The Importance of Length Normalization for XML Retrieval

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

Informatics Institute, University of Amsterdam

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 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 element 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 bias 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 element length normalization. Even after restricting the minimal size of XML elements occurring in the index, the importance of an extreme explicit length bias remains.

Keywords: XML retrieval, language models, length normalization, smoothing

1. Introduction

The importance of document length normalization is one of the recurring themes 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 (Salton and McGill, 1983). 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 (Buckley et al., 1996). As a

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

result, full-text retrieval necessitated a revision of document length normalization (Singhal et al., 1996). The introduction of XML retrieval marks a similar revolution in IR. Although a collection of XML text documents may contain 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. Even in XML collections that are moderately sized in terms of the number of documents they contain, there may be millions of XML elements that may 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 that are aimed at pinpointing highly relevant information; other examples include question answering (Voorhees, 2003) and the novelty track (Harman, 2003). 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 (INEX, 2004). 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 over 8,000,000 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 (Kamps et al., 2003b). Singhal et al. (1996, p.620) argued that "a system that retrieves documents of a certain length with a probability similar to that of finding a relevant document of that length, will outperform other systems that retrieve documents with very different probabilities from their probability of relevance." We believe that 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., 1996) directly to the problem of XML retrieval, we 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 would be default settings for XML retrieval. 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 those 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 respectively captured by the model(s), smoothing procedures, and priors (Miller et al., 1999). Our overall motivation for our work is to identify effective XML retrieval methods that are highly portable across XML collections in that they only exploit statistical aspects (both content and non-content) of XML documents, and do not depend on specific DTDs or tag sets. Specifically, as we will now explain, we aim to understand how *priors* and *smoothing* affect XML element retrieval performance.

For the *priors* aspect, we need to bridge the gap between 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 (Greiff and Morgan, 2003). An unbiased estimator need not be the best estimator; for a number of applications it can be 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 (Berger and Lafferty, 1999; Miller et al., 1999). 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. (1996) 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 (Kraaij et al., 2000; Hiemstra, 2001). In contrast, language models for high precision

tasks such as web retrieval tasks seem to perform better if very little smoothing is applied (Kraaij and Westerveld, 2001). In XML retrieval, smoothing plays a special role: 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 (Kraaij and Westerveld, 2001) and cross-lingual retrieval (Hiemstra, 2001). They 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 (Zhai and Lafferty, 2001). 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 (Miller et al., 1999; Zhai and Lafferty, 2001).

Working in the language modeling setting, Hiemstra and Kraaij (1999) show that document length serves as a helpful prior for the ad hoc task at TREC, and others have not found document priors to make a significant difference for ad hoc tasks (Lafferty and Zhai, 2003). Miller et al. (1999) combined information in their document priors, including document length. In the setting of web retrieval, Kraaij et al. (2002) 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 (Wilkinson, 1994); one of the positive outcomes, however, was that exploiting the structure of documents can lead to improved *document* retrieval performance. At INEX 2002 (Fuhr et al., 2003) and 2003 (Fuhr et al., 2004), 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. (2003) and Carmel et al. (2003) 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. (2003) 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 (2003; 2004) 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 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 (2003) 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 (2003) 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." For INEX 2003, the University of Amsterdam's team (Kamps et al., 2003b; Kamps et al., 2004) worked in a generative language modeling setting to experiment with length bias. Also at INEX 2003, Abolhassani et al. (2004) experimented with adaptations of a language model based on Amati's divergence from randomness (Amati and Van Rijsbergen, 2002) to XML element retrieval. Finally, the TIJAH XML-IR system by List et al. (2004) follows a 'standard' layered database architecture in which the conceptual level is built around a language modeling approach to information retrieval.

Table I. 20 longest elements and 20 most frequent tags in the INEX collection

Tag	Description	Mean length	Collection frequency	Tag	Description	Collection frequency	Mean length
index	journal index	3237.21	117	it	italicized text	1,148,636	1.73
article	article	3234.15	12,107	p	paragraph	743,683	35.39
bdy	body	2651.13	12,107	ref	reference	391,651	1.33
bm	back matter	594.24	10,058	au	author	317,709	2.47
dialog	dialog	526.44	194	snm	surname	311,621	1.05
sec	section	459.16	69,728	fnm	first-name	297,609	1.35
bib	bibliography	372.59	8,543	sub	subscript	244,717	1.08
bibl	bibliography	372.32	8,551	entry	table entry	243,208	1.91
ss1	(sub)section	252.02	61,454	ip1	paragraph	178,742	34.75
app	appendix	242.18	5,856	obi	other bib info	165,477	4.24
ss3	(sub)section	189.26	127	ti	title	159,574	4.82
ss2	(sub)section	169.73	16,276	pdt	publication date	154,984	1.51
dl	definition list	91.99	353	yr	year	154,948	1.00
fm	front matter	89.42	12,107	b	boldface text	152,241	2.68
lb	list	80.85	54	bb	citation	149,167	21.26
tgroup	table	80.73	5,805	st	section title	138,867	3.66
tbody	table body	78.40	5,800	atl	article title	134,286	6.04
l4	list	75.22	117	scp	small caps	110,018	1.03
edintro	editorial intro	69.89	571	pp	pages (citation)	108,134	3.07
proof	proof	68.33	3,765	li	list item	76,400	16.69

3. XML Element Length

In our experiments we use the INEX 2002 and 2003 XML information retrieval test-suites (Fuhr et al., 2003; Fuhr et al., 2004). The INEX document collection contains over 12,000 articles (consisting of over 8,000,000 elements) from 21 IEEE Computer Society journals, with layout marked up with XML tags. The collection contains 176 different tag-names, representing units as diverse as complete articles ⟨article⟩, sections ⟨sec⟩, paragraphs ⟨p⟩ and italics font ⟨it⟩. We calculate our statistics from the viewpoint of the retrieval system. That is, we use statistics from our index of 6,779,686 text-carrying elements. Due to data cleaning and stopword removal, these elements are shorter than in the original collection. Similarly, empty elements are not indexed and thus do not count in our statistics. We calculate length as the number of term occurrences in the elements.

Table II. The 20 most frequent tag-names in the strict relevance assessments

2002 assessments			2003 assessments		
Tag-name	Frequency	%	Tag-name	Frequency	%
p	383	27.47%	sec	303	20.89%
article	309	22.16%	p	303	20.89%
sec	291	20.87%	article	172	11.86%
ss1	115	8.24%	bdy	167	11.51%
bdy	90	6.45%	ss1	146	10.06%
ip1	61	4.37%	ip1	69	4.75%
ss2	25	1.79%	ss2	36	2.48%
abs	22	1.57%	fig	32	2.20%
fm	13	0.93%	app	20	1.37%
st	11	0.78%	bb	19	1.31%
item	8	0.57%	art	18	1.24%
app	7	0.50%	bm	17	1.17%
li	5	0.35%	atl	15	1.03%
it	5	0.35%	fm	14	0.96%
kwd	5	0.35%	li	14	0.96%
b	5	0.35%	abs	12	0.82%
atl	4	0.28%	fgc	11	0.75%
bb	4	0.28%	st	10	0.68%
tbl	3	0.21%	tig	9	0.62%
fig	3	0.21%	bib	8	0.55%

3.1. Analysis of XML Element Tags

To get some idea about the kind of elements we are dealing with, it is useful to look at some of the tag-names in the collection. Table I shows the 20 longest elements and the 20 most frequently occurring elements in the collection. The table also gives a description of the tag-name, the frequency of the tag-name in the collection, and the average length of elements bearing the tag-name. The total number of different tag-names in the collection is 176; some more tag names occur in the DTD. From the average lengths listed in Table I it is clear that the most common element types contain very little text. One may argue, therefore, that it is unlikely that they can satisfy the information need of a topic. Indeed, most of the 20 most frequent tag-names occur rarely, if at all, in the assessments.

So what kind of elements were judged relevant for the INEX CO topics in 2002 and in 2003? Table II shows the 20 most frequent tag-

Table III. Prior probability of relevance

2002 assessments			2003 assessments		
Tag-name	Frequency	Prob.Rel	Tag-name	Frequency	Prob.Rel
article	309	0.02552242	article	172	0.01420665
bdy	90	0.00743371	bdy	167	0.01379367
fn	1	0.00613496	index	1	0.00854700
sec	291	0.00417294	sec	303	0.00434502
abs	22	0.00298872	app	20	0.00341122
ss1	115	0.00187016	ss1	146	0.00237429
ss2	25	0.00153487	ss2	36	0.00221021
kwd	5	0.00132625	bm	17	0.00168902
brief	1	0.00123304	abs	12	0.00163021
app	7	0.00119392	fm	14	0.00115635
fm	13	0.00107375	bib	8	0.00093643
bq	2	0.00099850	bibl	8	0.00093556
p	383	0.00050247	kwd	3	0.00079575
ip1	61	0.00033216	tig	9	0.00074331
tbl	3	0.00023547	lc	5	0.00045150
bibl	2	0.00023389	fig	32	0.00041599
bm	2	0.00019870	p	303	0.00039752
lc	2	0.00018060	ip1	69	0.00037572
tgroup	1	0.00017176	hdr	4	0.00033038
tbody	1	0.00017176	hdr1	4	0.00033038

names of elements assessed relevant, both in absolute number over all topics, and as a percentage of all strict assessments. We see that the assessors seem to prefer the longer tag-types such as articles (⟨article⟩); bodies (⟨bdy⟩); sections (⟨sec⟩, ⟨ss1⟩, ⟨ss2⟩); and paragraphs (⟨p⟩, ⟨ip1⟩). Together, those tag-names covered 91% of the INEX 2002 assessments and 86% of the INEX 2003 assessments. Some of the short elements were also judged relevant, but less frequently than the longer elements. These short elements include section titles (⟨st⟩); article titles (⟨atl⟩, ⟨tig⟩); italicized words (⟨it⟩); and boldface words (⟨b⟩). If we compare the assessments for the two years, we see that the weight of articles in the assessments set has decreased somewhat between 2002 and 2003. The weight of the other longer elements (bodies, sections, and paragraphs) has increased.

When we combine the collection frequencies in Table I with the assessments frequency in Table II, we can estimate the prior probability of relevance of each of the tags. We do this by simply dividing the

Table IV. Exponential-sized bins.

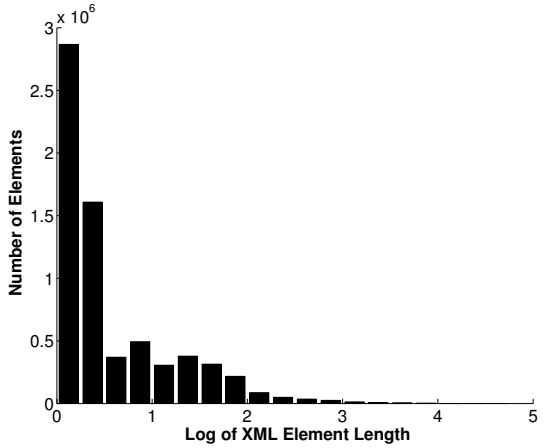
Bin	1	2	3	4	5	6	7	8	9	10
Log max length	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
Max length	1	3	6	10	18	32	56	100	178	316
Bin	11	12	13	14	15	16	17	18	19	20
Log max length	2.75	3.00	3.25	3.50	3.75	4.00	4.25	4.50	4.75	5.00
Max length	562	1000	1778	3162	5623	10000	17783	31623	56234	100000

assessments' frequency by the collection frequency for each of the tag names. Table III shows the 20 tag-names with the highest probability of relevance (for any of the INEX CO topics). The longer XML elements, such as articles, bodies, and sections, do not occur frequently in the collection, but have the highest frequencies in the assessments. As a result, the prior probability of relevance of the longer elements is much higher than that of the frequently occurring shorter elements. Even within the long elements, the probability of relevance reflects their lengths. The longest elements, ⟨article⟩, have the highest prior, followed by the second longest elements, ⟨bdy⟩, then followed by ⟨sec⟩.

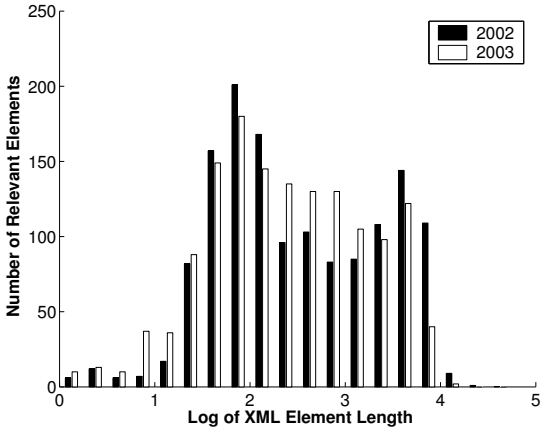
3.2. Analysis of XML Element Length

In our analysis above, we looked at the distribution of XML tags, that is, the distribution of names of the XML elements in the collection. We now analyze the distribution of XML elements over length regardless of their tag-name, and compare the length of arbitrary XML elements versus the length of relevant XML elements. We do this by ordering the elements in the INEX collection by length, and grouping them into several "bins" (Singhal et al., 1996). As before, we calculate length as the number of term occurrences in an element. Following (Kraaij et al., 2002), we use exponential-sized bins. Specifically, we use 20 bins on an exponential scale ranging from 10^0 (=1) to 10^5 (=100,000). Table IV gives the length of the longest element for each of the bins. Note that these bins do not depend on the collection at hand, allowing us to investigate the distribution of elements over length, and make comparisons over different collections.

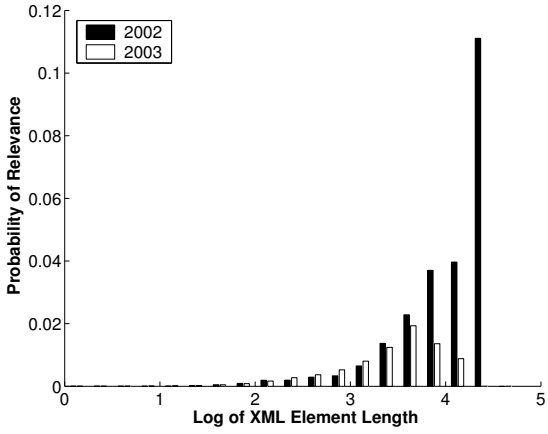
Figure 1(a) shows the number of XML elements for each of the bins. The distribution of elements is heavily skewed toward short elements, such as italics. The average XML element is short, with a length of 29, while the median length is only 2. We also investigate



(a) XML element lengths



(b) Relevant element lengths



(c) Probability of relevance

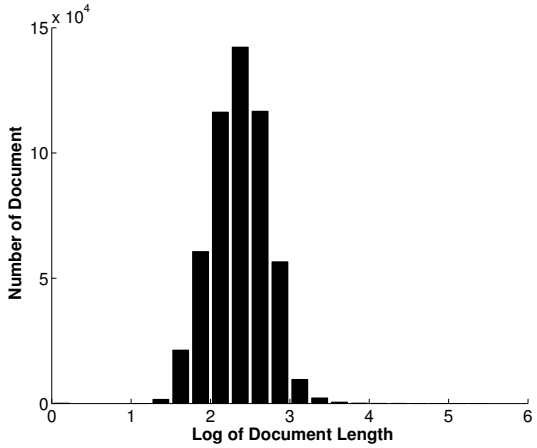
Figure 1. Length distribution of XML elements (INEX IEEE Computer Society Collection).

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 containing fewer than 10 terms, the distribution of elements is fairly even over the bins. There is a radical difference between the length distributions of relevant XML elements and of all XML elements in the collection. The average length of a relevant element is 1,469 (1,100) and the median length is 220 (226) for the 2002 (2003) topics.

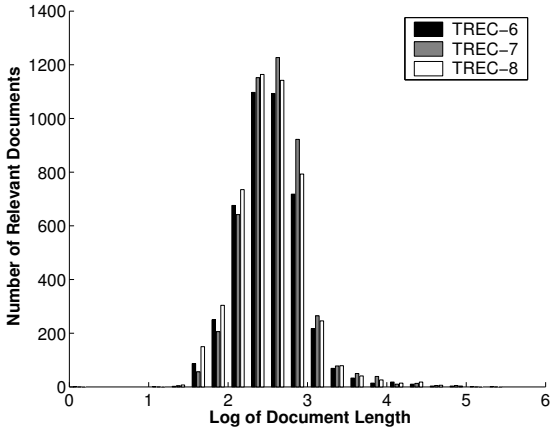
We can further investigate the observed difference by estimating the prior probability of relevance of XML elements in each of the bins. For each bin, Figure 1(c) shows the probability that an XML element in that bin is relevant for any of the INEX CO topics. The distribution is heavily skewed toward long elements, such as full articles. The difference between the probability of relevance curve 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 the collection 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.

For comparison, we conduct a similar analysis for ad hoc document retrieval using TREC-style collections. We use the combined collection consisting of the documents from the Financial Times, the Federal Register 1994, the LA Times, and the FBIS (i.e., TREC disks 4 and 5 without the Congressional Record). There are 528,155 documents in total, with an average length of 319 terms, and a median length of 231. We use the assessments of TREC 6 ad hoc task (topics 301–350), the TREC 7 ad hoc task (topics 351–400), and the TREC 8 ad hoc task (topics 401–450). For the TREC 8 ad hoc topics, there are 4,728 relevant documents, with a mean length of 730 and a median length of 318 terms.

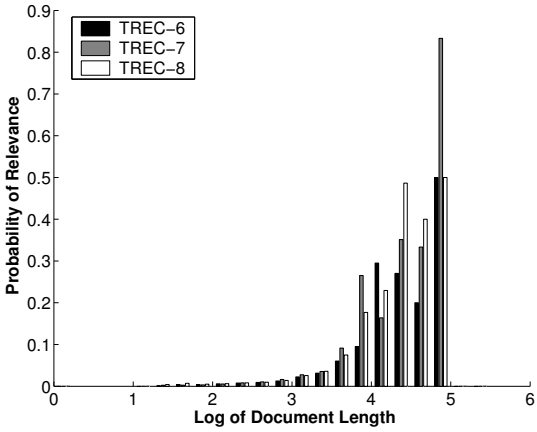
Figure 2(a) shows the number of documents for each of the bins. The documents have a normal distribution around the mean document length. The number of relevant documents per bin, shown in Figure 2(b), exhibits a very similar distribution as the total number of documents. Figure 2(c) shows, again, the prior probability of relevance per bin. The prior probability of relevance per bin is heavily skewed toward long documents. The probability of relevance of Figure 1(c) and that of Figure 2(c) show a similar distribution. This suggests that, despite the different retrieval task, assessors in INEX and TREC make comparable relevance judgments relative to the length of the retrieval unit. The main difference between XML retrieval and TREC-



(a) Document lengths



(b) Relevant document lengths



(c) Probability of relevance

Figure 2. Length distribution of documents (TREC Disks 4 and 5 minus the Congressional Record).

style document retrieval, then, is the heavily skewed distribution of XML elements.

4. Experiments

We explore the importance of length normalization for XML retrieval. In Section 3 we have seen that the INEX collection of XML elements is quite different from the TREC collections, with respect to the length of retrievable units. An analysis of the INEX assessments shows that a bias is needed toward retrieving relatively long elements. We focus on three aspects of our retrieval model which affect this bias:

1. smoothing of language models,
2. priors based on element length, and
3. by introducing a minimal length for elements being indexed (an index *cut-off* value).

Retrieval effectiveness depends on the settings of smoothing parameters (Zhai and Lafferty, 2001), and for XML retrieval the smoothing parameter has proved to indirectly introduce a length bias by decreasing the importance of the presence of query terms (Kamps et al., 2003a). The length prior directly introduces a preference for longer elements, which has proved important for XML retrieval (Kamps et al., 2003b). The index cut-off introduces a bias toward retrieval of long elements by leaving the many very short elements out of the index (Sigurbjörnsson et al., 2004). In our experiments we aim to investigate these aspects in more detail, and explore how they affect and complement each other.

4.1. Assessments and Evaluation Metrics

In the INEX initiative, relevance is assessed at the element level. Elements are assessed on a two dimensional graded relevance scale, one for topic relevance (or exhaustiveness) and another for element coverage (or specificity), see (Fuhr et al., 2003, p. 184) or (Fuhr et al., 2004, 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 (highly exhaustive) with exact coverage (highly specific). We use version 1.8 of the INEX 2002 relevance assessments and version 2.4 of the INEX 2003 assessments. Evaluation is done using the `trec_eval` program.[1]

[1] Available from the TREC web site `http://trec.nist.gov` for registered participants.

For the task we are evaluating the `trec_eval` program implements the same metrics and produces the same results as the strict `inex_eval` program provided by the INEX initiative (Fuhr et al., 2003; Fuhr et al., 2004). 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. All our evaluations are based on the 1000 most relevant elements returned by our system.

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 nested 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 any stemming algorithm, but lower-case all text and remove stopwords.[2]

Our retrieval model is a multinomial language model with Jelinek-Mercer smoothing (Hiemstra, 2001). In addition, we have a tunable length prior. We estimate a language model for each element and for a given query we rank the elements with respect to the likelihood that they 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. We divide the task into two steps: estimating the prior probability of the element, $P(e)$, and estimating the probability of the query, given an element, $P(q|e)$. For the probability of the query, we use a linear interpolation of the probabilities of a query term using a language model of an element and the term probability using a language model of the collection. The probability of a query $t_1, \ldots, t_n$ is estimated as:

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

where P_{mle} denotes probabilities estimated using maximum likelihood estimation: $P_{mle}(t_i|e)$ is the probability of observing term t_i in element

[2] Following standard practice, we remove stopwords and lower-case before indexing. As an aside, some recent studies have explored sophisticated ways to include morphological variants in the document ranking process rather than as a preprocessing step (Kraaij, 2004); none succeeded in improving over the straightforward 'normalization as preprocessing' method.

e, and $P_{mle}(t_i)$ is the probability of observing term t_i in the collection. The smoothing parameter λ determines how much emphasis is put on the appearance of a query term in the element. For the prior probability, we explore several estimation methods, all based on estimating the connection between an element's length and its prior probability. We introduce a parameter β and estimate the prior probability of an element e as:

$$P(e) = \frac{(\sum_t tf(t,e))^\beta}{\sum_d (\sum_t tf(t,e))^\beta} \tag{3}$$

where $tf(t,e)$ is the frequency of term t in element e. 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 (the prior probability is proportional to the length); for $\beta = 2$ a squared length prior (the prior is proportional to the square of its length), for $\beta = 3$ a cubic length prior, etcetera.

The calculation of the probabilities can be reduced, in the standard way, to the scoring formula for an element e and query $t_1, \ldots, t_n$:

$$\begin{aligned} s(e, t_1, \ldots, t_n) \;=\; & \beta \cdot \log\left(\sum_t tf(t,e)\right) \\ & + \sum_{i=1}^{n} \log\left(1 + \frac{\lambda \cdot tf(t_i,e) \cdot (\sum_t df(t))}{(1-\lambda) \cdot df(t_i) \cdot (\sum_t tf(t,e))}\right), \end{aligned} \tag{4}$$

where $df(t)$ is the count of elements in which term t occurs, and λ is the weight given to the element language model when smoothing with the collection model. The first line of the sum represents the length prior and the second one the relevance of the element to the query. 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. The effect of the parameter β depends on the length of the query and the appropriate value must be determined empirically.

Another way to try to bridge the 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 (Gövert et al., 2003) or by using existing relevance assessments (Mass and Mandelbrod, 2004). The other way is to restrict the view to elements that pass a certain length threshold, or cut-off N. We will explore the latter option since we want to explore retrieval methods that are independent of specific DTDs or

tag sets. We apply the index cut-off 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 in the same manner as the collection: the text is lower-cased and stop-words are removed.

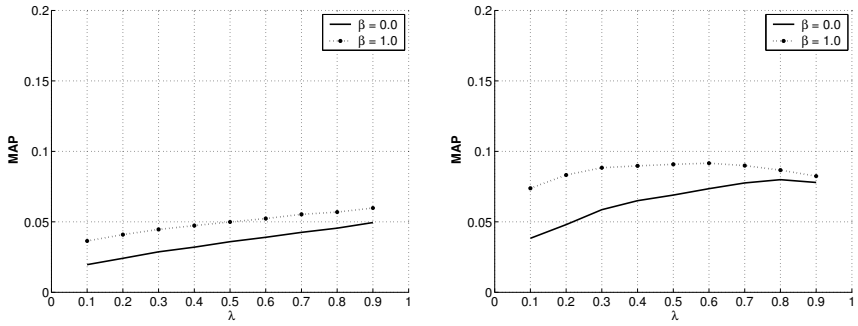
The three aspects introduced above (smoothing, length prior, and cut-off) affect length in different ways. The smoothing parameter, λ, indirectly introduces a length bias by increasing the importance of the presence of query terms in the retrieved element. The length prior's parameter, β, explicitly introduces a length bias proportional to the element length. The index-cut off indirectly favors the longer elements by removing the shortest elements from the index. These differences motivate us to investigate each aspect both in isolation and combined.

To determine the effect of smoothing we experiment with a range of values for the smoothing parameter λ from the interval $[0, 1]$.[3] To determine the importance of the length prior, β, we experiment with values ranging from 0.0 to 5.0. 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.

As we want to compare the effect of the length prior and the effect of index cut-off, we carry out 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 have the same aim; moving the attention away 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 our explorations to certain key values of the length prior and the cut-off parameters.

[3] We restrict our attention, in practice, to values in the range $[0.1, 0.9]$. This is, in general, the interesting range for the smoothing parameter. We do not show results for $\lambda = 0$ because it will rank document to the query-independent prior resulting in a very low score. We also do not show results for $\lambda = 1$ for it is not defined in the our scoring formula 4 (although it is defined for equation 1).



(a) INEX 2002 topics (b) INEX 2003 topics

Figure 3. Mean average precision for different values of the smoothing parameter λ. The lines show scores with and without using the traditional length prior.

5. Results and Discussion

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 (Efron, 1979; Efron and Tibshirani, 1993). The method has previously been applied to retrieval evaluation by, e.g., (Wilbur, 1994) and (Savoy, 1997). 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

Retrieval performance is generally sensitive to the value given to smoothing parameters. Smoothing is applied to account for data-sparseness and is therefore considered more useful for short text units than longer ones. The data-sparseness problem is particularly evident in a collection of very short texts, such as our collection of XML elements. This leads us to investigate the relation between smoothing settings, retrieval settings and element length. Figure 3 shows the mean average precision scores for different values of the smoothing parameter. The two lines are scores with and without using the normal language model length prior, i.e., using $\beta = 1.0$ and $\beta = 0.0$ respectively. Table V summarizes results for optimal smoothing. We see that, for both topic sets, the optimal value of the smoothing parameter λ is in the higher end, which means that little smoothing is required. This is surprising since these

Table V. Comparison between different retrieval methods and topic sets.

	λ	β	N	MAP	Change
Normal ad hoc settings (2002)	0.2	1.0	0	0.0409	(baseline)
Optimal smoothing (2002)	0.9	1.0	0	0.0598	+46%***
Normal ad hoc settings (2003)	0.2	1.0	0	0.0832	(baseline)
Optimal smoothing (2003)	0.6	1.0	0	0.0916	+10%

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. The explanation of this outcome lies in the relationship between smoothing parameter and length of retrieved elements. 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 4. 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 (Hiemstra, 2001)). Since long elements are more likely to contain many of the query terms, high values of λ have a length bias effect. Recall from Section 3 that assessors favor the somewhat longer elements of the collection. Little smoothing, i.e., a high value of the smoothing parameter, serves those users well. Choosing a low value for the smoothing parameter, however, leads to retrieval of shorter, and perhaps, unwanted elements.

Comparing the 2002 and 2003 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 and Table II). 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.

5.2. Length Prior

Although the smoothing parameter can be used to control the length of retrieved elements, this function seems to be better suited for the length prior parameter. To determine the effect of the length prior we look at the curves for $\beta = 2.0$ and $\beta = 3.0$ in Figure 5. In the remainder of this paper, we will refer to either of these values of the length prior

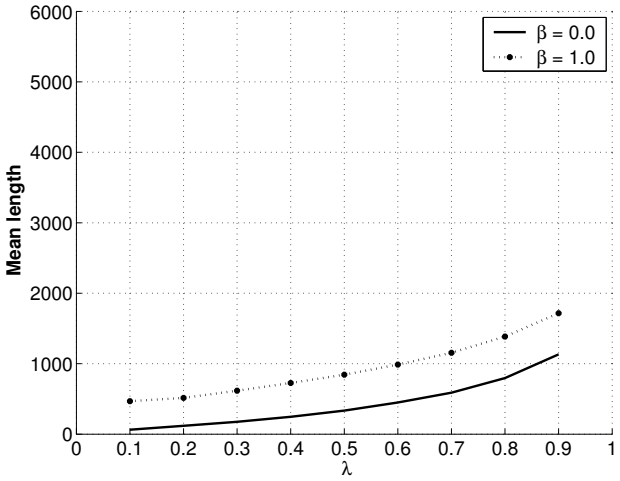
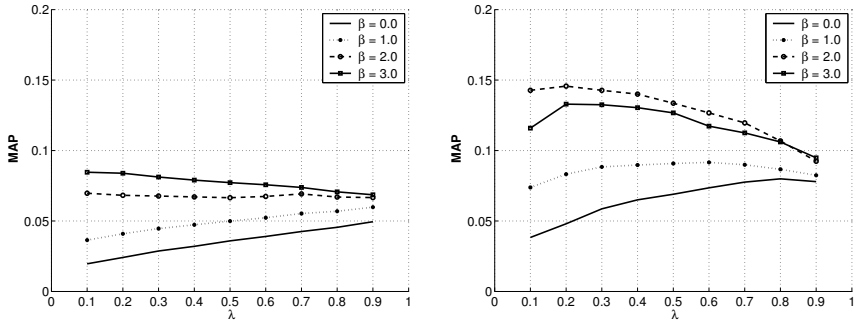


Figure 4. Average length of retrieved elements, with and without the normal length prior, plotted against the smoothing parameter λ. (Using 2002 topics.)



(a) INEX 2002 topics (b) INEX 2003 topics

Figure 5. Mean average precision for *extreme* length prior β, plotted against the smoothing parameter λ. Scores for the runs with and without the normal length prior are shown for comparison.

parameter as *extreme length prior*. Experiments with $\beta > 3.0$ resulted in a decrease in retrieval performance, and are not shown in detail. Table VI summarizes the results for the optimal settings for the extreme length prior. For both topic sets, the score improves significantly 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 role is taken over by the extreme length prior. This can be more clearly seen

Table VI. Comparison between different retrieval methods and topic sets.

	λ	β	N	MAP	Change
Normal ad hoc settings (2002)	0.2	1.0	0	0.0409	(baseline)
Extreme length prior (2002)	0.2	2.0	0	0.0682	$+67\%^{\star\star\star}$
	0.2	3.0	0	0.0839	$+105\%^{\star\star\star}$
Normal ad hoc settings (2003)	0.2	1.0	0	0.0832	(baseline)
Extreme length prior (2003)	0.2	2.0	0	0.1457	$+75\%^{\star\star\star}$
	0.2	3.0	0	0.1329	$+60\%^{\star}$

in Figure 6, which shows the average length of retrieved elements for different values of the smoothing parameter and length prior. We see that the length prior lifts the curve up to desired levels.

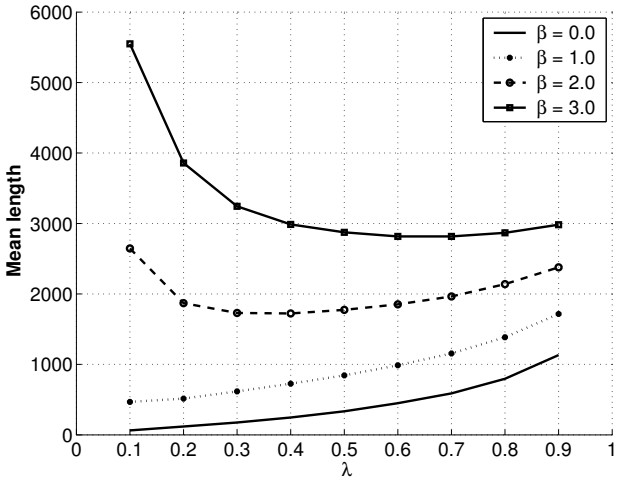
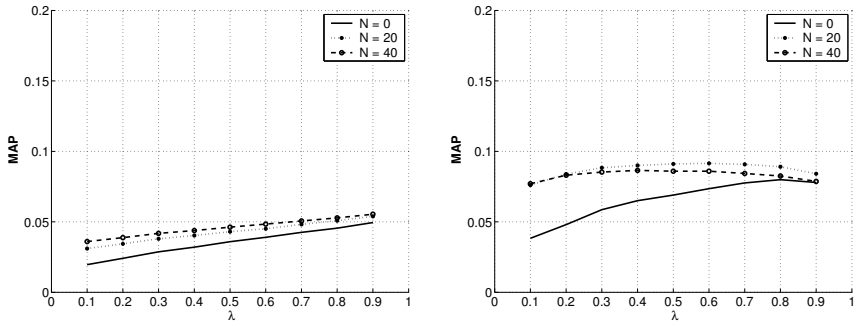


Figure 6. Average length of retrieved elements, for different values of the length prior parameter β, plotted against the smoothing parameter λ. (Using 2002 topics.)

Comparing the 2002 and 2003 topic sets we see that the optimal value for the length prior parameter 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-biases 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 set 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



(a) INEX 2002 topics (b) INEX 2003 topics

Figure 7. Mean average precision for different cut-off values N, plotted against the smoothing parameter λ.

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 an extreme length prior is 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 implicitly introduces a length bias. However, when we do use an extreme length prior, the smoothing parameter can go back to do what it does best, namely smoothing.

5.3. Element Length Cut-offs

The length prior has a dual effect: on the one hand it makes it effectively impossible to retrieve short elements, and on the other it influences the relative ranking of longer elements. Next, we investigate the relative importance of these two effects, restricting the minimal length 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 7 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. Table VII summarizes the results for the optimal settings for the index cut-off. There is not much difference

Table VII. Comparison between different retrieval methods and topic sets.

	λ	β	N	MAP	Change
Normal ad hoc settings (2002)	0.2	0.0	0	0.0409	(baseline)
Index cut-off (2002)	0.9	0.0	40	0.0551	+35%**
Normal ad hoc settings (2003)	0.2	0.0	0	0.0832	(baseline)
Index cut-off (2003)	0.6	0.0	20	0.0915	+10%

in performance between different cut-off values in the interval from 20 to 50. At cut-off values greater than 50, the performance starts to drop, since we are simply leaving out too many relevant elements from our index (see Figure 1).

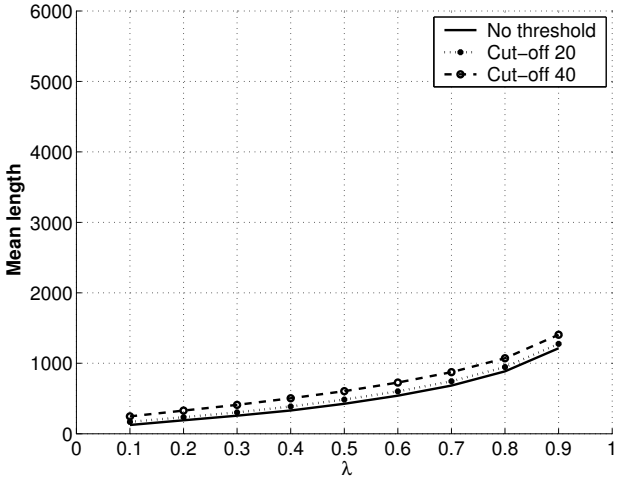


Figure 8. Average length of retrieved elements, for different cut-off values N, plotted against the smoothing parameter λ. (Using 2002 topics.)

For both topic sets, the cut-off improves scoring by far less than the extreme length prior. 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 8 shows the average length of retrieved elements for different cut-off values. It is interesting to note that the curves for the cut-off are very similar to the curve for the normal length prior in Figure 5. It is plausible that the reason why the normal length prior does not perform good enough is that its only effect might be to downplay the very many, 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 an extreme length prior and a cut-off can both lead to improved scoring, the two methods do not behave in the same way. While the extreme length prior introduces 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.

Table VIII. Comparison between different retrieval methods and topic sets.

	λ	β	N	MAP	Change
Normal ad hoc settings (2002)	0.2	1.0	0	0.0409	(baseline)
Cut-off + length prior (2002)	0.2	2.0	40	0.0781	+91%***
	0.2	3.0	40	0.0883	+115%***
Normal ad hoc settings (2003)	0.2	1.0	0	0.0832	(baseline)
Cut-off + length prior (2003)	0.2	2.0	40	0.1530	+84%**
	0.2	3.0	40	0.1364	+64%*

To demonstrate this effect we choose to apply the extreme length prior together with an index cut-off $N = 40$. This cut-off value is chosen based on experiments on the topic sets. There is, however, hardly any difference in performance between choosing different cut-off values in the interval from 20 to 50. Table VIII shows the results of applying an extreme length prior together with a cut-off. Combining a value of 3.0 for the length prior β and 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 extreme 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 extreme 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 extreme length prior ($\beta = 2.0$) alone. For the 2003 topic set the improvement is +84% (**) over the normal baseline and +5% (-) over the extreme 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

Table IX. Tag-names of elements retrieved using the baseline settings and optimized smoothing settings for both the INEX 2002 and 2003 collections

2002 collection				2003 collection			
baseline		optimal smoothing		baseline		optimal smoothing	
p	24.32%	article	22.15%	p	22.98%	p	19.69%
sec	11.88%	bdy	17.01%	sec	10.27%	sec	13.85%
atl	7.14%	sec	16.95%	ti	8.08%	article	10.19%
article	6.35%	p	11.87%	atl	7.67%	bdy	8.84%
ip1	5.86%	ss1	5.36%	ss1	5.16%	ss1	6.03%
ss1	5.61%	bm	3.88%	ip1	5.05%	ti	4.77%
bdy	5.49%	atl	2.99%	article	5.01%	atl	4.74%
bb	3.73%	ip1	2.91%	bdy	4.57%	ip1	4.47%
ti	2.91%	bibl	1.61%	bb	4.37%	bb	3.61%
it	2.21%	bib	1.61%	it	2.77%	bm	2.66%

extreme length prior. Combining an extreme length prior and a cut-off does improve over the use of an extreme length prior alone, but the improvement is statistically significant only for one of the two topic sets.

5.5. Tags returned

We have seen that the use of optimal smoothing settings, extreme length priors and appropriate cut-off settings result in significantly better retrieval results. That is, when we retrieve longer elements, there is a boost in retrieval effectiveness. The claim that longer elements are better than shorter ones does not necessarily mean that long articles are better than short articles or long sections better than short sections. The argument is rather that articles or sections are better than italicized phrases or article titles. We explore this effect by investigating the most frequent tag-names of elements retrieved using different settings.

Table IX shows the 10 most frequent element types returned for both normal settings and optimal smoothing settings. The baseline using normal settings returns a considerable amount of short elements, such as titles (⟨atl⟩ and ⟨ti⟩) and italicized text (⟨it⟩). However, the most frequent element types are paragraphs (⟨p⟩ and ⟨ip1⟩) and sections (⟨sec⟩ and ⟨ss1⟩). When the smoothing parameter is tuned we see an increasing weight of the longer elements such as articles (⟨article⟩) and bodies (⟨bdy⟩). Note that the optimal smoothing value for the two test sets is different. The 2002 assessment set was more strongly dominated

Table X. Tag-names of elements retrieved using extreme length prior, with and without cut-off.

2002 collection				2003 collection			
extreme prior		cut-off + ext. prior		extreme prior		cut-off + ext. prior	
article	37.80%	article	39.29%	article	16.94%	article	21.47%
bdy	25.79%	bdy	26.56%	p	15.65%	sec	17.91%
sec	12.97%	sec	13.91%	sec	14.10%	bdy	16.77%
p	5.46%	p	4.44%	bdy	13.42%	p	14.10%
ss1	3.85%	ss1	4.16%	ss1	5.60%	ss1	7.25%
bm	3.04%	bm	3.29%	atl	4.16%	bm	4.51%
ip1	1.30%	bibl	1.36%	ip1	3.56%	ip1	3.27%
bibl	1.18%	bib	1.36%	bb	3.12%	bibl	2.71%
bib	1.18%	ip1	1.11%	bm	3.10%	bib	2.71%
atl	0.85%	app	0.86%	ti	1.95%	app	1.35%

by articles than the 2003 assessment set, and thus gained from less smoothing. This is reflected in the elements returned. For the 2002 test set, when optimizing smoothing, the articles and bodies replace the paragraphs and sections at the top of the table. For the 2003 collection paragraphs and sections stay on top, but articles and bodies replace the two types of title elements.

Table X show the 10 most frequent element types returned by the runs using the extreme length prior. Results are shown for runs both with and without index cut-off. We see that using the extreme length prior, we get retrieval runs that are dominated by articles. The domination is less for the 2003 test set. Note that for each test set we report the more successful length prior settings, $\beta = 3.0$ for the 2002 set and $\beta = 2.0$ for the 2003 set. As to whether to not to use the index cut-off, we see that the difference is not large; however, we get rid of the very short title elements when we use a cut-off.

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. Earlier, Singhal et al. (1996) observed that, in TREC collections, "the likelihood of a document being judged relevant by a user increases with

the document length." Our analysis of data from INEX (Figure 1) and TREC (Figure 2) shows that the prior probability of relevance over length is comparable between document retrieval and XML retrieval. The main difference between XML retrieval and TREC-style document retrieval is the heavily skewed distribution of elements over length.

Within the language modeling framework, we investigated techniques that deal with length either directly or indirectly: length prior, index cut-off, and the amount of smoothing. We observed an implicit length bias introduced by the amount of smoothing, and showed the importance of an extreme length prior for XML retrieval. When used with an extreme length prior, 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 element length normalization. After restricting the minimal size of XML elements occurring in the index, the importance of an extreme length prior remains. The combination of an extreme length prior with an index cut-off does lead to a slight further improvement.

Although we find convincing evidence for our findings on the INEX collection, the usual disclaimers apply. As with any experimental result, there is no guarantee that these results will carry over to each and 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 also 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

We are grateful to our anonymous referees for their valuable comments. 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, 612.066.302, and 264-70-050.

References

Abolhassani, M., N. Fuhr, and S. Malik: 2004, 'HyREX at INEX 2003'. In (Fuhr et al., 2004), pp. 27–32.

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

Berger, A. and J. Lafferty: 1999, 'Information retrieval as statistical translation'. In: *Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 222–229, ACM Press.

Buckley, C., A. Singhal, and M. Mitra: 1996, 'New Retrieval Approaches Using SMART: TREC 4'. In: D. K. Harman (ed.): *The Fourth Text REtrieval Conference (TREC-4)*. pp. 25–48.

Carmel, D., Y. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer: 2003, 'Searching XML documents via XML fragments'. In: C. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton (eds.): *Proceedings of the 26th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 151–158.

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

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

Fuhr, N., N. Gövert, G. Kazai, and M. Lalmas (eds.): 2003, 'Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2002)'. ERCIM.

Fuhr, N., M. Lalmas, and S. Malik (eds.): 2004, 'INEX 2003 Workshop Proceedings'.

Gövert, N., M. Abolhassani, N. Fuhr, and K. Grossjohan: 2003, 'Content-based XML retrieval with HyRex'. In (Fuhr et al., 2003), pp. 26–32.

Greiff, W. and W. Morgan: 2003, 'Contributions of Language Modeling to the Theory and Practice of Information Retrieval'. In: W. Croft and J. Lafferty (eds.): *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, pp. 73–93.

Harman, D.: 2003, 'Overview of the TREC 2002 Novelty Track'. In: E. Voorhees and L. Buckland (eds.): *The Eleventh Text REtrieval Conference (TREC-11)*.

Hiemstra, D.: 2001, 'Using Language Models for Information Retrieval'. Ph.D. thesis, University of Twente.

Hiemstra, D.: 2003, 'A Database Approach to Content-based XML Retrieval'. In (Fuhr et al., 2003), pp. 111–118.

Hiemstra, D. and W. Kraaij: 1999, 'Twenty-One at TREC-7: Ad-hoc and cross-language track'. In: E. Voorhees and D. Harman (eds.): *The Seventh Text REtrieval Conference (TREC-7)*. pp. 227–238.

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

Kamps, J., M. de Rijke, and B. Sigurbjörnsson: 2003a, 'Topic Field Selection and Smoothing for XML Retrieval'. In: A. P. de Vries (ed.): *Proceedings of the 4th Dutch-Belgian Information Retrieval Workshop*. pp. 69–75.

Kamps, J., M. de Rijke, and B. Sigurbjörnsson: 2004, 'Length Normalization in XML Retrieval'. In: *Proceedings 27th Annual International ACM SIGIR Conference (SIGIR 2004)*. pp. 80–87.

Kamps, J., M. Marx, M. de Rijke, and B. Sigurbjörnsson: 2003b, 'XML Retrieval: What to Retrieve?'. In: C. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton (eds.): *Proceedings of the 26th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 409–410.

Kraaij, W.: 2004, 'Variations on Language Modeling for Information Retrieval'. Ph.D. thesis, University of Twente.

Kraaij, W., R. Pohlmann, and D. Hiemstra: 2000, 'Twenty-One at TREC-8: using language technology for information retrieval'. In: E. Voorhees and D. Harman (eds.): *The Eighth Text REtrieval Conference (TREC-8)*. pp. 285–300.

Kraaij, W. and T. Westerveld: 2001, 'Twenty-UT at TREC-9: How different are web documents?'. In: E. Voorhees and D. Harman (eds.): *The Ninth Text REtrieval Conference (TREC-9)*. pp. 665–672.

Kraaij, W., T. Westerveld, and D. Hiemstra: 2002, 'The importance of prior probabilities for entry page search'. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 27–34.

Lafferty, J. and C. Zhai: 2003, 'Probabilistic relevance models based on document and query generation'. In: W. Croft and J. Lafferty (eds.): *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, pp. 1–10.

List, J. and A. de Vries: 2003, 'CWI at INEX 2002'. In (Fuhr et al., 2003), pp. 133–140.

List, J., V. Mihajlovic, A. D. Vries, G. Ramírez, and D. Hiemstra: 2004, 'The TIJAH XML-IR system at INEX 2003'. In (Fuhr et al., 2004), pp. 102–109.

Mass, Y. and M. Mandelbrod: 2004, 'Retrieving the most relevant XML components'. In (Fuhr et al., 2004), pp. 53–58.

Mass, Y., M. Mandelbrod, E. Amitay, D. Carmel, Y. Maarek, and A. Soffer: 2003, 'JuruXML – an XML retrieval system at INEX'02'. In (Fuhr et al., 2003), pp. 73–80.

Miller, D., T. Leek, and R. Schwartz: 1999, 'A hidden Markov model information retrieval system'. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 214–221.

Ogilvie, P. and J. Callan: 2003, 'Language Models and Structured Document Retrieval'. In (Fuhr et al., 2003), pp. 33–44.

Ogilvie, P. and J. Callan: 2004, 'Using Language Models for flat text queries in XML retrieval'. In (Fuhr et al., 2004), pp. 12–18.

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

Savoy, J.: 1997, 'Statistical Inference in Retrieval Effectiveness Evaluation'. *Information Processing and Management* **33**, 495–512.

Sigurbjörnsson, B., J. Kamps, and M. de Rijke: 2004, 'An Element-Based Approch to XML Retrieval'. In (Fuhr et al., 2004), pp. 19–26.

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

Voorhees, E.: 2003, 'Overview of the TREC 2002 Question Answering Track'. In: E. Voorhees and L. Buckland (eds.): *The Eleventh Text REtrieval Conference (TREC-11)*.

Wilbur, J.: 1994, 'Non-Parametric Significance Tests of Retrieval Performance Comparisons'. *Journal of Information Science* **20**, 270–284.

Wilkinson, R.: 1994, 'Effective retrieval of structured documents'. In: *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 311–317, ACM Press.

Zhai, C. and J. Lafferty: 2001, 'A study of smoothing methods for language models applied to ad hoc information retrieval'. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 334–342.