The Importance of Morphological Normalization for XML Retrieval

Jaap Kamps, Maarten Marx, Maarten de Rijke, Börkur Sigurbjörnsson
Language and Inference Technology Group, ILLC, U. of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands
{kamps,marx,mdr,borkur}@science.uva.nl
http://lit.science.uva.nl

Abstract

Current information retrieval systems typically ignore structural aspects of documents, solely focusing on the textual content instead. But documents containing additional structure in the form of HTML, XML, or SGML mark-up are pervasive on the Internet. The XML retrieval task presents a number of challenges for information retrieval, for we can no longer rely on the appropriate unit of retrieval to be fixed, or to be known beforehand. This implies that the effectiveness of standard IR techniques, such as morphological normalization methods, may not carry over to this particular task. This paper describes the fully automatic runs for the INEX 2002 task submitted by the Language and Inference Technology Group at the University of Amsterdam. We investigate the effectiveness of two standard approaches to morphological normalization, both a linguistically motivated stemming algorithm and a knowledge-poor character n-gramming technique. Our results show that morphological normalization is an important issue for XML retrieval. For all measurements, the combined run and the n-gram run perform better than the stemmed run.

1 Introduction

With recent advances in computer and Internet technology, people have access to more information than ever before. Much of the information is available in free text with little or no metadata, and there is a tremendous need for tools to help organize, classify, and store the information, and to allow better access to the stored information. Current information retrieval systems allow us to locate documents that might contain the pertinent information, but most of them leave it to the user to extract the useful information from a ranked list. This leaves the (often unwilling) user with a relatively large amount of text to consume.

To address these issues, a number of recent initiatives are aimed at providing highly focused information 'pinpointing.' For instance, in the TREC question-answering track [17] participants are given a large document set and a set of questions; for each question, the system has to return an exact answer to the question and a document that supports that answer. Another approach to providing highly focussed information access is to return only new *and* relevant sentences (within context) rather than whole documents containing duplicate and extraneous information, as is done within TREC's novelty track [5].

We view XML retrieval as yet another approach to providing more focused information access than traditionally offered by search engines. An XML document collection differs from a traditional IR document collection: in the latter, documents contain only plain text and they are the natural unit of retrieval. Documents in an XML collection are divided into a hierarchy of text objects. These text objects provide restricted and, we hope, semantically meaningful contexts for satisfying users' information needs. It is natural, therefore, to take advantage of this structural information and look below the document level for a suitable unit of retrieval. The main question then becomes: To which extent can XML document structure help improve retrieval effectiveness? Obviously, the creation of an XML test collection is a key resource for answering this question.

The INEX 2002 collection, 21 IEEE Computer Society journals from 1995–2002, consists of $12,135$ documents with extensive XML-markup (when ignoring the volume.xml files). The test collection contains two types of topics. Content-only topics (CO) ignore the structure of the documents and, hence, are nothing but traditional IR topics. Content-and-structure (CAS) topics are aware of the structure of the documents. They can include constraints on the type of elements that are to be retrieved as well as constraints on the context in which the search terms should appear. The main difference with traditional IR tasks is that we may retrieve any XML component in the collection.

The aim of our official runs was to experiment with the effectiveness of different types of morphological normalization for structured corpora. The XML retrieval task departs from the strict boolean query matching used in traditional database theory, allowing for various gradations of relevance. In particular, related words like morphological variants (singular, plural, etc.) should share some of their relevance. Morphological normalization proved successful for plain text collections [8, 12]. In order to study the impact of morphological normalization in the setting of XML retrieval, we created stemmed and n-grammed indexes that preserve the XML-structure of the original documents. This allows for both the CO and CAS topics to be evaluated against both indexes.

Our strategy at INEX 2002 was to create a baseline system based on a traditional document index. That is, our index treats complete articles as the unit for retrieval. For the CO topics, the XML structure of the documents was not used, and we retrieve entire articles. For the CAS topics, we used a two step strategy. We first treated the topic as a CO topic and selected the 1000 highest ranking articles. Then we directly processed the (morphologically normalized) representation of these documents. All experiments were carried out with the FlexIR system developed at the University of Amsterdam [12].

The rest of this paper is organized as follows. We describe our experimental set-up in Section 2, and our official runs in Section 3. In Section 4 we present evaluation measures for XML retrieval and present our results. Section 5 provides a discussion of our results, and we end by drawing some conclusions.

2 Experimental Set-Up

2.1 The FlexIR information retrieval system

All submitted runs used FlexIR, an information retrieval system developed at the University of Amsterdam [12]. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl; it is built around the standard UNIX pipeline architecture, and supports many types of preprocessing, scoring, indexing, and retrieval tools, which proved to be a major asset for the INEX task. The retrieval model underlying FlexIR is the standard vector space model. All our runs used the Lnu.ltc weighting scheme [1] to compute the similarity between a query and a document; we fixed $slope$ at 0.2, while the pivot was set to the average number of unique words per document.

From both topics and documents we removed words occurring on a standard stop list with 391 words. Blind feedback was applied to expand the original query with related terms. Term weights were recomputed by using the standard Rocchio method [14], where we considered the top 10 documents to be relevant and the bottom 500 documents to be non-relevant. We allowed at most 20 terms to be added to the original query.

We experimented with two approaches to morphological normalization (discussed in Section 2.2 below). As a side issue, we wanted to experiment with combinations of (what we believed to be) different kinds of runs in an attempt to determine their impact on retrieval effectiveness. First, we normalized the retrieval status values (RSVs), since different runs may have radically different RSVs. Following [10], we mapped the values to $[0, 1]$ using $RSV'_i = (RSV_i - min_i)/(max_i - min_i)$. Next, we assigned new weights to the documents using a linear interpolation factor λ representing the relative weight of a run [15]: $RSV_{new} = \lambda \cdot RSV'_1 + (1 - \lambda) \cdot RSV'_2$. For $\lambda = 0.5$ this is the combSUM function of [3].

2.2 Morphological normalization

As pointed out above, our overall aim was to study the effect of morphological normalization on the effectiveness of XML retrieval. One approach to morphological normalization is to use linguistically informed methods; we decided to use a stemming algorithm for the English language. Alternatively, there are knowledge-poor approaches to morphological normalization which do not require any knowledge of the particular source language; here, we decided to use an n-gramming method.

n-Grams Our n-gram-based approach was based on character n-grams, where the n-gram length was set to 5; this setting was motivated by the results of experiments on the CLEF [2] data sets. For each word we stored both the word itself and all possible character n-grams of length 5 that can be obtained from it without crossing word boundaries. As an example, Figure 1(a) shows the original Topic 31, and Figure 1(b) shows the (stopped and) n-grammed version of the topic.

Stemming For the linguistically informed method with which we wanted to contrast the effect of the n-gram method we used Porter stemming [13]. Figure 1(c) shows the (stopped and) stemmed version of Topic 31.

```
<INEX-Topic topic-id="31" query-type="CO" ct-no="003">
   <Title>
      <cw>computational biology</cw>
   </Title>
   <Description>
      Challenges that arise, and approaches being explored, in the interdisciplinary
      field of computational biology.
   </Description>
   ...
</INEX-Topic>
```

(a) The original version of Topic 31.

```
.i 31
computational compu omput mputa putat utati tatio ation tiona ional biology biolo iolog ology
challenges chall halle allen ... biology biolo iolog ology
```

(b) The n-grammed version of Topic 31.

```
.i 31
comput biologi challeng aris approach explor interdisciplinari field comput biologi
```

(c) The stemmed version of Topic 31.

Figure 1: Topic 31.

3 Runs

We now describe how our runs were created. We built two base runs: one using the Porter stemmer and one in which we used n-grams in the manner described above. We then combined these two runs in the manner described in Section 2, thus producing a total of three official runs for INEX 2002:

Stemmed run We use a stemmed index and stemmed topics, the Lnu.ltc weighting scheme, and blind feedback.

n-Grammed run We use an n-grammed index and n-grammed topics, the Lnu.ltc weighting scheme, and blind feedback. We used n-gram-length 5, adding n-grams for words with length ≥ 4, while also keeping the originals words.

Combined run We combined the first two runs using an interpolation factor λ of 0.6 for the n-gram run. This higher weight for the n-gram run was motivated by the outcomes of experiments on the CLEF [2] data sets.

For both types of topics we wanted to use methods that were fully automatic and portable to other collections. In our retrieval we only used words from the title and description fields. In particular, we did not use the keywords provided with the topics: according to the topic development guidelines, keywords are supposed to be "good scan words that are used in the collection exploration phase of the topic development process" [7, p.107]. Furthermore, we did not use any information from the DTD either.

After the (document) pre-processing steps described in Section 2 were carried out, indexing of the collection was done at the article level, i.e., the indices were mappings from terms to articles in the collection. Since the topic processing and retrieval steps differ for the CO topics on the one hand and the CAS topics on the other, we describe them in separate subsections.

3.1 Content-only topics

For the CO topics, we automatically translated the topics into the FlexIR topic format, as illustrated in Figure 1, using only the words appearing in the title and description fields.

We ran the (stemmed or n-grammed) topics against the (stemmed or n-grammed) document index. The 100 documents with the highest RSVs were returned. The units of retrieval were articles. In other words, we always returned `/article[1]` in the path tag of the results.

```
<INEX-Topic topic-id="01" query-type="CAS" ct-no="010">
   <Title>
      <te>article/fm/au</te>
      <cw>description logics</cw><ce>abs, kwd</ce>
   </Title>
   <Description>
      Retrieve the names of authors of articles on description logic, in particular
      articles in which the abstract or the list of keywords contains a reference
      to description logic.
   </Description>
   ...
</INEX-Topic>
```

(a) The original version of topic 01.

```
.i 01
descript logic retriev author articl descript logic particular articl abstract list keyword
contain refer descript logic
```

(b) Stemmed version of the document retrieval translation.

```
.i 01
article/fm/au
abs|kwd, descript logic
```

(c) Stemmed version of the document filtering translation.

Figure 2: Topic 01.

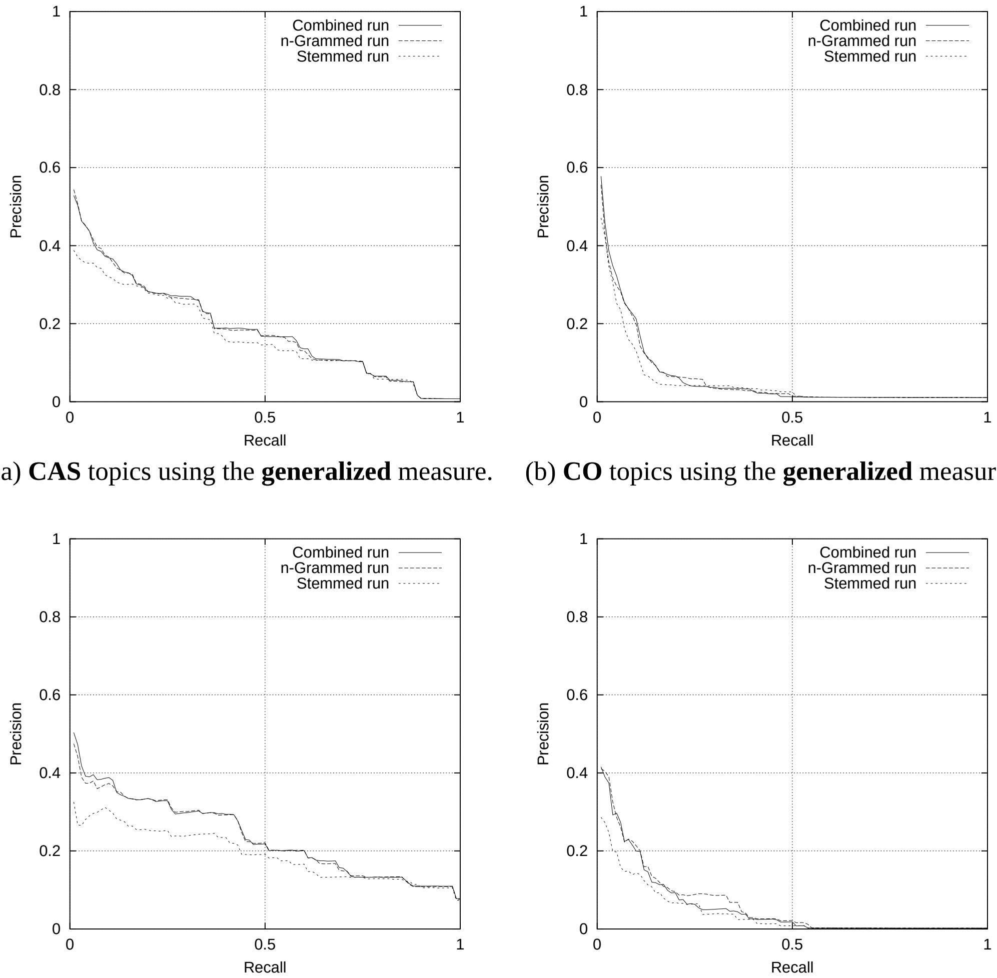
3.2 Content-and-structure topics

The CAS topics contain additional information in the `<ce>` and `<te>` tags; see Figure 2(a) for an example. For the CAS topics we divided the retrieval process into two subtasks: document retrieval and document filtering. This required two different topic translations, one for each subtask. For the document retrieval subtask, topics were processed similar to the CO topics: only the words in the title and description fields were selected, and from the title field we only selected the content of the `<cw>` field. For an example of this translation see Figure 2(b).

For the document filtering subtask, the `<Title>` field was processed to preserve the structural part of the query. For an example of this translation, see Figure 2(c). The first line contains the topic number, the second line gives the XML-field that is to be returned, the next line(s) give conditions for the document, consisting of a field name, and the words that are sought. This should be read as: retrieve the elements found by the XPath expression `//article/fm/au` in the documents whose elements found by the XPath expressions `//abs` or `//kwd` contain the words `descript` or `logic`. If no target element is specified in the topic title, we treat it as if the target element had been `<te>article</te>`. A connection between a disjunction of target elements and a disjunction of search criteria may lead to ambiguities. Hence we replaced disjunctions of target elements `<te>A,B,..</te>` by `<te>/article</te>`. Further motivation for this translation can be found in [11].

For the document retrieval subtask we ran the (stemmed or n-grammed) topics against the (stemmed or n-grammed) document index. The 1000 documents with the highest RSVs were returned. Our working assumption was that all relevant document were in this top 1000.

For the document filtering subtask, we created a special XML-file for each topic, containing these top 1000 documents. On these so-called doc-piles, we ran an XML-parser based on Perl's `XML::Twig` that handles XPath expressions. For each topic and for each context-element (`<ce>`) in its doc-pile, the XML-parser calculates a score for each context-element. This score is the count of how often a context-word (`<cw>`) appears in the context-element, divided by the number of words in the content-element. We sorted the documents in the doc-pile according to their highest scoring element. For each document in the doc-pile we extracted the target-elements (`<te>`), using the XML-parser. To each target-element we assign the score of the document that contains it. We select the 100 highest scoring target-elements. Those 100 elements are returned, sorted by RSV score of the document containing the element.



(a) **CAS** topics using the **generalized** measure. (b) **CO** topics using the **generalized** measure.

(c) **CAS** topics using the **strict** measure. (d) **CO** topics using the **strict** measure.

Figure 3: Precision recall graphs of our official runs for both topic types, using both evaluation measures.

4 Results

To evaluate our runs we used version 0.006 of the `inex_eval` program supplied by the organizers of INEX 2002. We used version 1.6 of the relevance assessments. The topics were assessed on a two dimensional graded relevance scale, one for topical relevance, with values taken from $\{0, 1, 2, 3\}$, and another for document coverage, with values taken from $\{$*exact, too_large, too_small, no_coverage*$\}$.

The evaluation software can create reports using two distinct measures, see [4] for details. The *strict* relevance measure considers only highly relevant items that have exact coverage. The strict relevance scores are calculated by means of the function f_s below.

$$f_s(e) := \begin{cases} 1 & \text{if } e = (3, exact) \\ 0 & \text{otherwise.} \end{cases} \qquad f_g(e) := \begin{cases} 1 & \text{if} \quad e = (3, exact) \\ 0.75 & \text{if} \quad e = (2, exact) \text{ or} \\ & \phantom{\text{if}} \quad e = (3, too\_large) \text{ or} \\ & \phantom{\text{if}} \quad e = (3, too\_small) \\ 0.5 & \text{if} \quad e = (1, exact) \text{ or} \\ & \phantom{\text{if}} \quad e = (2, too\_large) \text{ or} \\ & \phantom{\text{if}} \quad e = (2, too\_small) \\ 0.25 & \text{if} \quad e = (1, too\_large) \text{ or} \\ & \phantom{\text{if}} \quad e = (1, too\_small) \\ 0 & \text{otherwise} \end{cases}$$

The *generalized* relevance measure considers all combinations of all values of relevance and coverage. The generalized relevance scores are calculated by means of the function f_g given above.

The strict and generalized measures defined above differ from the standard mean average precision scores. When ignoring the coverage dimension, the strict measure is similar to the work on judging by highly relevant document [16]. This strict measure is still a dichotomous measure. When ignoring coverage, the generalized measure is similar to the graded measures of relevance [9].

Generalized measure CAS				
Run	MAP	Impr.	P. at 0	Impr.
Combined run	**0.185**	+12%	0.528	+36%
n-Grammed run	0.183	+11%	**0.544**	+40%
Stemmed run	0.165	0%	0.388	0%

Generalized measure CO				
Run	MAP	Impr.	P. at 0	Impr.
Combined run	**0.0576**	+19%	**0.578**	+23%
n-Grammed run	0.0568	+17%	0.556	+18%
Stemmed run	0.0484	0%	0.471	0%

Strict measure CAS				
Run	MAP	Impr.	P. at 0	Impr.
Combined run	**0.234**	+23%	**0.503**	+55%
n-Grammed run	0.232	+21%	0.475	+46%
Stemmed run	0.191	0%	0.325	0%

Strict measure CO				
Run	MAP	Impr.	P. at 0	Impr.
Combined run	0.0553	+34%	**0.415**	+45%
n-Grammed run	**0.0618**	+55%	0.411	+44%
Stemmed run	0.0399	0%	0.286	0%

Table 1: The mean average precision results for our official runs. The precision at zero is the interpolated precision over the interval $(0, 0.1]$. Improvements are computed relative to the stemmed run.

The results for our official runs are displayed in Figure 3 and Table 1. Some obvious remarks can be made. First, compared to TREC-style document retrieval results, the mean average precision (MAP) scores are much lower (at TREC where one would expect a MAP of at least twice the best score in the table). Also, the scores for CO are much lower than for CAS topics. Second, we included the precision at 0 in Table 1 as an indication of the quality of the top ranked retrieved documents. These numbers are reassuring, and far less dramatic than the low MAP scores for, especially, CO would suggest. In fact, both CAS and CO topics have comparable p@0 scores. Third, the difference in performance of the three runs is a clear indication that morphological normalization is an important issue for XML retrieval. The relative results are in favor of the knowledge-poor approach: the n-grammed run is performing better than the stemmed run in all four cases. Fourth, the combined run is better than the best underlying baserun in three cases (CAS and CO generalized), although the improvement is unimpressive. This can be explained by the difference in score of the underlying baseruns: when the difference between stemmed and n-grammed runs peaks at over 50% (CO strict), the combined run is not better than the n-gram run! Fifth, when comparing the strict and generalized scores, the strict scores are almost always higher. This is somewhat counterintuitive, because the generalized score is a more liberal score that regards more retrieved elements as relevant.

5 Discussion and Conclusions

We entered the INEX initiative for the evaluation of XML retrieval with modest ambitions. We wanted to set up a baseline system based on a traditional document index where the unit of retrieval is an article. Only for the CAS topics did we attempt to retrieve the particular XML element requested by the target element field.

Our goal was to have a fully automatic XML retrieval system that can easily be ported to different topics, collections, and DTDs. All our runs are fully automatic TD-runs that ignore the keywords and the narrative fields of the topics (which are considered to be additional information for the relevance judgments). We did not correct misspellings or other errors in the topics, resulting in the retrieval of no results for two CAS topics. We use no manual query processing steps, nor human knowledge on the semantics of the tags.

We expected our system's performance to be just a baseline for 'proper' XML retrieval systems, i.e., for systems that return smaller XML components than articles. To our surprise, our runs turn out to be among the top scoring submissions on both CAS and CO tasks, and on both generalized and strict evaluation measures; this is even more surprising if we take into account that several teams submitted manual runs and runs using the narrative. How should we interpret this? On the one hand, the results show that a system returning entire articles is competitive to systems returning smaller units of text—our system, indeed, can function as the baseline performance we hoped to obtain. On the other hand, the results suggest that we do not yet fully understand how users (and assessors) perceive the coverage dimension of relevance. It is clear that more research is needed to better understand what users (and assessors) regard as meaningful units of retrieval.

There are a few things one needs to keep in mind when looking at the output of the `inex_eval` software. The software's definition of total recall does not take into account the graded relevance nor the limit on the number of elements retrieved. The total recall of the strict measure is defined as the number of highly relevant elements in the collection that have exact coverage. The total recall of the generalized measure is defined as the number of relevant elements in the collection. This puts an upperbound on the mean average precision scores that systems can achieve, as shown in Table 2; the upperbounds are calculated for 'perfect' run that return 100 relevant items.[1]

These upperbounds partly explain why the strict evaluation measure gives a higher average precision than the generalized measure. This is counter-intuitive as we would expect to do worse on the strict scale, having in mind that we do article retrieval for all the CO topics and approximately one-third of the CAS topics. Thus we would expect a too_large coverage, giving no score on the strict measure. When taking into account the maximally obtainable scores in Table 2, our generalized scores do outperform the strict scores. Added to that, whole articles seem to have been quite frequently judged highly relevant with exact coverage. This sheds some light on how exact coverage is perceived by users and assessors.

Topic type	Measure	Possible MAP
CAS	generalized	0.596
CO	generalized	0.332
CAS	strict	0.897
CO	strict	0.931

Table 2: Upper bounds on the average precision.

The official runs of INEX 2002 had a maximum number of retrieved elements set at 100 elements. A problem with this upperbound is that the number of relevant elements in the assessments can be much higher than 100, even on average. We modified our runs by allowing 1000 results to be returned (as is customary for CLEF and TREC ad-hoc retrieval experiments). A comparison of the MAP scores between runs with cut-off points at 100 and 1000 results is displayed in Table 3. Although the scores do improve, they remain low compared to MAP values for unstructured documents. The improvement is higher for the generalized measure than for the strict measure. This may be due to the larger set of relevant items for the generalized measure. This may also explain why the improvement is greater for CO topics than for CAS topics, although this is partly caused by the lower score of the top-100 runs.

Generalized measure CAS			
	MAP		
Run	100	1000	Impr.
Combined run	**0.185**	**0.199**	+7.6%
n-Grammed run	0.183	0.196	+7.1%
Stemmed run	0.165	0.170	+3.0%

Generalized measure CO			
	MAP		
Run	100	1000	Impr.
Combined run	**0.0576**	**0.0677**	+18%
n-Grammed run	0.0568	0.0653	+15%
Stemmed run	0.0484	0.0551	+14%

Strict measure CAS			
	MAP		
Run	100	1000	Impr.
Combined run	**0.234**	**0.244**	+4.3%
n-Grammed run	0.232	0.240	+3.4%
Stemmed run	0.191	0.201	+5.2%

Strict measure CO			
	MAP		
Run	100	1000	Impr.
Combined run	0.0553	0.0609	+10%
n-Grammed run	**0.0618**	**0.0657**	+6.3%
Stemmed run	0.0399	0.0427	+7.0%

Table 3: Comparison of MAP scores for 100 and 1000 retrieved elements.

Our aim was to study the effect of morphological normalization for XML retrieval. We experimented with two distinct approaches to morphological normalization: by using linguistically informed methods and by using knowledge poor techniques. For the former we used the familiar Porter stemming algorithm for English. For the latter, we used character n-grams of length 5. Our results show a clear difference between the two approaches, which suggests that morphological normalization is an important issue for XML retrieval. Our results favor the knowledge-poor approach of n-gramming. For all measurements, the combined run and the n-gram run perform better than the stemmed run. This is consistent with results on plain text collections [6, 12]. We also experimented with the combination of the two approaches to morphological normalization. The combined runs score best in three out of four cases (CAS and CO generalized). Still, there is no remarkable difference between the combined run and the n-gram run; n-gramming seems to be the dominant factor of the combination, which, again, is consistent with the retrieval results for unstructured documents [8].

Using our INEX 2002 runs as a baseline, our future research focuses on how to retrieve smaller units of texts by

[1] For the strict measure, a perfect run without length restriction will score a MAP of 1.0; for the generalized measure, a perfect run cannot obtain the perfect score of 1.0. This is due to the definition of generalized recall [9, p.1123]. For example, if there are two relevant documents for a topic with relevance scores 1 and 0.5, respectively, then the generalized precision at generalized recall level 1 is only 0.75.

treating each tag occurring in the collection as a document by itself. Next to this, we are experimenting with ways of exploiting the collection's structure for improving retrieval on the article level, by considering the keywords assigned to documents, co-authors, citations, co-citations, etc. Finally, we are investigating efficient storage and processing architectures tailored to structured document collections.

Acknowledgments We want to thank Willem van Hage for his technical support and for assessing the Enigma topic 42. Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO), grant 400-20-036. Maarten Marx received support from NWO grant 612.000.106. 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, and 612.000.207.

References

[1] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48. NIST Special Publication 500-236, 1995.

[2] CLEF. Cross language evaluation forum, 2003. `http://www.clef-campaign.org/`.

[3] E. A. Fox and J. A. Shaw. Combination of multiple searches. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 243–252. National Institute for Standards and Technology. NIST Special Publication 500-215, 1994.

[4] N. Gövert and G. Kazai. Assessments and a preliminary evaluation metric for XML document retrieval. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *INEX 2002 Workshop Proceedings*, pages 117–120, 2002.

[5] D. Harman. Overview of the TREC 2002 novelty track. In E. M. Voorhees and D. K. Harman, editors, *The Eleventh Text REtrieval Conference (TREC 2002)*. National Institute for Standards and Technology, 2003.

[6] V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual retrieval for European languages. *Information Retrieval*, 6, 2003.

[7] INEX. INEX guidelines for topic development. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *INEX 2002 Workshop Proceedings*, pages 106–109, 2002.

[8] J. Kamps, C. Monz, and M. de Rijke. Combining evidence for cross-language information retrieval. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2002*, Lecture Notes in Computer Science. Springer, 2003.

[9] J. Kekäläinen and K. Järvelin. Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53:1120–1129, 2002.

[10] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188. ACM Press, New York NY, USA, 1995.

[11] M. Marx, J. Kamps, and M. de Rijke. The University of Amsterdam at INEX-2002. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *INEX 2002 Workshop Proceedings*, pages 24–28, 2002.

[12] C. Monz and M. de Rijke. Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2002.

[13] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[14] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System — Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[15] C. C. Vogt and G. W. Cottrell. Predicting the performance of linearly combined IR systems. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 190–196. ACM Press, New York NY, USA, 1998.

[16] E. M. Voorhees. Evaluation by highly relevant documents. In D. H. Kraft, W. B. Croft, D. J. Harper, and J. Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82. ACM Press, New York NY, USA, 2001.

[17] E. M. Voorhees. Overview of the TREC 2002 question answering track. In E. M. Voorhees and D. K. Harman, editors, *The Eleventh Text REtrieval Conference (TREC 2002)*. National Institute for Standards and Technology, 2003.