The University of Amsterdam at INEX 2004

Börkur Sigurbjörnsson Jaap Kamps[*] Maarten de Rijke

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
borkur,kamps,mdr@science.uva.nl

ABSTRACT

This paper describes the INEX 2004 participation of the Informatics Institute of the University of Amsterdam. We completely revamped our XML retrieval system, now implemented as a mixture language model on top of a standard search engine. To speed up structural reasoning, we indexed the collection's structure in a separate database. We address three research questions. First, we investigate the effectiveness of blind feedback based on top-ranking XML-elements. Second, we analyze the impact of removing overlapping elements in the result set. Third, for the content-and-structure topics, we want to compare the relative effectiveness of approaches that interpret the topic strict, or ignore the structural hints altogether. Experimental evidence is based on both of the INEX 2004 ad hoc tasks, content-only and content-and-structure, evaluated against a range of metrics.

1. INTRODUCTION

We follow an Information Retrieval (IR) approach to the Content-Only (CO) and Vague-Content-And-Structure (VCAS) ad hoc tasks at INEX. In our participation at INEX 2004 we build on top of our element-based approach at INEX 2003 [10], and extend our language modeling approach to XML retrieval.

Specifically, we addressed the following technological issues, mainly to obtain a statistically more transparent approach. For our INEX 2003 experiments we combined article and element scores outside our language model, meaning that we created a run based on an article index and one based on an element index, which were then combined using well-known run combination techniques [6]. This year we implemented a proper mixture language model for this combination. At INEX 2003 we estimated the language model for the collection by looking at statistics form our overlapping element index. For our experiments at INEX 2004 we estimate this collection model differently, by looking at statistics from our article index. The main changes in our blind feedback approach, compared to last year, is that this year we perform query expansion based on an element run, whereas last year we performed the expansion based on an article run. All our runs were created using the ILPS extension to the Lucene search engine [7, 3].

Our main research questions for both tasks were twofold. First, we wanted to investigate the effect of blind feedback on XML element retrieval. Second, we wanted to cast light on the problem of overlapping results; in particular, we investigate the effect of removing overlapping results top-down from a retrieval run. A third, additional research question only concerns the VCAS task: we investigate the difference between applying a content-only approach and a strict content-and-structure approach.

The remainder of this paper is organized as follows. In Section 2 we describe our experimental setup, and in Section 3 we provide details on the official runs we submitted to INEX 2004. Section 4 presents the results of our experiments, and in Section 5 we discuss our findings in the broader INEX context, and draw some initial conclusions.

2. EXPERIMENTAL SETUP

2.1 Index

Our approach to XML retrieval is IR-based. We create our runs using two types of inverted indexes, one for XML articles only and another for all XML elements. Furthermore, we maintain a separate index of the collection structure.

2.1.1 Article index

For the article index, the indexing unit is a complete XML document containing all the terms appearing at any nesting level within the ⟨`article`⟩ tag. Hence, this is a traditional inverted index as used for standard document retrieval.

2.1.2 Element index

For the element index, the indexing unit can be any XML element (including ⟨`article`⟩). For each element, all text nested inside it is indexed. Hence, the indexing units overlap (see Figure 1). Text appearing in a particular nested XML element is not only indexed as part of that element, but also as part of all its ancestor elements. The article index can be viewed as a restricted version of the element index, where only elements with tag-name ⟨`article`⟩ are indexed.

Both the article and the element index were word-based: we applied lower-casing and stop-words were removed using the stopword list that comes with the English version on the Snowball stemmer [12], but other than that words were indexed as they occur in the text, and no stemming was applied.

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

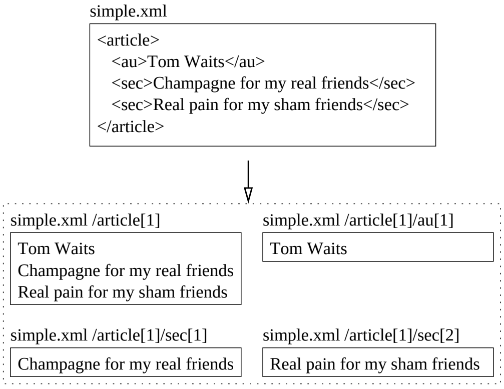
simple.xml
<article>
<au>Tom Waits</au>
<sec>Champagne for my real friends</sec>
<sec>Real pain for my sham friends</sec>
</article>
simple.xml /article[1]
Tom Waits
Champagne for my real friends
Real pain for my sham friends
simple.xml /article[1]/au[1]
Tom Waits
simple.xml /article[1]/sec[1]
Champagne for my real friends
simple.xml /article[1]/sec[2]
Real pain for my sham friends

Figure 1: Simplified figure of how XML documents are split up into overlapping indexing units.

2.1.3 Structure index

The structure of the collection is indexed using a relational database. To index the XML trees we use pre-order and post-order information of the nodes in the XML trees [1].

2.2 Query processing

For both the CO and the VCAS task we only use the ⟨`title`⟩ part of the topics. We remove words and phrases bounded by a minus-sign from the queries; other than that, we do not use the plus-signs, or phrase marking of the queries.

For the CAS topics we have a NEXI tokenizer which can decompose the query into a set of `about` functions [11]. If there is a disjunction in a location-path, we break it up into a disjunction of about functions. That is,

```
about(.//(abs|kwd), xml)
```

becomes

```
about(.//abs,xml) or about(.//kwd,xml).
```

If there are multiple about functions with the same scope we merge them into a single one. That is,

```
about(., broadband) or about(., dial-up)
```

becomes

```
about(., broadband dial-up).
```

For some of the VCAS runs we ignore the structural constraints and use only a collection of content query terms. That is, from the query

```
//article[about(.,sorting)]//sec[about(.,heap sort)]
```

we collect the query terms

```
sorting heap sort.
```

We will refer to these as the *full content queries*.

2.3 Retrieval model

All our runs use a multinomial language model with Jelinek-Mercer smoothing [2]. We estimate a language model for each of the elements. The elements are then ranked according to their prior probability of being relevant and the likelihood of the query, given the estimated language model for the element:

$$P(e|q) \propto P(e) \cdot P(q|e). \tag{1}$$

We assume that query terms are independent, and thus we rank our elements according to:

$$P(e|q) \propto P(e) \cdot \prod_{i=1}^{k} P(t_i|e), \tag{2}$$

where q is a query made out of the terms $t_1, \ldots, t_k$. To account for data sparseness we estimate the element language model by taking a linear interpolation of three language models: one for the element itself, one for the article that contains the element, and a third one for the collection. That is, $P(t_i|e)$ is calculated as

$$\lambda_e \cdot P_{mle}(t_i|e) + \lambda_d \cdot P_{mle}(t_i|d) + (1 - \lambda_e - \lambda_d) \cdot P_{mle}(t_i), \tag{3}$$

where $P_{mle}(\cdot|e)$ is a language model for element e; $P_{mle}(\cdot|d)$ is a language model for document d; and $P_{mle}(\cdot)$ is a language model of the collection. The parameters λ_e and λ_d are interpolation factors (smoothing parameters). We estimate the language models, $P_{mle}(\cdot|\cdot)$ and $P_{mle}(\cdot)$, using maximum likelihood estimation. For the element model we use statistics from the element index; for the document model we use statistics from the article index; and for the collection model we use document frequencies from the article index.

The language modeling framework allows us to easily model non-content features. One of the non-content that proved to be useful during our experiments for INEX 2003 is document length. Specifically, we assign a prior probability to an element e relative to its length in the following manner:

$$P(e) = \frac{|e|}{\sum_e |e|}, \tag{4}$$

where $|e|$ is the size of an element e.

2.4 Query Expansion

We have been experimenting with blind feedback in all editions of INEX so far, focusing on query expansion for the content-only task exclusively. Initially, we experimented with Rocchio-style reweighting to select up to 10 terms from the top 10 documents [9]. In INEX 2002 we observed that query expansion with Rocchio on the article index gave intuitively useful expanded queries, leading to the kind of improvements that familiar from article retrieval [5]. However, expanding queries based on the top 10 retrieved XML elements seemed not to work due to the short and overlapping elements in the top 10 results. Hence, we decided to expand queries on the article index, and then run the expanded queries against the element index. This did, indeed, give us a boost for the 2002 topics, but, alas, substantially lowered our score for the 2003 topics [11].

Our analysis of the failure of article-index based feedback in INEX 2003 was that the terms were useful, but highly unlikely to occur in the proper element. An example is getting prominent author names from the bibliography, which are relevant and useful retrieval cues but generally do not appear in a paragraph (maybe in the author field, or the bibliography).[1]

We decided to go back to the idea of doing blind feedback directly on the XML element index. This has the advantage of conser-

[1] We have been planning to incorporate context (i.e., tags in which term occurs) into our model, but this would requires some CAS features for the CO runs that are non-trivial to implement.

Run-id	λ_e	λ_d	units	terms	% overlap
UAms-CO-T	0.1	0.3	–	–	71.96
UAms-CO-T-FBack	0.1	0.3	15	5	81.85
UAms-CO-T-FBack-NoOverl	0.1	0.3	15	5	0.00

Table 1: Overview of our official content-only runs for INEX 2004. All runs are *automatic* runs, that only use the T (title) topic field.

vatism, the initially retrieved top 10 elements will keep their high ranking, but the problem of overlap in the initial result set remains. In pre-submission experiments, the language modeling approach to feedback [8] proved more robust, and improved performance on the 2003 topics.

3. RUNS

In this section we describe the official runs submitted by the University of Amsterdam for INEX 20004.

All our runs use the language modeling framework described in the previous section. For all runs we use a two level smoothing procedure: we smooth against both the article and the collection. Our collection model uses the document frequencies from the article index. For computing the likelihood of a term given an element, see Equation 3, we use the following parameter settings for all runs: $\lambda_e = 0.1$ and $\lambda_d = 0.3$. All runs also use the same length prior settings in Equation 4.

3.1 Content-Only task

Table 1 provides an overview of our CO runs. We now describe the specifics of each of the CO runs.

UAms-CO-T

This run uses the mixture language model approach and parameter settings as described above.

UAms-CO-T-FBack

This run uses the same model and parameters as the previous run. Additionally, this run uses blind feedback to expand the queries. An element run was used as a basis for our feedback. We considered the top 15 elements to be relevant and chose the 5 best query terms as described in [8].

UAms-CO-T-FBack-NoOverl

This run uses the same model, parameters and feedback approach as the previous run. Additionally, overlapping results are filtered away. The filtering is done in a top-down manner. That is, the result list is processed from the most relevant to the least relevant element. A result is removed from the result list if it overlaps with an element that has been processed previously.

3.2 Vague Content-And-Structure task

We now describe our VCAS runs; again, we provide a table with an overview; cf. Table 2.

UAms-CAS-T-FBack

This run uses the full-content version of the queries. The run is identical to UAms-CO-T-FBack, except for the topics, of course.

Run-id	λ_e	λ_d	units	terms	% overlap
UAms-CAS-T-FBack	0.1	0.3	15	5	77.76
UAms-CAS-T-FBack-NoOverl	0.1	0.3	15	5	0.00
UAms-CAS-T-XPath	–	–	–	–	18.77

Table 2: Overview of our official vague content-and-structure runs for INEX 2004. All runs are *automatic* runs, that only use the T (title) topic field.

UAms-CAS-T-FBack-NoOverl

This run uses the full-content version of the queries. The run is identical to UAms-CO-T-FBack-NoOverl, except for the topics, of course.

UAms-CAS-T-XPath

This run is created using our system for the INEX 2003 Strict Content and Structure task. It uses both content and structural constraints. Target constraints are interpreted as strict. We refer to [11] for a detailed description of the retrieval approach used. The run is identical to the run referred to as "Full propagation run" in that paper.

4. RESULTS

In this section we will try to analyze the results of our retrieval efforts. Result analysis for XML retrieval remains a difficult task: there are still many open questions regarding how to evaluate XML element retrieval. We will show our results for all the suggested measures and try to interpret the flow of numbers.

4.1 Content-Only task

Table 3 shows the results for our CO runs, using all different metrics. We see that the run which uses blind feedback outperforms the normal run on all metrics except for the measures where high exhaustivity is rewarded. Hence, at first glance, it seems thus that blind feedback does more for the specificity of the results than for the exhaustiveness of the results. This seems somewhat counterintuitive and we will discuss it further below. The run where overlap was removed does not score well for any metric.

Figure 2 shows precision-recall curves for our CO runs for three measures: strict, generalized measure, specificity oriented (so). Query expansion gives improvements on all recall levels. The normal and non-overlapping runs have similar precision at zero, but the non-overlapping run quickly drops. The non-overlapping run simply fails to retrieve many of the relevant elements. This comes as no surprise since the relevant elements, themselves, are frequently overlapping.

	Run		
Measure	CO-T	CO-T-FBack	CO-T-FBack-NoOverl
aggregate	0.1030	*0.1174*	0.0270
strict	0.1013	*0.1100*	0.0332
generalized	0.0929	*0.1225*	0.0198
so	0.0717	*0.1060*	0.0149
s3_e321	0.0528	*0.0877*	0.0148
s3_e32	0.0668	*0.0891*	0.0168
e3_s321	*0.1840*	0.1699	0.0507
e3_s32	*0.1515*	0.1368	0.0387

Table 3: Average scores for our CO runs, with this best scoring run in italics.

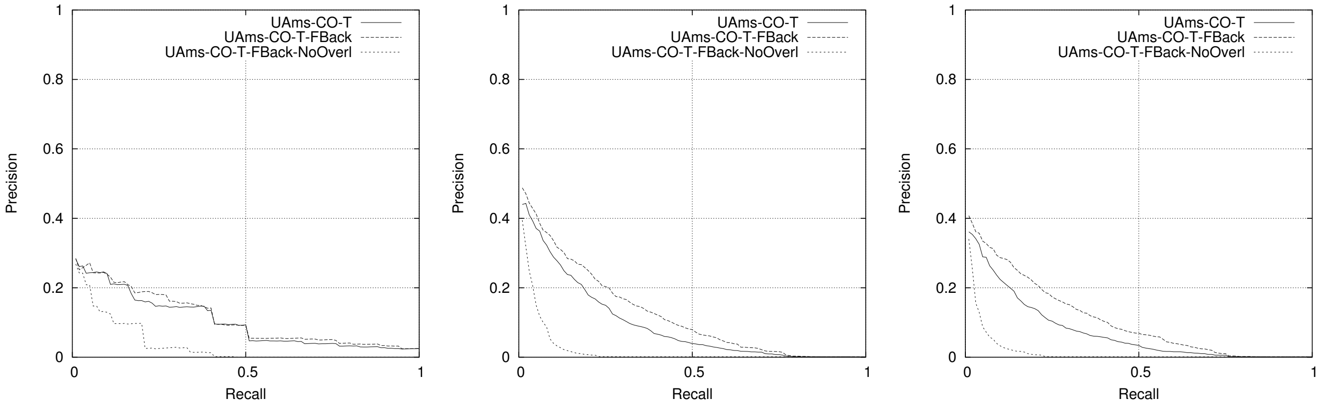

Figure 2: Precision-recall curves for our CO runs. (Left): strict measure. (Center): generalized measure. (Right): specificity oriented (so) measure.

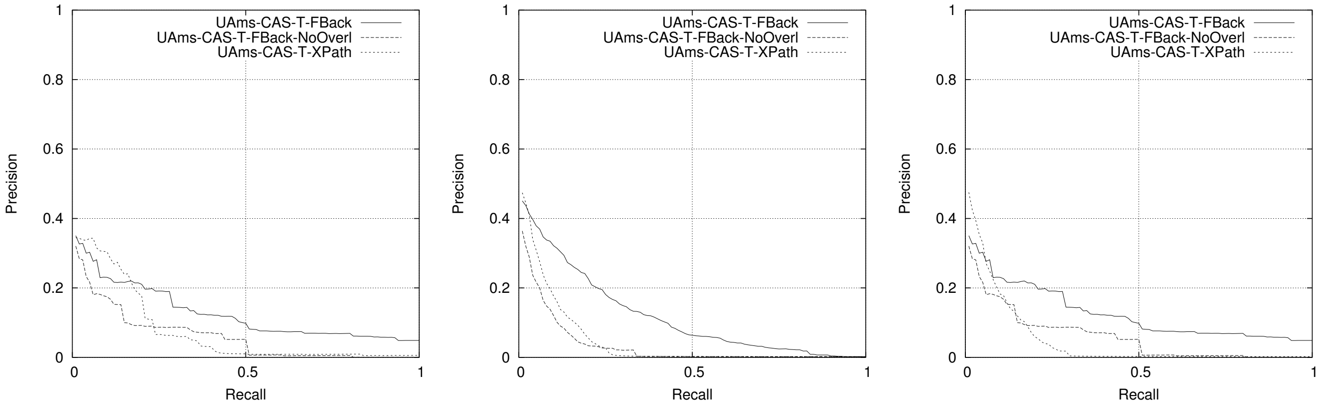

Figure 3: Precision-recall curves for our VCAS runs. (Left): strict measure. (Center): generalized measure. (Right): specificity oriented (so) measure.

4.2 Vague Content-And-Structure task

Table 4 shows the results for our VCAS runs, using all the different metrics. We see that the CO-style run clearly outperforms the XPath-style run with respect to all metrics. Again, the run without overlap scores the least of the three.

Figure 3 shows precision-recall curves for our VCAS runs, again for three measures: strict, generalized measure, and specificity oriented (so). The XPath-style run, tailored for a strict interpretation of content-and-structure topics, seems to function as a precision device. The run outperforms the CO-style run at lower recall levels. The low scores on higher recall level can immediately be explained by the fact that the target element is respected in the XPath-style run, but not in the relevance judgments.

	Run (CAS-T-. . .)		
Measure	. . . FBack	. . . FBack-NoOverl	. . . XPath
aggregate	*0.1065*	0.0397	0.0619
strict	*0.1260*	0.0582	0.0735
generalized	*0.1167*	0.0330	0.0451
so	*0.0912*	0.0282	0.0472
s3_e321	*0.0770*	0.0318	0.0537
s3_e32	*0.0817*	0.0365	0.0781
e3_s321	*0.1508*	0.0495	0.0581
e3_s32	*0.1020*	0.0404	0.0774

Table 4: Average scores for our VCAS runs, with the best scoring run in italics.

5. DISCUSSION AND CONCLUSIONS

In this paper, we documented our experiments at the INEX 2004 ad hoc retrieval track. We addressed three main research questions. First, we investigated the effectiveness of element-based query expansion, and found that it improved retrieval effectiveness on all but the exhaustiveness-oriented measures. We will discuss this case below. Second, we investigated the impact of (non-)overlap on the runs, and found that returning overlapping results results in superior scores on all measures. Our non-overlapping runs were, indeed, completely non-overlapping. Perhaps this is an unrealistically strong requirement, for it proves difficult to predict the choices of the assessors, and many relevant elements will be removed from the ranking. On a more positive note, the XPath-style run for SCAS had only 19% overlap, and got the best score at low recall levels. Third, our results for the VCAS task showed clear superiority of content-oriented-based approaches over a strict interpretation of the content-and-structure topics. From the vantage point of a retrieval system, our experiments highlighted the great

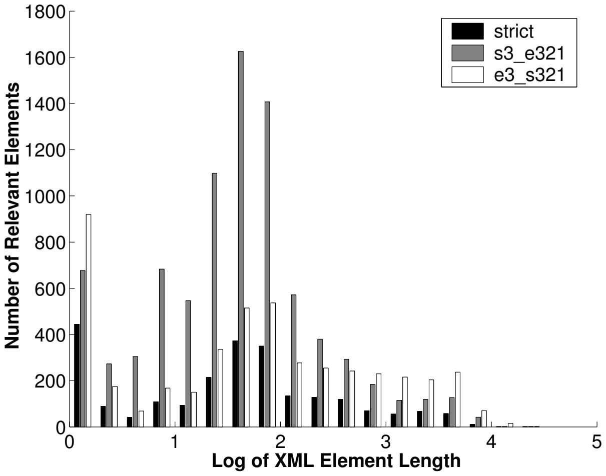

Figure 4: Length of relevant elements for strict, specificity-oriented, and exhaustiveness oriented measures in INEX 2004.

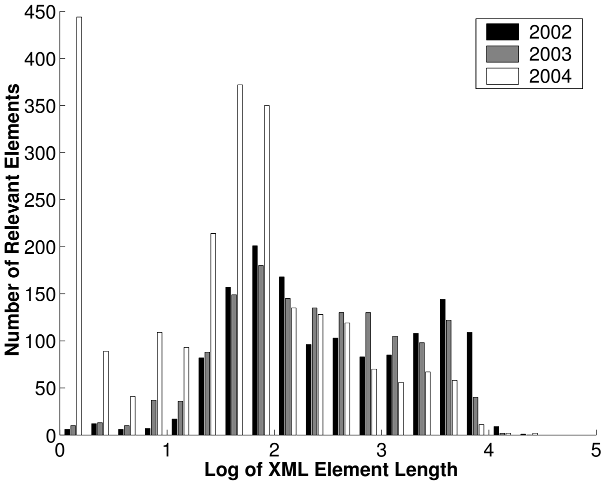

Figure 5: Length of relevant elements in INEX 2002–2004, measured using the strict measure.

similarity between the CO and VCAS tasks. The most notable difference, perhaps, is the fact that the XPath-style run can function as a precision device.

Previously, we have shown that a more radical length bias is essential to achieve good results [4]. Those experiments were performed using both the title and description fields of the topics. In the language modeling framework, as shown in Section 2.3, the final score of an element is the product of the prior probability of an element and the likelihood of the query given an element. However, the length of a query does have an effect on the number calculated for the query-likelihood. As a result, the normal length bias has a bigger impact on the shorter queries. Initial pre-submission experiments for the title-only topics showed the normal length-prior settings in Equation 4 in Section 2.3 to be sufficient. We did use blind feedback to expand queries with up to 5 terms. This will result again in longer queries, and perhaps may suggest that these are similar to the longer TD-topics. This is true in part, but there is an important difference between the TD-topics and (expanded) T-topics: all keywords from the title are content-bearing words specific for the query, as are supposedly the expanded terms. This may also be a factor that lessens the need for the extreme length-priors shown to be crucial for TD-topics [4].

We now return to the finding that query expansion does not help on the exhaustiveness-oriented measures, i.e., e3_s32 and e3_s321. One would expect that strict expansion of the query with useful terms (witnessing the other measures), leads to improvement of recall, and therefore would help exhaustiveness rather than specificity. In contrast, we see improvements on all measures *but* the exhaustiveness-oriented ones. This is clearly counterintuitive. Our best explanation to date has to do with the changing recall base of the measures. In Figure 4 we plot the (log of) element length against the number of relevant elements, where relevancy is determined by one of three measures: strict, specificity, and exhaustiveness. As can be seen from the plot, the strict and specificity measure return different counts but a very similar distribution over length. The exhaustiveness measure, in contrast, has a preference for much larger elements. This is not unexpected: if we stress the exhaustiveness dimension, we would generally expect to find larger chunks of text containing more information. As a result, our normal length prior is clearly insufficient to satisfy the exhaustiveness measure. The normal length prior creates more bias for the shorter unexpanded queries. Thus, for runs with a larger length bias may still show improvements for the expanded queries.

Figure 5 presents the distribution of the length of relevant XML elements over the three years of INEX CO, where relevancy is measured using the strict measure. While one has to be careful in making performance and test set comparisons across years, the following observations seem legit.

First, over the three years there is an declining preference for the larger elements such as full articles. In the first edition of INEX, i.e., in 2002, assessors frequently judged the larger elements relevant. In 2003, there was less of a preference for large elements, and in 2004 trend seems to persist: an even smaller fraction of the longer elements were judged relevant. With exception of the (almost) empty elements, the distribution of elements is qualitatively not very different from earlier years. Second, there is an amazing number of very small elements, ranging from empty to just one or a few words, that is judged as relevant. This raises a number of questions regarding the INEX relevance assessment stage. We find it implausible that an element with no or just one or two words can completely satisfy the information need of the topic (i.e., be judged as highly exhaustive and highly specific).

6. ACKNOWLEDGMENTS

Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO), under project number 612.066.302. Maarten de Rijke was supported by grants from NWO, under project numbers 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.-000.207, 612.066.302, 264-70-050, and 017.001.190.

7. REFERENCES

[1] T. Grust. Accelerating XPath Location Steps. In *Proc. SIGMOD*, pages 109–120. ACM Press, 2002.

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

[3] ILPS. The ILPS extension of the Lucene search engine,

2004. http://ilps.science.uva.nl/Resources/.

[4] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2004)*, pages 80–87, 2004.

[5] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. The importance of morphological normalization for XML retrieval. In *Proceedings of the First Workshop of the INitiative for the Evaluation of XML retrieval (INEX)*, pages 41–48. ERCIM Publications, 2003.

[6] J. Kamps and M. De Rijke. The effectiveness of combining information retrieval strategies for European languages. In *Proceedings 19th Annual ACM Symposium on Applied Computing*, pages 1073–1077, 2004.

[7] Lucene. The Lucene search engine, 2004. http://jakarta.apache.org/lucene/.

[8] J. Ponte. Language models for relevance feedback. In W.B. Croft, editor, *Advances in Information Retrieval*, chapter 3, pages 73–96. Kluwer Academic Publishers, Boston, 2000.

[9] J. J. Rocchio, Jr. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ, 1971.

[10] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. An Element-Based Approch to XML Retrieval. In *INEX 2003 Workshop Proceedings*, pages 19–26, 2004.

[11] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Processing content-oriented XPath queries. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM 2004)*, pages 371–380. ACM Press, 2004.

[12] Snowball. The Snowball string processing language, 2004. http://snowball.tartarus.org/.