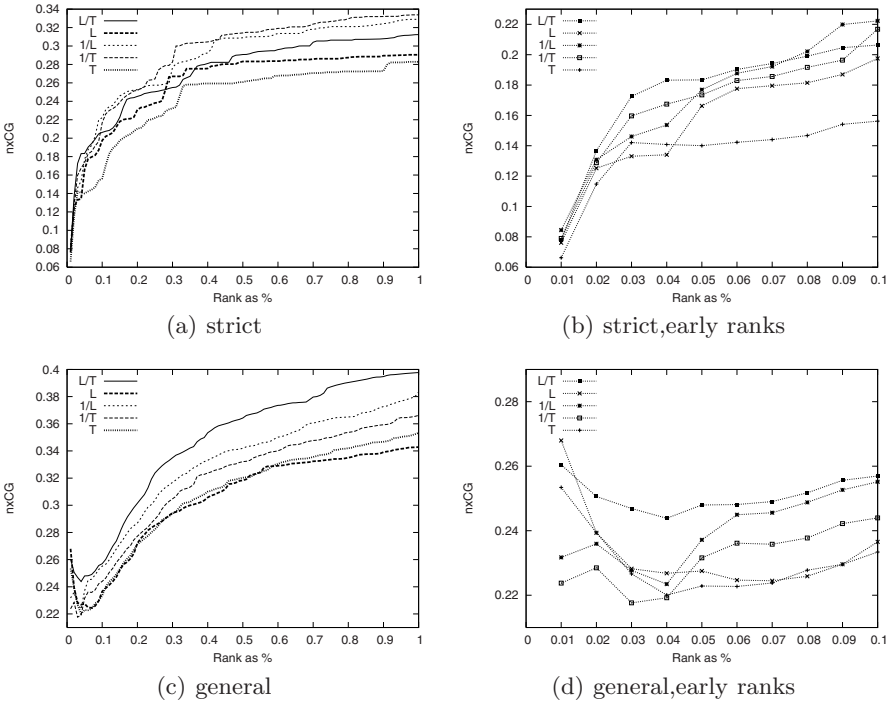


(b) general

Fig. 1. Mean Average effort precision (MAep) for different element-specific smoothing approaches

off points 10 and 25, where we get (+25%,17.8%) and (25%,15%) improvements, respectively (see Table 1). Similar to the system-oriented evaluation, these results again support our argument in Section 4, i.e. the smoothing process should be treated differently depending on the relevance dimensions.

To conclude, the use of the number of topic shifts led to improvement of performance particularly when combined with element length, thus indicating that the number of topic shifts is a useful evidence in focussed XML retrieval.



(a) strict (b) strict,early ranks

(c) general (d) general,early ranks

Fig. 2. Evaluation based on normalised eXtended Cumulated Gain (nxCG)

7 Conclusion

INEX defines a relevant element to be at the right level of granularity if it is exhaustive to the user request – i.e., it discusses fully the topic requested in the user's query – *and* it is specific to that user request – i.e., it does not discuss other topics. The exhaustivity and specificity dimensions are both expressed in terms of the "quantity" of topics discussed within each element. We therefore use the number of topic shifts in an XML element, to express the "quantity" of topics discussed in an element. We experimented with a number of element-specific smoothing methods within the language modelling framework. These methods enable us to adjust the amount of smoothing required for each XML element with respect to the specificity and exhaustivity dimensions of relevance. Using the number of topic shifts combined with element length improved retrieval effectiveness, thus indicating that the number of topic shifts is a useful evidence in focussed XML retrieval. Our other finding was that the smoothing process should be treated differently if we are concerned with the specificity dimension.

For our future work, our first aim is to go beyond the element level for smoothing and provide a term-specific smoothing based on the number of topic shifts and the distribution of the terms inside XML elements. Secondly, we will

investigate the effects of applying the proposed smoothing approach on the Wikipedia XML collection, which is the collection used in INEX 2006[6]. On another direction, we will investigate whether other segmentation algorithms are better suited for XML documents, and whether we can eventually obtain other, more effective means to calculate the number of topic shifts of XML elements.

Acknowledgments

This work was carried in the context of INEX, an activity of the DELOS Network of Excellence.

References

1. E. Ashoori, M. Lalmas and Theodora. Tsikrika. Examining Topic Shifts in Content-Oriented XML Retrieval, submitted, 2006.
2. C. Caracciolo and M. de Rijke. Generating and retrieving text segments for focused access to scientific documents. In *Proceedings ECIR 2006*, pages 350–361, 2006.
3. W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, 2003.
4. B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.
5. N. Fuhr and M. Lalmas. Report on the INEX 2003 Workshop, Schloss Dagstuhl, 15-17 December 2003. *SIGIR Forum*, 38(1):42–47, June 2004.
6. N. Fuhr, M. Lalmas, S. Malik, and G. Kazai, editors. *Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005)*, 2006.
7. M. A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Association for Computational Linguistics*, pages 9–16, 1994.
8. D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proceedings of the 25th ACM SIGIR Conference*, pages 35–41, 2002.
9. J. Kamps, M. de Rijke, and B. Sigurbjörnsson. The importance of length normalization for XML retrieval. *Information Retrieval*, 8(4):631–654, 2005.
10. G. Kazai and M. Lalmas. INEX 2005 Evaluation Metrics. In Fuhr et al. [6].
11. V. Mihajlovic, G. Ramírez, T. Westerveld, D. Hiemstra, H. E. Blok, and A. P. de Vries. TIJAH Scratches INEX 2005: Vague Element Selection, Image Search, Overlap, and Relevance Feedback. In Fuhr et al. [6].
12. P. Ogilvie and J. Callan. Hierarchical language models for XML component retrieval. In *Proceedings of the INEX 2004 Workshop*, pages 224–237, 2005.
13. G. Ramirez, T. Westerveld, and A. P. de Vries. Using structural relationships for focused XML retrieval. In *Proceedings FQAS 2006*, 2006.
14. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM SIGIR conference*, pages 334–342, 2001.

[6] http://inex.is.informatik.uni-duisburg.de/2006/index.html