TRIX 2004 – Struggling with the Overlap

Jaana Kekäläinen[1], Marko Junkkari[2], Paavo Arvola[2], and Timo Aalto[1]

[1] University of Tampere, Department of Information Studies,
33014 University of Tampere, Finland
{jaana.kekäläinen, timo.aalto}@uta.fi
[2] University of Tampere, Department of Computer Sciences,
33014 University of Tampere, Finland
marko.junkkari@cs.uta.fi, paavo.arvola@uta.fi

Abstract. In this paper, we present a new XML retrieval system prototype employing structural indices and a *tf***idf* weighting modification. We test retrieval methods that a) emphasize the *tf* part in weighting and b) allow overlap in run results to different degrees. It seems that increasing the overlap percentage leads to a better performance. Emphasizing the *tf* part enables us to increase exhaustiveness of the returned results.

1 Introduction

In this report, we present an XML retrieval system prototype, TRIX (Tampere retrieval and indexing system for XML), employing structural indices and a *tf***idf* weighting modification based on BM25 [3], [10]. The system is aimed for full scale XML retrieval. Extensibility and generality for heterogeneous XML collections have been the main goals in designing TRIX. This prototype is able to manipulate content_only (CO) queries but not content_and_structure (CAS) queries. However, with the CO approach of TRIX we achieved tolerable ranking for VCAS runs in INEX 2004.

One idea of XML is to distinguish the content (or data) element structure from stylesheet descriptions. From the perspective of information retrieval, stylesheet descriptions are typically irrelevant. However, in the INEX collection these markups are not totally separated. Moreover, some elements are irrelevant for information retrieval. We preprocessed the INEX collection so that we removed the irrelevant parts from the collection. The main goal of the preprocessing of the INEX collection was to achieve a structure in which the content element has a natural interpretation. In the terminology of the present paper, the content element means an element that has own textual content. The ranking in TRIX is based on weighting the words (keys) with a *tf***idf* modification, in which the length normalization and *idf* are based on content elements instead of documents.

The overlap problem is an open question in XML information retrieval. On one hand, it would be ideal that the result list does not contain overlapping elements [7]. On the other hand, the metrics of INEX 2004 encourage for a large overlap among results. In this paper, we introduce how the ranking of runs depends on the degree of overlap. For this, we have three degrees of overlap:

N. Fuhr et al. (Eds.): INEX 2004, LNCS 3493, pp. 127 – 139, 2005.

1. No overlapping is allowed. This means that any element is discarded in the ranking list if its subelement (descendant) or superelement (ancestor) is ranked higher in the result list.
2. Partial overlapping is allowed. The partial overlapping means that the immediate subelements and superelement are not allowed in the result list relating to those elements which have a higher score.
3. Full overlapping is allowed.

In this report we present the performance of two slightly different scoring schemes and three different overlapping degrees for both CO and VCAS tasks. The report is organized as follows: TRIX is described in Section 2, the results are given in Section 3, and discussion and conclusions in Sections 4 and 5 respectively.

2 TRIX 2004

2.1 Indices

The manipulation of XML documents in TRIX is based on the structural indices [4]. In the XML context this way of indexing is known better as Dewey ordering [11]. To our knowledge the first proposal for manipulating hierarchical data structures using structural (or Dewey) indices is found in [9]. The idea of structural indices is that the topmost element is indexed by ⟨1⟩ and its immediate subelements by ⟨1,1⟩, ⟨1,2⟩, ⟨1,3⟩ etc. Further the immediate subelements of ⟨1,1⟩ are labeled by ⟨1,1,1⟩, ⟨1,1,2⟩ ⟨1,1,3⟩ etc. This kind of indexing enables analyzing any hierarchal data structure in a straightforward way. For example, the superelements of the element labeled by ⟨1,3,4,2⟩ are found from indices ⟨1,3,4⟩, ⟨1,3⟩ and ⟨1⟩. In turn, any subelement related to the index ⟨1,3⟩ is labeled by ⟨1,3,ξ⟩ where ξ is a non-empty subscripts of the index.

In TRIX we have utilized structural indices in various tasks. First, documents and elements are identified by them. Second, the structure of the inverted file for elements is based on structural indices. Third, algorithms for degrees of overlapping are based on them. A detailed introduction to Dewey ordering in designing and manipulating inverted index is given in [1].

2.2 Weighting Function and Relevance Scoring

The content element is crucial in our weighing function. In this study, the content element is an element that has own textual content but none of its ancestors possess own textual content. Content elements are index units. For example, if the paragraph level is the highest level in which text is represented then paragraphs are manipulated as content elements and their descendants are not indexed. Content elements are chosen automatically for each document in the indexing process.

In TRIX the weighting of keys is based on a modification of the BM25 weighting function [3], [10]. Related to a single key k in a CO query the weight associated with the element e is calculated as follows:

$$w(k,e) = \frac{kf_e}{kf_e + v \cdot ((1-b) + b \cdot l\_norm(k,e))} \cdot \frac{\log\left(\frac{N}{m}\right)}{\log N} \quad (1)$$

where

- kf_e is the number of times k occurs in element e,
- m is the number of content elements containing k in the collection,
- N is the total number of content elements in the collection,
- v and b are constants for tuning the weighting,
- $l\_norm(k,e)$ is a normalization function defined based on the ratio of the number (ef_c) of all descendant content elements of the element e, and the number (ef_k) of descendant content elements of e containing k. If the element e is a content element then $l\_norm(k,e)$ yields the value 1. Formally, the length normalization function is defined as follows:

$$l\_norm(k,e) = \begin{cases} 1, \text{if } e \text{ is a content element} \\ efc / \sqrt{efk}, \text{otherwise} \end{cases} \quad (2)$$

The weighting formula 1 yields weights scaled into the interval [0,...,1].

TRIX does not support proximity searching for phrases. Instead, we require that each key k_i ($i \in \{1,\ldots,n\}$) in a phrase $p =$"$k_1,\ldots,k_n$" must appear in the same content element. This is a very simple approximation for weighting of phrases but it works well when content elements are short – such as paragraphs and titles.

Related to the element e the weight of the phrase p is is calculated as follows:

$$w(p,e) = \frac{\min(p,e)}{\min(p,e) + v \cdot \left((1-b) + b \cdot lp\_norm(p,e)\right)} \cdot \frac{\log\left(\frac{N}{m_p}\right)}{\log N} \quad (3)$$

where

- $\min(p,e)$ gives the lowest frequency among the keys in p in the element e,
- m_p is the number of content elements containing all the keys in p.
- v, b, N and ef_c have the same interpretation as in formula 1.
- lp_norm where ef_p is the number of descendant content elements of e containing all the keys in p,

$$lp\_norm(p,e) = \begin{cases} 1, \text{if } e \text{ is a content element} \\ efc / \sqrt{efp}, \text{otherwise} \end{cases} \quad (4)$$

In CO queries, a query fragment or sub-query (denoted by sq below) is either a key or phrase with a possible +/- prefix. The '+' prefix in queries is used to emphasize the importance of a search key. In TRIX the weight of the key is increased by taking a square root of the weight:

$$w(+sq,e) = \sqrt{w(sq,e)} \quad (5)$$

The square root increases the weight related to the element *e* and the query fragment *sq* (either *k* or *p*) because the weight of a query fragment is scaled between 0 and 1.

The '-' prefix in queries denotes an unwanted key. In TRIX the weight of such a key is decreased by changing the weight to its negation. For any key or phrase *sq* the minus expression -*sq* is weighted by the negation of the original weight as follows:

$$w(-sq,e) = -w(sq,e) \tag{6}$$

In other words, unwanted query fragments are manipulated in the interval [-1,0].

For combination of query fragments (with a possible +/- subscript) two operation have been implemented: average and a fuzzy operation called Einstein's sum [8]. Using the average the weight $w(q,e)$ related to the CO query $q = sq_1 \ldots sq_n$ is formulated as follows:

$$w(q,e) = \frac{\sum_{i=1}^{n} w(sq_i,e)}{n} \tag{7}$$

The other implemented alternative, Einstein's sum (denoted by $\oplus$), means that two weights w_1 and w_2 are combined as follows:

$$w_1 \oplus w_2 = \frac{w_1 + w_2}{1 + w_1 \cdot w_2} \tag{8}$$

Unlike the average the operation $\oplus$ is associative, i.e. $w_1 \oplus w_2 \oplus w_3 = (w_1 \oplus w_2) \oplus w_3 = w_1 \oplus (w_2 \oplus w_3)$. Thus, the weight (denoted by *w'*) of a CO query $q = sq_1\ sq_2 \ldots sq_n$ can be calculated follows:

$$w'(q,e) = w(sq_1,e) \oplus w(sq_2,e) \oplus \cdots \oplus w(sq_n,e) \tag{9}$$

To illustrate this function we apply it to topic 166 (+"tree edit distance" +xml -image) for an element *e*:

w'(+"tree edit distance" +xml -image, *e*)

First, Equation 9 is applied as follows:

w(+"tree edit distance", *e*) $\oplus$ *w*(+xml, *e*) $\oplus$ *w*(-image, *e*)

Then, Equations 5 and 6 are used (sqrt means square root in Equation 5)

sqrt(*w*("tree edit distance", *e*)) $\oplus$ sqrt(*w*(xml, *e*)) $\oplus$ -*w*(image, *e*)

Now, *w*("tree edit distance", *e*) is calculated using Equation 3 and the others using Equation 1.

2.3 Implementation

The TRIX is implemented in C++ for Windows/XP but the implementation is aimed for UNIX/LINUX as well. In implementing the present TRIX prototype we have paid

attention for effective manipulation of XML data structures based on structural indices. However, the efficiency has not been the main goal of TRIX.

The TRIX prototype has two modes: online mode and batch mode. In the online mode the user can run CO queries in the default database (XML collection). The batch mode enables running a set of CO queries. In this mode queries are saved in a text file. Running the CO queries of INEX 2004 in the batch mode takes about 40 minutes in a sample PC (Intel Pentium 4, 2.4 GHz, 512MB of RAM). The weights are calculated at query time for every element. The size of the inverted index is 174 MB.

The command-based user interface of the TRIX prototype is tailored for testing various aspects of XML information retrieval. This means that a query can be run with various options. For example, the user can select:

- the method (average or Einstein's sum) used in combining the query term weights,
- the degree of overlap (no overlapping, partial overlapping or full overlapping), and
- the values of the constants.

For example, the command string

```
TRIX –e –o b=0.1 queries2004co.txt
```

means that Einstein's sum is used for combining weights (parameter -e), full overlapping is allowed (parameter -o) and the *b* is 0.1. Finally, queries2004co.txt denotes the file from which the query set, at hand, is found. Actually, there is no assumption of ordering for the parameters of a query. For example, the command string

```
TRIX –o queries2004co.txt b=0.1 –e
```

is equivalent with the previous query.

The online mode of TRIX is chosen by the command

```
TRIX
```

After this command the user may give his/her query, e.g.:

```
+"tree edit distance" +xml -image
```

3 Data and Results

We preprocessed the INEX collection so that from a retrieval point of view irrelevant parts were removed. As irrelevant content we considered elements consisting of non-natural language expressions, e.g. formulas, abbreviations, codes. We classified irrelevant parts into three classes. First, there are elements which possess relevant content but the tags are irrelevant. Tags which only denote styles, such as boldface or italic, inhere in this class. These tags were removed but the content of elements was maintained. Second, there are elements whose content is irrelevant but their tags are necessary in order to maintain the coherent structure of documents. For example we appraised the content of <sgmlmath> and <tmath> elements to inhere in this class. Third, there are elements having irrelevant content whose tags are not necessary in

structural sense. These elements, such as <doi> and <en>, were removed. The selection of the parts to be removed was done by researchers, the removal was automatic.

For INEX 2004 we submitted both CO and VCAS runs though our system actually supports only CO queries. In both cases, the title field was used in automatic query construction. Phrases marked in titles were interpreted as 'TRIX phrases' in queries, i.e. all the phrase components were required to appear in the same element. In addition, all the components were added as single keys to queries. For example, topic 166 is formulated into a query as follows:

```
+"tree edit distance" +xml -image tree edit distance
```

In VCAS queries the structural conditions were neglected and all keys were collected into a flat query. Word form normalization for the INEX collection and queries was Porter stemming, and a stoplist of 419 words was employed.

3.1 Tuning Constants

Setting the values of the constants v and b in the weighting function has an impact on the size of elements retrieved. For analyzing this impact, v was tested with values 1 and 2, and b was varied between 0 and 1. The value $v = 2$ gave better performance than $v = 1$, and the former is thus used as default now on. We ran the CO queries using average scoring, no-overlap and full overlap with different values of b. Then, the result lists were analyzed for the percentage of different element types at document cut-off values (DCV) 100 and 1500. Our categorization was rather coarse; percentages of articles, sections, abstracts, paragraphs and others ware calculated in the result lists. Category *section* contains sections and 'equivalent elements' (see [5]); category *paragraph* contains paragraphs and equivalent elements. Only DCV 100 is reported below because DCV 1500 gave very similar results.

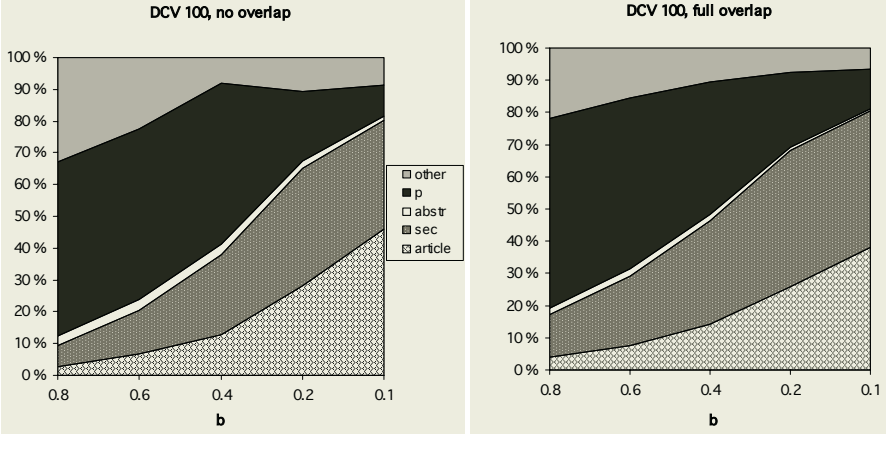

Fig. 1. Percentages of elements of different size in the result sets when b is varied

Figure 1 illustrates the change in the size of retrieved elements when *b* is varied between 0.8 and 0.1. The percentage of the smaller units increases from *b* value 0.1 to *b* value 0.8. In other words large *b* values promote small elements and the small b values promote large elements. This is due to strengthening the *tf* part in the weighting scheme by weakening length normalization. Although our categorization is rough and the category 'other' includes also large elements, the trend is visible. The change is apparent with and without overlap. In our official submissions *b* was 0.4. However, later tests revealed, that *b* value 0.1 gives better performance. Results with both values of *b*, 0.1 and 0.4, are reported in the following sections.

3.2 CO Runs

The evaluation measure used in INEX 2004 was precision at standard recall levels (see [2]) with different quantizations for relevance dimensions (see Relevance Assessment Guide elsewhere in these Proceedings). In the strict quantization only those elements that are highly exhaustive and highly specific are considered relevant, others non-relevant. In other quantization functions elements' degree of relevance is taken into account by crediting elements according to their level of specificity and exhaustiveness. (For details of metrics, see [12] or Evaluation metrics 2004 in these Proceedings.) The results are based on the topic set with relevance assessements for 34 topics. Our official submissions were:

1. CO_avrg: run using *w* weighting (average) with no overlapping when $b = 0.4$,
2. CO_Einstein: run using *w'* weighting (Einstein's sum) with no overlapping when $b = 0.4$,
3. CO_avrg_part_overlap: run using *w* weighting with partial overlapping when $b = 0.4$.

The results for 1 and 2 were so similar that we report the results based on average only. Further, in our official submissions two overlap degrees were tested: no overlapping and partial overlapping. Later on we added the full overlapping case.

Table 1. Mean average precision (MAP) and ranking of CO runs with average scoring

	b	MAP	Rank
No overlapping	0.4	0.0198	45
	0.1	0.0239	42
Partial overlapping	0.4	0.0443	31
	0.1	0.0487	25
Full overlapping	0.4	0.0831	11
	0.1	0.0957	10

Aggregate precision values, given in Table 1, are macro-averages over the different quantizations used in INEX 2004. Table 1 shows the effect of different overlaps and tuning of *b* to aggregate precision and rank. Decreasing *b* has a slight positive effect on the aggregate score and rank. When the different metrics are considered, it is

obvious that small *b* values enhance the dimension of exhaustiveness at specificity's expense. Figures 4 - 5 in Appendix show P-R-curves for CO runs with specificity- and exhaustiveness-oriented quantizations. In case of specificity-oriented quantization (Figures 4a-b and 5 a-b) average precision decreases as *b* decreases. Figures 4c-d and 5c-d in the appendix show an exhaustiveness-oriented quantization, and there average precision increases as *b* decreases. The mean average precision figures with all quantizations for our official submissions are given in Table 2.

Table 2. MAP figures for University of Tampere official CO submissions, $b = 0.4$

	MAP						
	strict	gen.	so	s3_e321	s3_e32	e3_s321	e3_s32
CO_avg	0.022	0.016	0.015	0.016	0.017	0.026	0.026
CO_Einstein	0.023	0.016	0.015	0.014	0.017	0.034	0.029
CO_avg_po	0.044	0.041	0.041	0.039	0.042	0.051	0.054

The effect of overlap is more substantial: allowing the full overlapping changes the aggregate rank from 45th to 11th when $b = 0.4$, or from 42nd to 10th when $b = 0.1$. Figure 2 illustrates the increase in the aggregate score when overlap percentage increases. (No overlap 0%; partial overlap 40%/44%; full overlap 63%/69%. Compare also Figures 4a and 5a, and 4b and 5b, etc. in Appendix). Whether the change in the result lists is desirable from the user's point of view is questionable because it means returning several overlapping elements from the same document in a row.

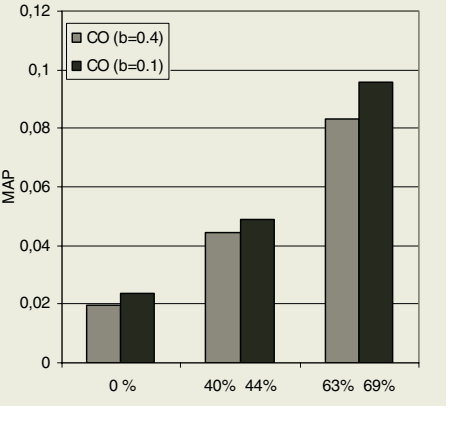

Fig. 2. Mean average precision and overlap percentage of CO runs with average scoring

3.3 VCAS Runs

The results of the VCAS runs are very similar to CO runs. Decreasing *b* value gives better exhaustivity-oriented results but impairs specificity. Increasing the overlap enhances effectiveness. Both these tactics have a positive effect on the aggregate score (see Table 3).

Table 3. Mean average precision and ranking of VCAS runs with average scoring

	b	MAP	Rank
No overlapping	0.4	0.269	30
	0.1	0.031	30
Partial overlapping	0.4	0.038	25
	0.1	0.042	22
Full overlapping	0.4	0.061	11
	0.1	0.075	7

Figure 3 shows the overlap percentages for different VCAS runs. Also here the benefits of allowing the overlap are evident though not as strong as with CO queries.

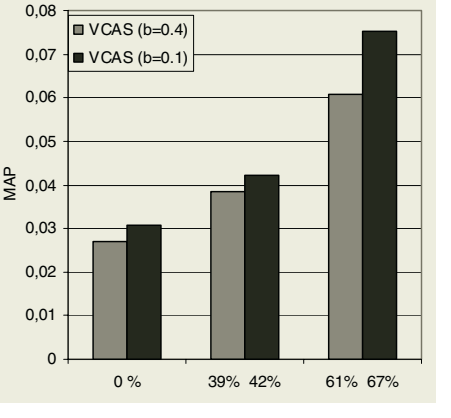

Fig. 3. Mean average precision and overlap percentage of CAS runs with average scoring

4 Discussion

In INEX 2004 University of Tampere group was struggling with the overlap. Our basic design principle was not to allow overlap in the result lists. Because of the structural indices of our system overlap is easy to eliminate. However, the reports of the previous INEX workshop led us to test effectiveness of partial overlap. Because of improved performance, we tested several runs with and without overlap, and allowing full overlap yielded the best performance. Nevertheless, the overlap percentage,

showing the percentage of elements that have either a superelement or a subelement ranked higher in the result list, is almost 70 in case of full overlap. This means that in the result list of 10 elements only 3 elements are 'totally new' for the user. It seems that the INEX metrics encourage returning overlapping elements though this might not be beneficial for the user. Our original idea of eliminating overlap was supported by an alternative measure, addressing the problem of overlapping relevant elements, proposed in [6]. The measure, XCG, ranked our runs without overlap higher than runs with overlap.

Our retrieval system, TRIX, employs a modification of *tf*idf* weighting. The number of content subelements is used in element length normalization. In the present mode, TRIX only supports CO queries but we aim at introducing a query language for content and structure queries. Because only titles of the topics – providing a very terse description of the information need – were allowed in query construction, and we did not expand the queries, a mediocre effectiveness was to be expected. Since TRIX does not support querying with structural conditions we submitted VCAS runs processed similarly as CO runs. Surprisingly our success with the VCAS task was not worse than with the CO task. However, if structural conditions are not considered when assessing the relevance, it is understandable that CO and VCAS tasks resemble each other.

Our further work with TRIX is aimed at introducing a query expansion or enhancing module. Incapability to deal with short content queries is a well-known disadvantage. Also, a CAS query language allowing also document restructuring is under construction.

5 Conclusion

In this paper we presented a *tf*idf* modification for XML retrieval. Instead of normalization based on the length of documents or elements we proposed a normalization function based on the number of content elements. We have shown how the well-known BM25 method, primarily intended to full-text information retrieval, can be applied to favor different sizes of XML elements. This sizing of result elements also has effects on the performance of queries. As our study indicates the performance strongly depends on the degree of overlap when such metrics as in INEX 2004 are used. The redundancy in returned elements might not serve the user. Therefore, if the user point of view is taken into account, nåw measures are needed.

Acknowledgments

This research was supported by the Academy of Finland under grant number 52894.

References

1. Guo, L., Shao, F., Botev, C. and Shanmugasundaram, J.: XRANK: Ranked Keyword Search over XML Documents. In: Proc. of ACM SIGMOD 2003, San Diego, CA (2003) 16-27

2. Gövert, N., and Kazai, G. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In: Proc. of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), (2002), pp. 1-17. Retrieved 27.1.2005 from http://qmir.dcs.qmul.ac.uk/inex/Workshop.html
3. Hawking, D., Thistlewaite, P., and Craswell, P. ANU/ACSys TREC-6 experiments. In: Proc. of TREC-6, (1998). Retrieved 10.3.2004 from http://trec.nist.gov/pubs/trec6/papers/anu.ps
4. Junkkari, M. PSE: An object-oriented representation for modeling and managing part-of relationships. Journal of Intelligent Information Systems, to appear.
5. Kazai, G. Lalmas, M, and Malik, S. INEX´03 guidelines for topic developmen. In: INEX 2003 Workshop Proc., (2003), pp. 192-199. Retrieved 21.1.2005 from http://inex.is.informatik.uni-duisburg.de:2003/internal/downloads/INEXTopicDevGuide.pdf
6. Kazai, G., Lalmas, M., and de Vries, A.P. Reliability tests for the XCG and INEX-2002 metrics. In INEX 2004 Workshop Pre-Proc. (2004), pp. 158-166. Retrieved 18.1.2005 from http://inex.is.informatik.uni-duisburg.de: 2004/pdf/INEX2004PreProceedings.pdf
7. Kazai, G., Lalmas, M., and de Vries, A.P. The overlap problem in content-oriented XML retrieval evaluation. In Proc. of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, (2004), pp.72-79.
8. Mattila, J.K. Sumean Logiikan Oppikirja: Johdatusta Sumean Matematiikkaan. Art House, Helsinki, (1998).
9. Niemi, T. A seven-tuple representation for hierarchical data structures. Information systems, 8, 3 (1983), 151-157.
10. Robertson S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., and Gatford, M. Okapi at TREC-3. In: NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3), (1994). Retrieved 21.11.2004 from http://trec.nist.gov/pubs/trec3/papers/city.ps.gz
11. Tatarinov, I., Viglas, S.D., Beyer, K., Shanmugasundaram, J., Shekita, E., and Zhang, C. Storing and querying ordered XML using a relational database system. In Proc. of the SIGMOD Conference, (2002), pp. 204-215.
12. de Vries, A.P, Kazai, G., and Lalmas, M. Evaluation metrics 2004. In INEX 2004 Workshop Pre-proc., (2004), pp. 249-250. Retrieved 18.1.2005 from http://inex.is.informatik.uni-duisburg.de:2004/pdf/INEX2004PreProceedings.pdf

Appendix

Precision-Recall Curves for CO Queries

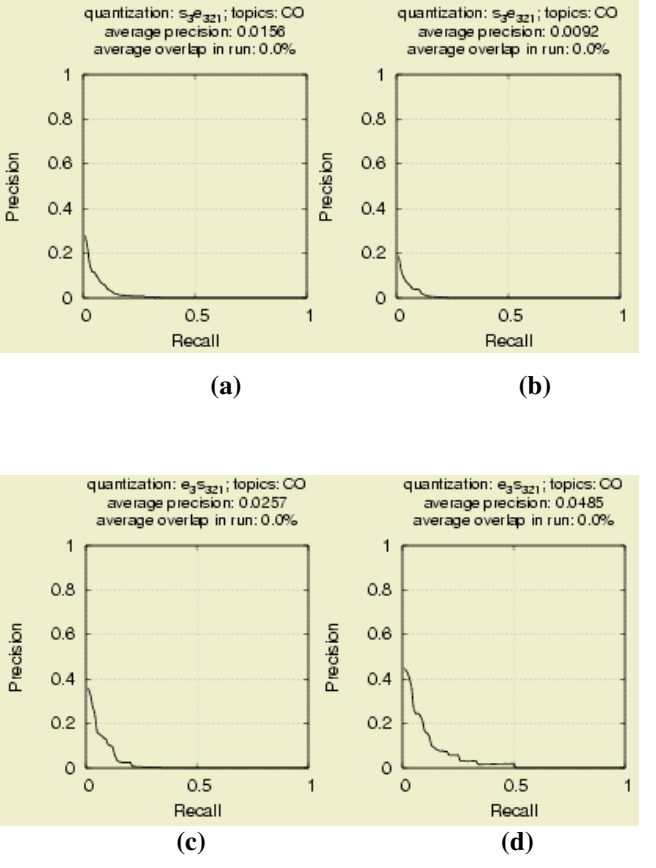

Fig. 4. CO without overlap. Quantization: s_3e_{321} (a) $b = 0.4$, rank 39/70; (b) $b = 0.1$, rank 46/70. Quantization e_3s_{321} (c) $b = 0.4$, rank 45/70 ; (d) $b = 0.1$, rank 39/70

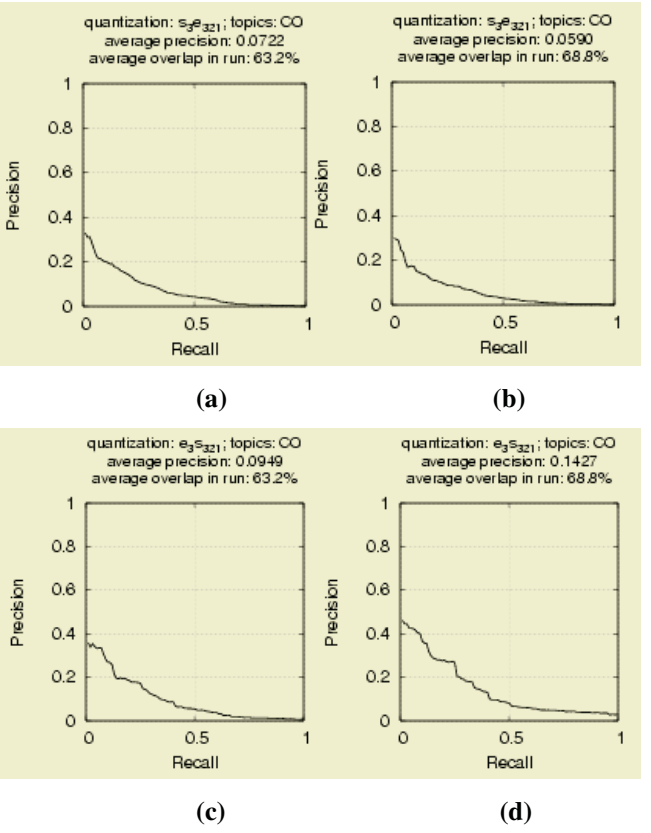

Fig. 5. CO with full overlap. Quantization: s_3e_{321} (a) $b = 0.4$, rank 8/70; (b) $b = 0.1$, rank 12/70. Quantization: e_3s_{321} (c) $b = 0.4$, rank 17/70; (d) $b = 0.1$, rank 11/70