Length Normalization in XML Retrieval

Jaap Kamps* Maarten de Rijke Börkur Sigurbjörnsson
kamps@science.uva.nl mdr@science.uva.nl borkur@science.uva.nl

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098SJ Amsterdam, The Netherlands

ABSTRACT

XML retrieval is a departure from standard document retrieval in which each individual XML element, ranging from italicized words or phrases to full blown articles, is a potentially retrievable unit. The distribution of XML element lengths is unlike what we usually observe in standard document collections, prompting us to revisit the issue of document length normalization. We perform a comparative analysis of arbitrary elements versus relevant elements, and show the importance of length as a parameter for XML retrieval. Within the language modeling framework, we investigate a range of techniques that deal with length either directly or indirectly. We observe a length bias introduced by the amount of smoothing, and show the importance of extreme length priors for XML retrieval. We also show that simply removing shorter elements from the index (by introducing a cut-off value) does not create an appropriate document length normalization. Even after increasing the minimal size of XML elements occurring in the index, the importance of an extreme length bias remains.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Experimentation

Keywords

XML retrieval, language models, length normalization, smoothing

1. INTRODUCTION

The importance of document length normalization is a recurring theme in information retrieval (IR). In the early days of IR, test collections were based on abstracts, resulting in short documents about a single topic. Here, taking into account parts of the document not about the topic at hand (negative information) was as important as accounting for positive information [29]. This motivated techniques like the standard SMART method of document length normalization using a cosine function. The advent of TREC in 1992 introduced large-scale test collections with full-text documents. Documents in these collections were much longer, and had more length variety than the collections based on abstracts. Full-text documents usually have multiple subtopics, frustrating the use of negative information [4]. As a result, full-text retrieval necessitated a revision of document length normalization [32]. The introduction of XML retrieval marks a similar revolution in IR. Although a text collection of XML documents may have a similar number of articles as standard TREC-sized collections, the number of XML elements in the collection takes us to quite a different scale. There are millions of XML elements that can potentially be retrieved as an answer to a query, having a great variety in length (ranging from single words or phrases put in italics or in titles, to full-blown articles). XML retrieval prompts us to revisit the issue of length normalization.

The task on which we focus in this paper is XML *element* retrieval. Here, each of the text elements into which XML documents are divided, is an object that can in principle be returned in response to a query. Thus, XML element retrieval is one of several recent retrieval tasks aimed at pinpointing highly relevant information; other examples include question answering [34] and the novelty track [33]. The INitiative for the Evaluation of XML retrieval (INEX) was launched in 2002 to assess the effectiveness of retrieval methods for XML document and element retrieval [16]. We focus on so-called *content-only* (CO) topics, which are traditional IR topics written in natural language. Length-wise there are several noteworthy aspects of the INEX test collection. First, the collection has over 12,000 articles, but nearly 7,000,000 XML elements. Second, the XML element length distribution is much more skewed than normal document length distributions. Third, in XML element retrieval the assessors have a strong bias toward retrieval of long elements [17]. By accounting for these length aspects of XML elements during retrieval, systems can improve performance.

Although we could have applied the methodology of Singhal et al. [32] directly to the problem of XML retrieval, we

*Currently at Archives and Information Studies, Faculty of Humanities, University of Amsterdam.

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

follow a somewhat different approach. The reasons for this are twofold. First, we do not want to rely on a particular retrieval system for our analysis, since there is no consensus yet on what the default settings for XML retrieval are. In fact, systems using standard settings from ad hoc retrieval do not perform impressively. Second, we want to address the problem within the language modeling framework, focusing on techniques that address length either directly or indirectly. No matter which retrieval model one uses, main components that affect the importance of a term in a text are the term frequency, the inverse document frequency, and document length. In the generative language modeling approach that we adopt in this paper, these three aspects are captured by the model(s), smoothing procedures, and priors [26]. Our overall motivation is to identify effective XML retrieval methods that are highly portable across XML collections in the sense that they only exploit statistical aspects (both content and non-content) of XML documents, and do not depend on specific schemas or tag sets. Specifically, in this paper we aim to understand how *priors* and *smoothing* affect XML element retrieval performance. We address these two issues in the following manner.

For the *priors* aspect, we need to bridge the gap between the average element length and average *relevant* element length. Since we want to balance the "pinpointing" nature of the XML element retrieval task with the (apparent) importance of long elements, we want to do something more intelligent than only returning the longest possible elements (i.e., articles) in the collection. One of the important contributions of language modeling in IR is the recognition of parameter estimation as a fundamental issue in IR [12]. An unbiased estimator need not be the best estimator; for a number of applications it can be highly advantageous to accept a certain degree of estimation bias if in return there is a reduction in estimation variance. Document priors and smoothing provide a convenient way of biasing estimates [3, 26]. Priors allow one to import "non-content" features of documents (or elements) into the scoring mechanism. Document length is a good example of information about a document that is not directly related to its contents, but might still be related to the possible relevance of the document. Singhal et al. [32] showed that for ad hoc document retrieval, there is a correlation between document length and a priori probability of relevance.

Our other main issue in this paper is *smoothing* for XML element retrieval. Since document (and element) language models may suffer from inaccuracy due to data sparseness, a core issue in language modeling is *smoothing*, which refers to adjusting the maximum likelihood estimator for the document (or element) language model by combining it with a background language model. Two things are at stake: first, since element scores are constructed from very short amounts of text, improving the probability estimates is very important. Second, smoothing facilitates the generation of common terms (a $tf \cdot idf$ like function). Smoothing is known to be task dependent. Language models for ad hoc retrieval, and other tasks assessed in terms of mean average precision scores, tend to perform better if much smoothing is done [18, 13]. In contrast, language models for high precision tasks such as web retrieval tasks seem to perform better if very little smoothing is applied [19]. Smoothing plays a special role in XML retrieval: With smoothing, short elements containing only one or a few of the query terms will receive a high relevance score. Without smoothing, only elements containing all query terms will be returned. So within the language modeling framework, the amount of smoothing is a factor that may affect the length of retrieved elements.

The rest of this paper is organized as follows. We discuss related work in Section 2. In Section 3 we take a closer look at length features of the INEX 2002 and 2003 test suites. Section 4 details our retrieval model, and describes our experiments with the effect of length priors and smoothing on XML element retrieval performance, and in Section 5 we discuss the results of our experiments. Section 6 concludes the paper.

2. RELATED WORK

Related work comes in several kinds; here we discuss language modeling and XML retrieval.

2.1 Language Modeling

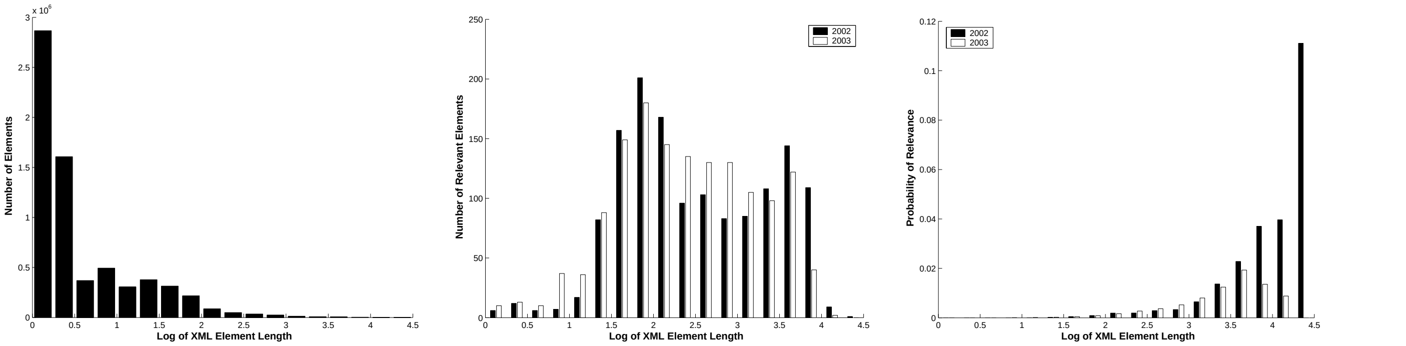
Language modeling approaches to IR provide a promising formal framework for describing a range of retrieval processes, such as web retrieval [19] and cross-lingual retrieval [13]. Language models provide a natural setting for modeling structured documents. The basic idea is to estimate a language model for each document, and then rank documents by the likelihood of the query according to the estimated model. Since the document (or element) score is generally a sum of logarithms of the probability of a word given a document (or element) model, the retrieval performance is generally sensitive to the smoothing parameters [37]. A simple, yet effective smoothing procedure, which has been successfully used for ad hoc and other retrieval tasks alike (and which we also use in this paper) is linear interpolation [26, 37].

Working in the language modeling setting, Hiemstra and Kraaij [15] show that document length serves as a helpful prior for the ad hoc task at TREC, but others have not found document priors to make a significant difference for ad hoc tasks [21]. Miller et al. [26] combined information in their document priors, including document length. In the setting of web retrieval, Kraaij et al. [20] used priors based on the depth of the URL.

2.2 XML Retrieval

Early work by Wilkinson on structured documents showed that extracting XML elements from a ranked list of documents is a poor strategy [36]; one of the positive outcomes, however, was that exploiting the structure of documents can lead to improved *document* retrieval performance. At INEX 2002 [8] and 2003 [9], a broad spectrum of techniques was used to exploit non-content aspects of XML documents in addressing the XML element retrieval task. For instance, the JuruXML system by Mass et al. [25, 5] extends the traditional vector space model by allowing XML collections to be searched through so-called "XML fragments" which combine content and structure features. Similarly, Gövert et al. [10] exploit content and structure features to identify relevant elements and to redistribute relevancy from elements to their enclosing elements.

Several teams have used a language modeling approach to XML element retrieval. E.g., Ogilvie and Callan [27, 28] use a tree-based generative language model for ranking documents and components. Nodes in the tree correspond to document components, and at each node in the document



(a) XML element lengths (b) Relevant element lengths (c) Probability of relevance

Figure 1: Length distribution of XML elements.

tree, there is a language model. The language model for a leaf node is estimated from the component associated with the node; inner nodes are estimated using a linear interpolation among the children nodes. List and De Vries [22] use a language modeling approach where structural properties of documents are mapped to dimensions of relevance and these dimensions are used for retrieval purposes. Hiemstra [14] presents a complex architecture, catering for XPath queries and traditional IR-style statements of an information need, based on a language modeling component for the IR part; one of his findings is that "it is beneficial to assign a higher prior probability of relevance to bigger fragments of XML data than to smaller XML fragments." At INEX 2003, Abolhassani et al. [1] experimented with adaptations of a language model based on Amati's divergence from randomness [2] to XML element retrieval. Finally, the TIJAH XML-IR system by List et al. [23] follows a 'standard' layered database architecture in which the conceptual level is built around a language modeling approach to information retrieval.

3. XML ELEMENT LENGTH

To better understand the importance of element length for XML retrieval, we analyze the length of XML elements versus the length of relevant XML elements. We do this by ordering the elements in the INEX collection by length, and dividing them into several "bins" [32]. We calculate length by the number of term occurrences in an element. Following [20], we use exponential-sized bins. Specifically, we use 20 bins on an exponential scale ranging from 10^0 (=1) to $10^{4.5}$ (=31,623). We only look at text-carrying elements as they occur in our index, which are shorter due to stopword removal.

Figure 1(a) shows the number of XML elements for each of the bins. The distribution of elements is heavily skewed toward short elements, such as italics. The average XML element is short, with a length of 29, while the median length is only 2. We also investigate the length of relevant XML elements, by using the strict assessments of INEX 2002 and 2003 CO topics. Figure 1(b) shows the number of relevant XML elements over all INEX 2002 and 2003 CO topics. Apart from the shortest elements, say shorter than 10, the distribution of elements is fairly even over the bins. The two topic sets are, qualitatively, quite similar, although relatively fewer of the longest elements are relevant for the 2003 topic set. There is a radical difference between the length distributions of relevant XML elements and of all XML elements in the collection. The average relevant element is substantially longer than the average element in the collection. For the 2002 topic set, the mean length is 1,484 and the median length 221. For the 2003 topic sets, the mean length is 1,010 with a median of 226.

We can further investigate the observed difference by estimating the prior probability of relevance of XML elements in each of the bins. Figure 1(c) shows the probability that an XML element in each of the bins is relevant for any of the INEX CO topics. The distribution is heavily skewed toward long elements, such as full articles. The main difference between the 2002 and 2003 topic sets is in the longest elements having thousands of terms; considering that the average article has length 3,234 (with a median of 2,737) these are mostly long articles or their bodies. The difference between the probability of relevance curves in Figure 1(c) and the XML element length curve in Figure 1(a) could hardly be more striking. If we do XML retrieval that is unbiased with respect to length, our retrieved elements will be distributed like in Figure 1(a). Given the prior probability of relevance, this is far from optimal. This clearly shows that XML element length is a crucial parameter for XML retrieval.

4. EXPERIMENTS

4.1 Collection and Metrics

We evaluate our methods against the INEX 2002 and 2003 XML information retrieval test-suites [11]. The INEX document collection contains over 12,000 articles (consisting of nearly 7,000,000 elements) from 21 IEEE Computer Society journals, with layout marked up with XML tags. The collection contains around 170 different tag-names, representing units as diverse as complete articles ⟨article⟩, sections ⟨sec⟩, paragraphs ⟨p⟩ and italics font ⟨it⟩.

In the INEX initiative, relevance is assessed at the element level. Elements are assessed on a two dimensional graded relevance scale, with one dimension for topic relevance (or exhaustiveness) and another for element coverage (or specificity). Relevance or exhaustiveness measures the extent to which an element covers or discusses the topic at hand, while coverage or specificity measures the extent to which the element focuses on the topic at hand (see [8, p. 184] or [9, p. 204] for details). We evaluate our method on a strict scale, considering an element relevant if, and only if, it is judged highly relevant (exhaustive) with exact cover-

	2002 topic set					2003 topic set				
	λ	β	N	MAP	Change	λ	β	N	MAP	Change
Normal ad hoc settings	0.2	1.0	0	0.0409	(baseline)	0.2	1.0	0	0.0832	(baseline)
Optimal smoothing	0.9	1.0	0	0.0598	+46%***	0.6	1.0	0	0.0916	+10%
Extreme length prior	0.2	2.0	0	0.0682	+67%***	0.2	2.0	0	0.1457	+75%***
	0.2	3.0	0	0.0839	+105%***	0.2	3.0	0	0.1329	+60%*
Cut-off	0.9	0.0	40	0.0551	+35%**	0.6	0.0	20	0.0915	+10%
Cut-off + length prior	0.2	2.0	40	0.0781	+91%***	0.2	2.0	40	0.1530	+84%**
	0.2	3.0	40	0.0883	+115%***	0.2	3.0	40	0.1364	+64%*

Table 1: Comparison between different retrieval methods and topic sets.

age (highly specific). We use version 1.8 of the INEX 2002 relevance assessments and version 2.4 of the INEX 2003 assessments. There are assessments for 23 topics (2002) and 26 topics (2003), the average number of relevant elements per topic is 61 (2002) and 56 (2003).

Evaluation is done using the `trec_eval` program. For the task we are evaluating the `trec_eval` program gives the same results as the `inex_eval` program provided by the INEX initiative [8, 9]. We choose to use `trec_eval` since it has been thoroughly tested and since it allows us to use our previously developed tools for result analysis and significance testing.

4.2 Retrieval Framework

Since individual XML elements are the unit of retrieval, we treat each element as a separate indexing unit. For each element we index all the text that is contained within it, including the text within its descendants. Hence we create an overlapping index, since the text nested at depth n in the XML tree is indexed as part of n different indexing units. We do not apply a stemming algorithm, but lowercase all text and remove stop-words. Our index contains 6,779,686 text-carrying elements, 12,107 of which are the articles. This index is highly redundant with 196,960,033 terms, while the non-overlapping article elements contain only 39,155,803 terms (a factor of 5 smaller).

All our retrieval runs are based on a multinomial language model, with tunable length prior and Jelinek-Mercer smoothing [13]. We estimate a language model for each XML element in the collection. For a given query we rank the elements with respect to the likelihood that the element language models generate the query. This can be viewed as estimating the probability $P(e, q)$,

$$P(e, q) = P(e) \cdot P(q|e) \tag{1}$$

where e is an element and q is the query. Thus we need to estimate two probabilities: the prior probability of the element, $P(e)$; and the probability of generating the query, $P(q|e)$. We will explore several ways estimating the prior probability of an element as a function of its length. For the probability of the query we assume the terms to be independent, and we use a linear interpolation of an element model and a collection model to estimate the probability of a query term. The probability of a query $t_1, \ldots, t_n$ is estimated as,

$$P(t_1, \ldots, t_n|e) = \prod_{i=1}^{n} (\lambda \cdot P(t_i|e) + (1 - \lambda) \cdot P(t_i)), \tag{2}$$

where $P(t_i|e)$ is the probability of observing a term in an element, and $P(t_i)$ is the probability of observing the term in the collection. Both probabilities are estimated using maximum likelihood estimation. The parameter λ is the so-called smoothing parameter. The calculation of probabilities can be reduced to the scoring formula for an indexing unit e and query $t_1, \ldots, t_n$,

$$\begin{aligned} s(e, t_1, \ldots, t_n) &= \beta \cdot \log \left(\sum_t tf(t, e) \right) \\ &+ \sum_{i=1}^{n} \log \left(1 + \frac{\lambda \cdot tf(t_i, e) \cdot \left(\sum_t df(t) \right)}{(1 - \lambda) \cdot df(t_i) \cdot \left(\sum_t tf(t, e) \right)} \right), \end{aligned} \tag{3}$$

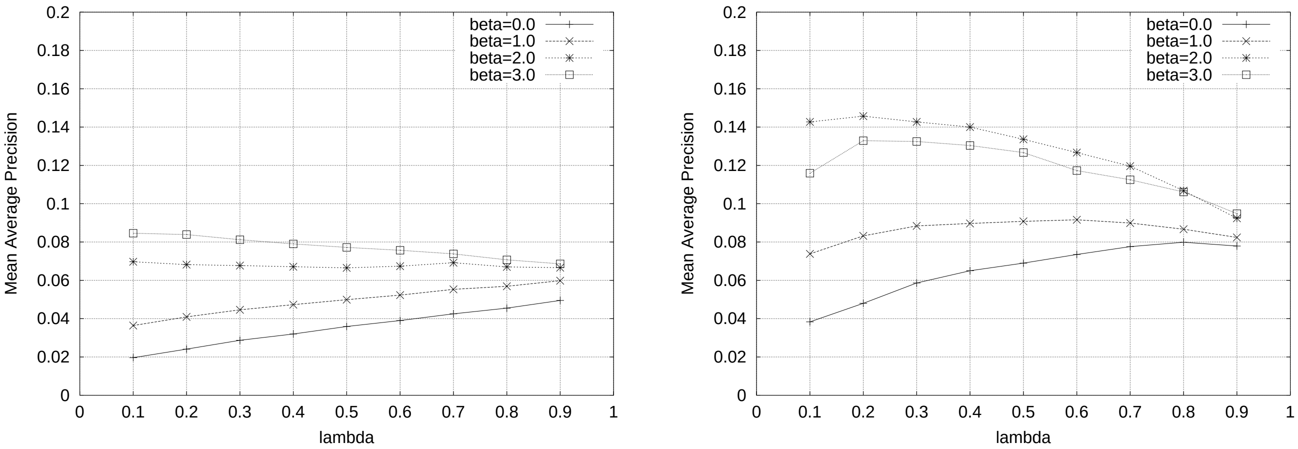
where $tf(t, e)$ is the frequency of term t in unit e; $df(t)$ is the count of units in which term t occurs; and λ is the weight given to the element language model when smoothing with the collection model. Our introduction of the parameter β serves as a handy knob to turn when trying to bridge the length gap between an average element and an average relevant element. For $\beta = 0$ this results in a uniform distribution over length or using no length prior, for $\beta = 1$ this results in a normal length prior, for $\beta = 2$ a squared length prior (the prior probability of an element is proportional to the square of its length), etcetera.

Another way to try to bridge the length gap between average elements and average relevant elements is to restrict the view of the index to the elements that are the most likely to be relevant to a query. Two approaches to this aim have been proposed. One is to index only elements whose tag names are from a predefined list of tag names. The list can be compiled after careful analysis of tag name semantics [10] or by using existing relevance assessments [24]. The other way is to restrict the view to elements that pass a certain length threshold, or a cut-off value N. We will explore the latter option since we want to explore retrieval methods that do not rely on specific schemas or tag sets. We apply the index cut-off value N after building the index, but also prune the statistics accordingly, making it equivalent to having only indexed elements of size at least N.

4.3 Runs

Our runs are made using only the *title* and *description* fields of the topics. The topics are processed as the collection; the text is lower-cased and stop-words are removed.

To determine the effect of smoothing we experiment with a range of values for the smoothing parameter λ from the interval $[0.1, 0.9]$. To determine the importance of the length prior, β, we experiment with values ranging from 0.0 to 5.0. In order to examine the interplay between the smoothing parameter and the length prior we run experiments on the two-dimensional search space determined by λ and β.



(a) INEX 2002 topics

(b) INEX 2003 topics

Figure 2: Mean average precision for different length priors β, plotted against the smoothing parameter λ.

As we want to compare the effect of the length prior and the effect of index cut-off, we perform similar experiments for different index cut-off values as we do for different length prior values. Hence, we explore the two dimensional search space determined by λ and N, where the index cut-off values N range from 0 to 60.

The length prior and the index cut-off value have the same aim; to draw the attention from the very many, very short elements in the collection, to the fewer, longer elements appreciated by assessors. To find out whether these two methods complement each other we explore the search space determined by β and N. Here, we will limit the search space to key values of the length prior and cut-off parameters.

5. RESULTS AND DISCUSSION

Table 1 shows a summary of the results we will discuss in this section. As our baseline we choose a retrieval run with parameter settings that are considered traditional for ad hoc retrieval. That is, we use a low value for the smoothing parameter ($\lambda = 0.2$) and we use a normal length prior ($\beta = 1.0$). To determine statistical significance we use the bootstrapping method, a non-parametric inference test [6, 7]. The method has previously been applied to retrieval evaluation by, e.g., [35] and [30]. We take 100,000 re-samples and look for improvements (one-tailed) at significance levels 0.95 (*), 0.99 (**), and 0.999 (***). Because of the bewildering size, the parameter space has not been fully explored to find the optimal parameter for each method.

5.1 Smoothing for Length Bias

Figure 2 shows the mean average precision for different values of the length prior and different values of the smoothing parameter. To determine the effect of smoothing we look at the curves where there is no length prior ($\beta = 0.0$) or where the normal length prior is used ($\beta = 1.0$). We see that, for both topic sets, the optimal value of the smoothing parameter λ is in the higher end. This is surprising since these are settings normally applied for high-precision retrieval tasks. The XML retrieval task, in contrast, is an ad hoc retrieval task evaluated with mean average precision. A closer look at the retrieved elements shows that, on average, longer elements are returned when a higher value is given to the smoothing parameter λ. The average length of retrieved elements can be seen in Figure 3. The figure shows the runs for the 2002 topic set, the plot for the 2003 topic set is nearly identical. A high value of λ means that the presence of a query term in an element is rewarded (we are approaching coordination level matching). Since long elements are more likely to contain many of the query terms, high values of λ have the required length bias effect.

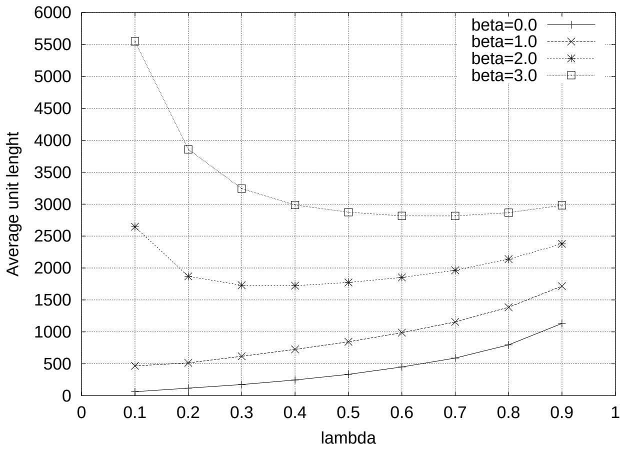
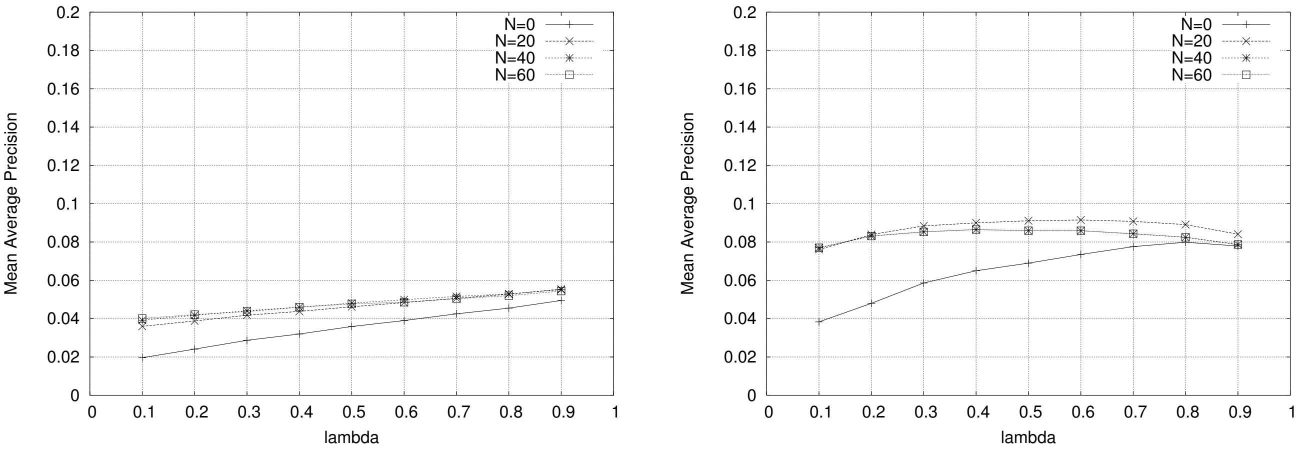


Figure 3: Average length of retrieved elements, for different length priors β, plotted against the smoothing parameter λ. (Using the 2002 topics.)

Comparing the two topic sets we see that the optimal value for the smoothing parameter λ is slightly different. The optimal value is 0.9 for the 2002 collection but 0.6 for the 2003 collection. This can be explained by the fact that the 2003 assessments seem to have slightly less bias toward long elements than the 2002 assessments do (see Figure 1). By increasing the smoothing parameter we get an improvement of +46% (***) and +10% ($^{-}$) over the standard ad hoc settings, respectively for the 2002 and 2003 topic sets.



(a) INEX 2002 topics

(b) INEX 2003 topics

Figure 4: Mean average precision for different cut-off values N, plotted against the smoothing parameter λ.

5.2 Length Priors

To determine the effect of the length prior we look at the curves for $\beta = 2.0$ and $\beta = 3.0$ in Figure 2. In the remainder of this paper, we will refer to either of these values of the length prior as *extreme length priors*. For both topic sets, the score improves over the normal length prior settings. Also, the optimal value for the smoothing parameter λ moves to the lower portion of the search space. The optimal value for the smoothing parameter is now in line with other tasks evaluated using mean average precision. This is because the smoothing parameter no longer works as a length bias. That function is taken care of by the extreme length priors. See Figure 3 for the average length of retrieved elements. Even higher values of the smoothing parameter did not result in improved scoring.

Comparing the two topic sets we see that the optimal value for the length prior β is different. While the runs using the 2002 topic set peak at $\beta = 3.0$, the runs using the 2003 topic set peak at $\beta = 2.0$. The fact that a less extreme length prior is needed for the 2003 topic set is, again, in line with the observations on length bias for the INEX topics in Section 3. Increasing the length prior β up to 3.0 gives us an improvement of +105% (***) over the baseline for the 2002 topic sets and +60% (*) for the 2003 topic set. Although we are not using the optimal value for the 2003 topic set, we still get a statistically significant result. If we look at the optimal value for the 2003 topic set, $\beta = 2.0$, the improvement is +67% (***) for the 2002 topic set and +75% (***) for the 2003 topic set. The optimal value for the length prior changes between topic sets. However, for both topic sets we get quite remarkable and statistically significant improvements, even when we use the sub-optimal length prior learned from experiments on the other topic set. This tells us that length must be taken seriously when retrieving XML elements, and extreme length priors are of great importance.

Finally, we have seen that the smoothing parameter is dependent on the length prior. In the absence of a length prior, the smoothing parameter introduces a length bias. However, when we do use an extreme length prior for the length bias, the smoothing parameter can go back to what it does best, namely smoothing.

5.3 Element Length Cut-offs

The length priors have a dual effect: on the one hand they make it effectively impossible to retrieve short elements, and on the other hand they influence the relative ranking of longer elements. We investigate now the relative importance of these two effects, restricting the minimal size of XML elements in our index. That is, we explore the effect of different values for the index cut-off, N, using no length prior ($\beta = 0.0$), but different values for the smoothing parameter λ. Remember that using an index cut-off N is equivalent to using an index where we only index elements containing N or more terms.

Figure 4 shows the effect of a few cut-off settings on both the INEX 2002 and INEX 2003 topic sets. Using cut-offs does indeed improve scoring. There is not much difference in performance between different cut-off values in the interval from 20 to 50. As the cut-off value further increases, the performance tapers off and starts to drop, since we are simply leaving out too many relevant elements from our index (see Figure 1). Again we see a slight difference in the optimal value for the two test sets: the 2002 topic set peaks at $N = 40$ and $\lambda = 0.9$ with +35% (**), and the 2003 topic set peaks at $N = 20$ and $\lambda = 0.6$ with +10% ($^{-}$). The performance for the 2002 topics drops only slightly when going from $N = 40$ to $N = 60$. The performance for the 2003 topics drops already when going from $N = 20$ to $N = 40$. This difference can again be explained by the different length bias in the two assessment sets.

For both topic sets, the cut-off improves scoring by far less than the extreme length priors. Cut-off does help to get rid of the very many very short elements, but we still need an explicit length bias to distinguish between the longer and shorter elements remaining in our index. Figure 5 shows the average length of retrieved elements for different cut-off values. It is interesting to note that the curves for the cut-off, with no length prior, are very similar to the curve for the single length prior in Figure 2. It is plausible that the reason why the single length prior does not perform good enough is that its only effect might be to downplay the very many,

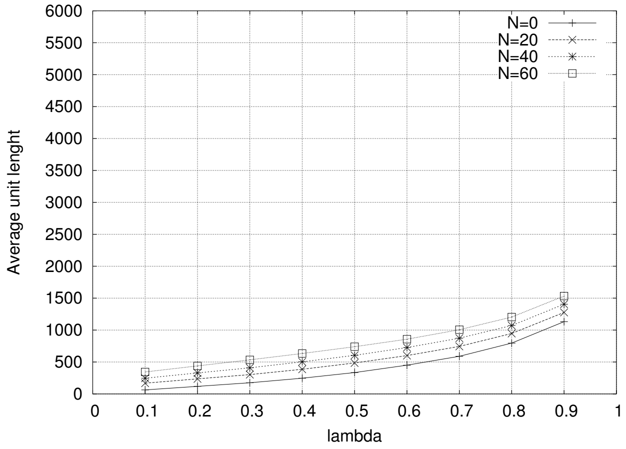


Figure 5: Average length of retrieved elements, for different cut-off values N, plotted against the smoothing parameter λ. (Using the 2002 topics.)

very short elements, but it does not introduce enough bias toward the relatively long elements considered meaningful by assessors.

5.4 Length Prior plus Cut-off Value

We have seen that although extreme length priors and a cut-off value can both lead to improved scoring, the two methods do not behave in the same way. While the extreme length priors introduce a bias toward longer elements within an index, the cut-off merely keeps the very short elements outside of the index. Furthermore, the extreme length prior leads to far greater improvements than the cut-off. Therefore, it is interesting to see whether the extreme length prior runs can improve further when applied on a cut-off index.

To demonstrate this effect we choose to apply the extreme length prior together with an index cut-off $N = 40$. There is, however, hardly any difference in performance between choosing different cut-off values in the interval from 20 to 50. Combining a value of 3.0 for length prior β with a value of 40 for cut-off N does indeed give us further improvements. For the 2002 topic set the improvement is $+115\%$ (***) over the normal baseline and we get $+5.2\%$ (*) improvement over the length prior ($\beta = 3.0$) alone; both improvements are statistically significant. For the 2003 topic set the improvement is $+64\%$ (*) over the normal baseline and $+2.6\%$ ($^{-}$) over the length prior alone; here, only the improvement over the baseline is statistically significant. Using $\beta = 2.0$ for the length prior and $N = 40$ for the cut-off also improves the results. For the 2002 topic set the improvement is $+91\%$ (***) over the normal baseline and and $+15\%$ (**) over the length prior ($\beta = 2.0$) alone. For the 2003 topic set the improvement is $+84\%$ (**) over the normal baseline and $+5\%$ ($^{-}$) over the length prior alone.

The improvement effect of index cut-off is not as clear as the effect of the extreme length prior. Alone, it is by far inferior to the extreme length priors. Combining length prior and cut-off does improve over the use of length prior alone, but the improvement is statistically significant only for one of the two topic sets.

6. CONCLUSION

This paper revisited document length normalization in the context of an XML element retrieval task. We performed a comparative analysis of the length of arbitrary elements versus that of relevant elements, and highlighted the importance of length as a parameter for XML retrieval. Within the language modeling framework, we investigated techniques that deal with length either directly or indirectly: length priors, index cut-off, and the amount of smoothing. We observed a length bias introduced by the amount of smoothing, and showed the importance of extreme length priors for XML retrieval. When used with extreme length priors, the smoothing parameter regains its normal function of controlling term importance. Furthermore, we showed that simply removing shorter elements from the index (by introducing a cut-off value) does not create an appropriate document length normalization. After restricting the minimal size of XML elements occurring in the index, the importance of an extreme length bias remains. The combination of length priors with index cut-off does lead to a slight further improvement.

The importance of extreme length normalization in XML retrieval should not be interpreted as a general claim that long XML elements are inherently more relevant than short elements. Rather, it is a way to counterbalance the heavily skewed length distribution of elements in the collection. Our analysis showed that relevant elements are fairly evenly distributed over length. We need the extreme length normalization in order to retrieve a distribution of elements that is not skewed over length. Based on these observations, we postulated that accounting for these length aspects of XML retrieval leads to improvement of performance, which was confirmed by our experiments.

Although we find convincing evidence for our findings on the INEX collection, and although the value of the approach has been demonstrated by the top ranking results of a system implementing the approach at the INEX 2003 workshop [31], the usual disclaimers apply. As with any experimental result, there is no guarantee that these results will carry over to every other collection. XML collections can have great variety in structure, potentially very different from that in full-text digital libraries like the IEEE Computer Society. Furthermore, the INEX test-suite is based on peer-assessments by one judge per topic (leading to considerable variety in judgments especially between topics) and facilitated by a particular interface (potentially creating some biases, e.g., elements are presented within the context of the full article). By the same token, it is clear that the observed length effects are not unique for XML retrieval. Similar effects may be observed in every collection where the distribution of lengths of documents is very skewed. Arguably, some of these effects, such as the length-bias introduced through smoothing, play a role with every collection, be it to a lesser extent than in the case of XML retrieval.

7. ACKNOWLEDGMENTS

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO) under project number 612.066.302. Maarten de Rijke was supported by grants from NWO, under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.000.207, and 612.-066.302.

8. REFERENCES

[1] M. Abolhassani, N. Fuhr, and S. Malik. HyREX at INEX 2003. In Fuhr et al. [9], pages 27–32.

[2] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20:357–389, 2002.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd ACM SIGIR Conference*, pages 222–229, 1999.

[4] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In *The Fourth Text REtrieval Conference (TREC-4)*, pages 25–48.

[5] D. Carmel, Y. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching XML documents via XML fragments. In *Proceedings of the 26th ACM SIGIR Conference*, pages 151–158, 2003.

[6] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

[7] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, 1993.

[8] N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors. *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2002)*. ERCIM, 2003.

[9] N. Fuhr, M. Lalmas, and S. Malik, editors. *INEX 2003 Workshop Proceedings*, 2004.

[10] N. Gövert, M. Abolhassani, N. Fuhr, and K. Grossjohan. Content-based XML retrieval with HyRex. In Fuhr et al. [8], pages 26–32.

[11] N. Gövert and G. Kazai. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In Fuhr et al. [8], pages 1–17.

[12] W. R. Greiff and W. T. Morgan. Contributions of language modeling to the theory and practice of information retrieval. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 73–93. Kluwer Academic Publishers, 2003.

[13] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[14] D. Hiemstra. A Database Approach to Content-based XML Retrieval. In Fuhr et al. [8], pages 111–118.

[15] D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In *The Seventh Text REtrieval Conference (TREC-7)*, pages 227–238, 1999.

[16] INEX. Initiative for the evaluation of XML retrieval, 2004. `http://www.is.informatik.uni-duisburg.de/projects/inex03/`.

[17] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. XML Retrieval: What to Retrieve? In *Proceedings of the 26th ACM SIGIR Conference*, pages 409–410, 2003.

[18] W. Kraaij, R. Pohlmann, and D. Hiemstra. Twenty-One at TREC-8: using language technology for information retrieval. In *The Eighth Text REtrieval Conference (TREC-8)*, pages 285–300, 2000.

[19] W. Kraaij and T. Westerveld. Twenty-UT at TREC-9: How different are web documents? In *The Ninth Text REtrieval Conference (TREC-9)*, pages 665–672, 2001.

[20] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM SIGIR Conference*, pages 27–34, 2002.

[21] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2003.

[22] J. List and A. P. de Vries. CWI at INEX 2002. In Fuhr et al. [8], pages 133–140.

[23] J. A. List, V. Mihajlovic, A. P. de Vries, G. Ramírez, and D. Hiemstra. The TIJAH XML-IR system at INEX 2003. In Fuhr et al. [9], pages 102–109.

[24] Y. Mass and M. Mandelbrod. Retrieving the most relevant XML components. In Fuhr et al. [9], pages 53–58.

[25] Y. Mass, M. Mandelbrod, E. Amitay, D. Carmel, Y. Maarek, and A. Soffer. JuruXML – an XML retrieval system at INEX'02. In Fuhr et al. [8], pages 73–80.

[26] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM SIGIR Conference*, pages 214–221, 1999.

[27] P. Ogilvie and J. Callan. Language models and structured document retrieval. In Fuhr et al. [8], pages 33–44.

[28] P. Ogilvie and J. Callan. Using language models for flat text queries in XML retrieval. In Fuhr et al. [9], pages 12–18.

[29] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill computer science series. McGraw-Hill, New York, 1983.

[30] J. Savoy. Statistical inference in retrieval effectiveness evaluation. *Information Processing and Management*, 33:495–512, 1997.

[31] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approch to XML Retrieval. In Fuhr et al. [9], pages 19–26.

[32] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing & Management*, 32:619–633, 1996.

[33] I. Soboroff and D. Harman. Overview of the TREC 2003 Novelty Track. In *The Twelfth Text REtrieval Conference (TREC-12)*, 2004.

[34] E. M. Voorhees. Overview of the TREC 2003 Question Answering Track. In *The Twelfth Text REtrieval Conference (TREC-12)*, 2004.

[35] J. Wilbur. Non-parametric significance tests of retrieval performance comparisons. *Journal of Information Science*, 20:270–284, 1994.

[36] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th ACM SIGIR Conference*, pages 311–317, 1994.

[37] 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.