Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

Field-Weighted XML Retrieval Based on BM25

Wei Lu
Center for Studies of Information Resources
School of Information Management
Wuhan University, China
sa713@soi.city.ac.uk

Stephen Robertson
Microsoft Research
Cambridge, U.K.
ser@microsoft.com

Andrew Macfarlane
Centre for Interactive Systems Research
Department of Information Science
City University London
andym@soi.city.ac.uk

Abstract

This is the first year for the Centre for Interactive Systems Research participation of INEX. Based on a newly developed XML indexing and retrieval system on Okapi, we extend Robertson's field-weighted BM25F for document retrieval to element level retrieval function BM25E. In this paper, we introduce this new function and our experimental method in detail, and then show how we tuned weights for our selected fields by using INEX 2004 topics and assessments. Based on the tuned models we submitted our runs for CO.Thorough, CO.FetchBrowse, the methods we propose show real promise. Existing problems and future work are also discussed.

1. Introduction

Being an important data exchange and information storage standard, XML is now widely used, especially for scientific data repositories, Digital Libraries and on the Web, which has brought about an explosion in the research of information retrieval for XML. Many sophisticated systems [1, 2, 3, 4, 5] and retrieval models for XML documents have been proposed [6, 7, 8, 9, 10].

XML documents often contain sub-fields (elements), eg. INEX collections from IEEE contain fields such as **title, abs, bdy, bm, st** etc. Practitioners have found it beneficial to exploit the document's internal structure to improve retrieval performance [11]. Researchers have looked at various techniques in order to investigate this problem. Wilkinson [12] and Ogilvie et al [13] have proposed and tested different ways to weight and combine the scores obtained on different fields of a document; Kraaij et al [14] propose a flexible algorithm based on language models but have not implemented it; and Myaeng et al [15] combine terms found in different document representations using Bayesian inference networks. Robertson et al [11] give a more detailed review of this area in

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

their paper.

In practice, many systems use a linear combination of the scores obtained from scoring every field due to the complexity of the ranking algorithms deployed. Robertson et al [11] discuss the dangers of linear combination in detail and propose an alternative solution, the linear combination of term frequencies based on BM25 (BM25F will be used in the rest of the paper instead of "field-weighted models based on BM25"), to extend standard ranking functions to multiple weighted fields. Their experiment based on two existing collection Reuters vol. I collection and the 2002 TREC Web-Track crawl of the .gov for document level retrieval shows that the method was beneficial. Some related work using Okapi, BM25 or field combination in INEX 2004 are documented in [16, 17, 18, 19, 20].

In this paper, we extend this method further to element level XML retrieval based on INEX 05 collections. In section 2, we discuss in detail the field-weighted models. Section 3 further illustrates the experiment of this method on INEX 05 and Evaluation results are reported in section 4. A conclusion and further work to be undertaken are described at the end.

2. BM25F model

In this section we describe BM25F model in detail. We first introduce the models for document level weighting in section 2.1. And then we further discuss the implementation of the model to XML element level retrieval.

2.1 BM25F for document level weighting

BM25F is the field-weighted version of BM25. It is derived from Robertson et al [11] for document level retrieval. For ad-hoc retrieval, and ignoring any repetition of terms in the query, BM25 can be simplified to [11]:

$$w_j(\overline{d},C) = \frac{(k_1+1)tf_j}{k_1((1-b)+b\frac{dl}{avdl})+tf_j}\log\frac{N-df_j+0.5}{df_j+0.5} \qquad (1)$$

where C denotes the document collection, tf_j is the term frequency of the *j*th term in $\overline{d}$, df_j is the document frequency of term *j*, *dl* is the document length, *avdl* is the average document length across the collection, and *k*1 and *b* are tuning parameters.

Then the document score is obtained by term weights of terms matching the query *q*:

$$W(\overline{d},q,c) = \sum_j w_j(\overline{d},C)\cdot q_j \qquad (2)$$

Being a linear weighted combination of term frequency of in these fields, function BM25F is shown as follows:

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

$$wf_j(\overline{d}, C) = \frac{(k'_1 + 1)tf'_j}{k'_1((1-b) + b\frac{dl'}{avdl'} + tf'_j} \log \frac{N - df_j + 0.5}{df_j - 0.5} \quad (3)$$

where tf'_j denotes the weighted term frequency of the *j*th term in $\overline{d}$, dl' is the weighted document length, $avdl'$ is the weighted average document length across the collection. k'_1 is the weighted free parameter.

Suppose we have *nF* fields *f = 1, . . . , nF*. In a given document *d*, term *t* has frequency $tf_{d,t,f}$ in field *f*. There are various ways of defining the length of fields or documents, but the simplest way is to use the number of indexed terms (tokens). This means that the length of the field in this document is

$$dl_f = \sum_{t \in v} tf_{d,t,f}$$

where V is the vocabulary, i.e. all indexed terms.

With no field weighting, the term frequency of t in the whole document is

$$tf_{d,t} = \sum_f tf_{d,t,f}$$

and the document length is

$$dl = \sum_f dl_f = \sum_f \sum_t tf_{d,t,f} = \sum_t tf_{d,t}$$

Average document length is

$$avdl = \frac{1}{N} \sum dl$$

With field weights W_f, these are modified as follows:

$$tf'_{d,t} = \sum_f w_f tf_{d,t,f}$$

$$dl' = \sum_f w_f dl_f = \sum_f \sum_t w_f tf_{d,t,f} = \sum_t tf'_{d,t}$$

$$avdl' = \frac{1}{N} \sum dl'$$

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

and

$$k'_1 = k_1 \frac{atf_{weighted}}{atf_{unweighted}} = k_1 \frac{avdl'}{avdl}$$

where *atf* is the average term frequency.

Function (3) is used for document weighting. However XML retrieval requires not only document level but also element level retrieval. This means an algorithm for element weighting is required. In section 2.2, we further discuss the field-weighted weighting function for element level retrieval (BM25E) derived from function (3).

2.2 Proposed model BM25E for element weighting

From function (3), we can see that linear combination of weighted field frequencies is used instead of original term frequency in specified document. We hypothesize that this method could also be applied to element retrieval. Our basic view is that an element is to be treated like a document, except that it may inherit information from other elements in the document. Thus each element has (in addition to its own text, which is treated as one field) extra fields consisting of text inherited from other elements. The details of our idea are as follows:

Suppose we have *nE* elements *e = 1, . . . , nE* in given collection C. Term *t* has frequency $tf_{d,t,e}$ in element *e*. *el* is the element length and *avel* is the average element length. Then we simply extend BM25 to element retrieval as follows:

$$w_j(e,\overline{d},C) = \frac{(k_1+1)tf_{e,j}}{k_1((1-b)+b\frac{el}{avel})+tf_{e,j}} \log\frac{N-df_j+0.5}{df_j+0.5} \qquad (4)$$

Accordingly, Function BM25E would be,

$$wf_j(e,\overline{d},C) = \frac{(k'_1+1)tf'_{e,j}}{k'_1((1-b)+b\frac{el'}{avel'})+tf'_{e,j}} \log\frac{N-df_j+0.5}{df_j+0.5} \qquad (5)$$

where $tf'_{e,j}$ denotes the weighted term frequency of *j*th term *t* in *e*, el' is the weighted element length, $avel'$ is the weighted average element length across the collection. k'_1 is the weighted free parameter. Similar to those parameters in section 2.1, given a field weights W_f to elements which contributes to a given element's Weight,

$$tf'_{f,t} = \sum_{f \in e} w_f tf_{f,t}$$

$$avel' = \frac{1}{M}\sum el'$$

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

$$el^{'} = \sum_{f \in e} w_f el = \sum_{f \in e} \sum_{t} w_f tf_{f,t} = \sum_{f,t} tf^{'}{}_{f,t}$$

and

$$k^{'}{}_1 = k_1 \frac{atf_{weighted}}{atf_{unweighted}} = k_1 \frac{avel^{'}}{avel}$$

where M is the total number of element in collection C.

(5) implies that given an element e in collection C, if it exists some fields(element) f contributing to the weight of the element, then a linear combination of field-weighted term frequency of field are applied based on BM25F. Theoretically, f could be any element in collection C. In fact , if all elements in a document d contribute to a given element in this document, then we come back to BM25F (3). And if all W_f equal 1, then we further come back to BM25 (1).

What we need to say is that this statement does not in any way define the implementation, but merely the principle of how elements are to be treated. Detail implementation of our experiment is further discussed in section 3.

3 Experiment of BM25E on INEX 2005

In this section, INEX collection and its structure will be introduced. We will then describe the assumptions we used for our experiments. Finally, our experiment environment and procedures are introduced.

3.1 Data sets

There are 2 data sets have been used for our experiment: INEX 1.4 and INEX 1.7. Both of these two collections are from IEEE Computer Society publications.

Inex 1.4: This data set is INEX collection for 2004 which contains 12107 articles of IEEE Computer Society publications from 1995 to 2002.

Inex 1.6: This data set is INEX collection for 2005 which contains 16819 articles of IEEE Computer Society publications from 1995 to 2004.

More details of these collections can be found in table 1.

Data sets	**INEX 1.4**	**INEX 1.6**
Size of Data(MB)	494	705
# of elements	8,239,873	11,411,135
# of attributes	2,204,688	4,669,699
# of Articles	12,107	16,819
Avg. Path Level	8	8

Table 1: figures of INEX collections

3.2 Data structures

As stated in section 1, being academic collections, most of the articles in it contain elements tags which represent article's title, abstract, body text, section, section title, paragraph, bibliography and

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

appendix etc. These tags in INEX collection are shown in Table 2:

Content Name	Tags
article title	atl
article abstract	abs
body text	bdy
section	sec, ss1, ss2, ss3
section title	st
paragraph	ilrj, ip1, ip2, ip3, ip4, ip5, item-none, p, p1, p2, p3
bibliography	bib
appendix	bm

Table 2: INEX important tags and its meaning

As it's discussed in [11], W_f needs to be tuned for each selected field which contributes to the document's weight in BM25F. The same method should also be used for BM25E. Although in theory, every context element would contribute to given element e, in practice, there are more than about ten-million elements in each INEX collections and it is very difficult to tune every element's W_f. The problem then lies in what elements should be chosen for optimisation.

Robertson et al [11] chose title as the tuned field. In this experiment, consider the data structures of INEX, we choose **atl, abs** and **st** as the tuned elements. We believe that title and abstract in some extent reflect the content of an article, and section title in some extent tells us the section and its sub-elements' content. We believe these elements could contribute to the weight of relevant elements. This issue will be discussed in more detail in section 3.3.

3.3 Some assumptions for BM25E on INEX 2005

Due to the costs of implementation and some other factors such as time limitations, we declare our assumptions for the experiments on the elements which should be inherited for other retrievable ones and the ways to compute $avel'$ and k'_1. They are as follows:

Assumption 1: elements in one document do not have effect on elements in other documents. Elements except **atl, abs** and **st** also don't have effect on other elements which are not their ancestors in the same document.

Assumption 2: Elements **atl** and **abs** contributes to the weight of elements **bdy**, **bm** and their child elements. Elements **st** contributes to the weight of the section it belongs to, and also of the section's child elements and article element. All **st** elements have the same W_f without considering the level they belong to.

Assumption 3: Due to the complexity to compute parameters $avel'$ and k'_1, we believe the values of the article level can be used instead of them for all elements.

Assumption 1 is simple and easy to understand. In Assumption 2, the question may lie in that what role element **st** plays in the relevant section's other parent elements except article element. And the question in Assumption 3 is that whether the simple replacement of the parameters would affect much of the result. These issues will be tackled in further research.

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

3.4 Experiment environment and procedures

This is the first year that the CISR has taken part in INEX. We largely conduct our work on Okapi in a Linux environment (using Red Hat 9). Being a traditional retrieval experiment system, Okapi undertake all the processing which was required by INEX experimentation. We have therefore done significant development work for both XML indexing and element level XML retrieval in order to participating in INEX.

Our experimental procedure is as follows: firstly, we tune W_f for selected elements **atl**, **abs** and **st**; secondly, we use Okapi's Basic Search System (BSS) to get a document result set; and finally we use a newly designed XML element weighting and displaying interface to get our final submissions required by INEX, among which, selected W_f parameters are used to get optimized runs. We should also state that only article, **abs**, **bdy**, **bm** and **section** and **paragraph** elements are considered as potential relevant elements for our final runs in our experiment. This may lose some relevant elements, but some small irrelevant elements are filtered at the same time. In the next section, we report our evaluation result for INEX 05.

4. Evaluation

In order to examine the new data structures and algorithms build for our INEX experiments, we used INEX 04 ad-hoc topics and assessment to tune W_f for **atl**, **abs** and **st** on document level by using the average precision score, (we did not evaluate using the INEX methodology at the element level). Our method shows that tuning W_f for these selected elements contributes to an improvement in retrieval performance on the INEX 04 collection. The tuning values for W_f are all integers. We first tuned W_f {**atl**, **abs st}** from {1, 1, 1} to {10, 10, 10} using increments of 1. Result shows that the values of{10, 3, 10} for W_f get the highest average precision score. The best tuning results were obtained when the tuning values for **atl** and **abs** are both 10 and tuning values for **st** are all between 3 to 6, we therefore investigated the tuning scope for **atl** and **abs**. We then tried to tune W_f {**atl**, **abs st}** from {1, 1, 1} to {50, 10, 50} in increments of 1. The results shows that a higher value for **atl** yielded better results, the best scope for **st** is from 12 to 25, while the best scope for **abs** was about the same for the first set of tuning experiments conducted. We conducted some further tuning experiments with a larger scope for **atl** and the ranges for **abs** and **st** set to between 1~10 and 10~30 respectively. In these experiments we tuned **atl** from 1 to 300 using increments of 10 and then used increments of 50 for **atl**, to a maximum value of 3000.. We believed that there was no point in investigating larger values. The best average precision score was recorded when the tuned value for **atl** is around 2400. Finally, we tuned **atl** from 2100 to 2700 in increments of 1 in order to obtain the best optimized results. Our experiment shows when using the values of 2356, 4 and 22 for W_f in elements **atl, abs** and **st** respectively we obtained the highest performance for article level retrieval on INEX 04 data. We are a little surprised that the best tuned value for **atl** is so high. The implication is that the selected elements, particularly **atl** and **st** contributed much to the document level XML retrieval in the INEX collection. See table 3 for some of our tuned result for INEX 04.

W_f {atl, abs, st}	Sum of (Avg precision for co all topics)
2356, 4, 22	0.143698
2416, 5, 22	0.143678

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

2668, 5, 25	0.143435
10, 4, 9	0.129819
1, 1, 1	0.124023

Table 3: tuned results for INEX 04 on document level

Due to the time and resource limitations, we only submited runs for CO.Thorough and CO.FetchBrowse. Based on these tuning experiments and considering the difference between document level retrieval and element level retrieval, and also being concerned that tuned W_f values for **atl** and **st** would be to high, we choose 3 sets of tuning constants of values for W_f {atl, abs, st}, namely {2356, 4, 22}, {1000, 4, 22} and {15, 4, 8} , for submitting CO.Thorough runs; and chose another 3 sets of tuning constants of values for W_f {atl, abs, st}, namely {1000, 4, 22}, {300, 4, 18} and {98, 4, 13}, for submitting CO.FetchBrowse runs.

Though we tuned W_f in document level, we are still pleased to see that our official runs for CO.Thorough rank at the top of the total 39 official runs, especially for "Metric: nxCG(25), Quantization: strict, Overlap=off", our 3 runs ranks 1st, 2nd and 22nd respectively; for "Metric: nxCG(50), Quantization: strict, Overlap=off", our 3 runs ranks 1st, 2nd and 10th respectively; and for "Metric: ep-gr, Quantization: strict,Overlap=off", our 3 runs ranks 1st, 5th and 13th respectively. See Fig. 1, Fig. 2 and Fig. 3 [21] for more information. We also tried to use metric nxCG to compare our 3 official runs for CO.Thorough with the non field-weighted runs whose W_f{ atl, abs, st } is {0, 0, 0}. Result shows that non field-weighted one ranks at the last while the former two runs rank at the top.

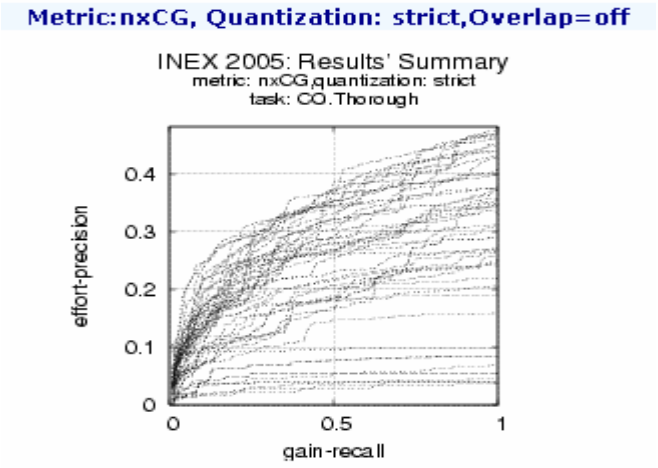

Fig. 1 Metric nxCG(25), Quantization: strict, Overlap=off

Published in: Advances in XML Information Retrieval and Evaluation (INEX 2005).
LNCS 3977, Springer 2006 (pp 161-171).

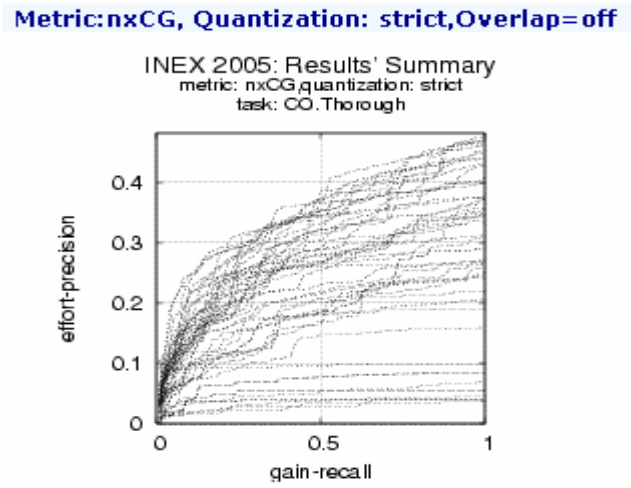

Fig. 2 Metric nxCG(50), Quantization: strict, Overlap=off

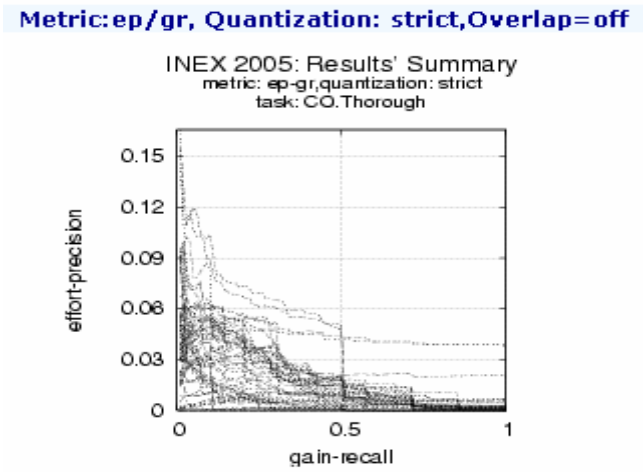

Fig. 3 Metric ep-gr, Quantization: strict,Overlap=off

The experiment shows that the first two sets of tuning constants, W_f {1000, 4, 22} and W_f {2356, 4, 22}, ranks better than the third groups W_f (15, 4, 8). The evidence is that **atl** and **st** does contribute to retrieval performance and it also implies that combining field-weighted term frequencies of selected elements is a beneficial method. Tuning constant set W_f {1000, 4, 22} rank first for Metric "nxCG(25 and 50), Quantization: strict, Overlap=off" also suggests that it may be better if W_f is tuned on element level. This behaviour may also be caused by the difference of the topics and data sets between INEX 2004 and INEX 2005 etc. It is worth doing a further set of tuning experiments on the INEX 2005 topics and data sets.

Results also show that our method performs better for models which consider only fully specific and highly exhaustive components than those models which considering varying levels of relevant components. The reason may be because the selection of elements we chose to submit for our experiments. We intend to investigate this issue further.

5 Conclusion

We extend document level field-weighted retrieval function BM25F to element level retrieval function BM25E. We have applied this method to INEX 2005 CO XML retrieval and results show

that our method is beneficial.

However there are still some limitations in our element level retrieval function. Firstly, values for $avel'$ and k_1' are used at the article level, not element level. The creation of a practical algorithm to generate values for tuning parameters at the element level is a challenging task. Secondly, parameter tuning is undertaken at document level by using average precision method, not on element level by using INEX official metrics. It should be noted that the element **st** has the same weight at different levels, and further experiments need to be undertaken to investigate this problem. Thirdly, we only submit runs for CO.Thorough and CO.FetchBrowse tasks, so more tasks need to be done to test our method. And also our system for XML element retrieval needs to be upgraded. We will investigate these problems in further research.

Acknowledgements

Thanks to Chinese Scholarship Council (CSC) for funding the visitor of the first author to City Univesity, London in order to conduct this research.

References

[1] A. Deutsch, M. Fernandez and D. Suciu. Storing semistructured data with STORED. In Proc. SIGMOD, 1999.

[2] J. Harding, Q. Li, B. Moon. XISS/R: XML Indexing and Storage System Using RDBMS. In Proceedings of the 29th VLDB Conference, 2003

[3] Software AG. Tamino XML database. http://www.softwareag.com/tamino/.

[4] XYZFind. XML Database. http://www.xyzfind.com.

[5] HYREX. http://ls6-www.cs.uni-dortmund.de/ir/projects/hyrex/.

[6] N. Fuhr and K. Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In Research and Development in Information Retrieval, 2001.

[7] J. E. Wolff, H. Florke, and A. B. Cremers. Searching and Browsing Collections of Structural Information. In Proc. IEEE Forum on Research and Technology Advances in Digital Libraries, 2000.

[8] T. Schlieder and H. Meuss. Querying and Ranking XML Documents. Special Topic Issue Journal American Society for Informations Systems on XML and Information Retrieval, 2002.

[9] T. Schlieder. Similarity Search in XML Data using Cost-Based Query Transformations. In Proc. 4th Intern. Workshop on the Web and Databases, 2001.

[10] A. Theobald and G. Weikum. The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking. In Proc. 8th Internation Conf. on Extending Database Technology, 2002.

[11] S. Robertson, H. Zaragoza, M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. CIKM'04, 2004.

[12] R. Wilkinson. Effective retrieval of structured documents. In Research and Development in Information Retrieval, 1994.

[13] P. Ogilvie and J. Callan. Combining document representations for known item search. In Proceedings
of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), 2003.

[14] W. Kraaij, T. Westerveld, D. Hiemstra. 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, 2002.
[15] S.Myaeng, D. Jang, M. Kim, Z. Zhoo. A flexible model for retrieval of SGML documents. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,1998.
[16] L. A. Clarke, L. Tilker. MultiText Experiments for INEX 2004. In INEX 2004 Proceedings, 2004.
[17] J. Vittaut, B. Piwowarski, Patrick Gallinari. An algebra for Structured Queries in Bayesian Networks. In INEX 2004 Workshop Proceedings, 2004.
[18] Jaana Kekäläinen, Marko Junkkari, Paavo Arvola. TRIX 2004 – struggling with the overlap. In INEX 2004 Workshop Proceedings, 2004.
[19] R. Larson. Cheshire II at INEX '04: Fusion and Feedback for the Adhoc and Heterogeneous Tracks. In INEX 2004 Workshop Proceedings, 2004.
[20] P. Ogilvie, J. Callan. Hierarchical Language Models for XML Component Retrieval. In INEX 2004 Workshop Proceedings, 2004.
[21] Evaluation results of CO.Thorough.
http://inex.is.informatik.uni-duisburg.de/2005/internal/results/CO.Thorough.html.