UJM at INEX 2009 Ad Hoc track

Mathias Géry, Christine Largeron

UJM at INEX 2009 Ad Hoc track

Mathias Géry, Christine Largeron

Université de Lyon, F-42023, Saint-Étienne, France
CNRS UMR 5516, Laboratoire Hubert Curien
Université de Saint-Étienne Jean Monnet, F-42023, France
{mathias.gery, christine.largeron}@univ-st-etienne.fr

Abstract. This paper[1] presents our participation to the INEX 2009 Ad-Hoc track. We have experimented the tuning of various parameters using a "training" collection (i.e. INEX 2008) quite different than the "testing" collection used for 2009 INEX Ad-Hoc track. Several parameters have been studied for article retrieval as well as for element retrieval, especially the two main BM25 weighting function parameters: b and k_1.

1 Introduction

The focused information retrieval (IR) aims at exploiting the documents structure in order to retrieve the relevant elements (parts of documents) matching the user information need. The structure can be used to emphasize some words or some parts of the document: the importance of a term depends on its formatting (*e.g.* bold font, italic, etc.), and also on its position in the document (*e.g.*, title versus text body). During our previous INEX participations, we have developed a probabilistic model that learns a weight for each XML tag, representing its capability to emphasize relevant text fragments [3] [2]. One interesting result was that article retrieval based on BM25 weighting gives good results against element retrieval, even when considering a precision oriented measure ($iP[0.01]$): 3 article retrieval runs appear in the top-10 of the focused task (2^{nd}, 4^{th} and 8^{th}, cf. [6]), and the 3 best $MAiP$ runs are 3 article retrieval runs! Thus a question comes: "Is BM25 suitable for element retrieval"? Indeed, we can imagine that, BM25 being developed for article retrieval, its adaptation to element retrieval is challenging. This problem has been addressed *e.g.* with BM25e [8].

Our objective during INEX 2009 was to answer to two questions, using the 2008 INEX collection as a training collection:

- is it possible to reuse the parameters tuned with INEX 2008 collection?
- is it still possible to obtain good results with article retrieval against element retrieval, regarding MAiP as well as $iP[0.01]$?

We present the experimental protocol in section 2, then our system overview in section 3, our tuning experiments using INEX 2008 in section 4, and finally our INEX 2009 results in section 5.

[1] This work has been partly funded by the Web Intelligence project (région Rhône-Alpes, cf. http://www.web-intelligence-rhone-alpes.org).

2 Experimental protocol

We have used the INEX Ad-Hoc 2008 collection as a training collection, and the INEX 2009 collection as a test collection. INEX 2008 collection contains 70 queries and 659,388 XML articles extracted from the English Wikipedia in early 2006 [1], while INEX 2009 collection contains 115 queries and 2,666,190 XML articles extracted from Wikipedia in 2008 [10]. We used the main INEX measures: $iP[x]$ the precision value at recall x, AiP the *interpolated average precision*, $MAiP$ the *mean AiP* and MAgP the *generalized mean average precision* [7]. The main INEX ranking is based on $iP[0.01]$ instead of the overall measure $MAiP$, allowing to emphasize the precision at low recall levels.

Given that every experiment is submitted to INEX in the form of a ranked list of 1,500 XML elements for each query, such measures favor, in terms of recall, the experiments for which whole articles are found (thereby providing a greater quantity of information for 1,500 documents). This is an issue, because focused answers may be penalized even if it is the very purpose of Focused IR to be able to return better granulated answers (i.e. relevant elements extracted from a whole article). Thus, we also calculated $R[1500]$, the recall rate for 1,500 documents, and $S[1500]$, the size (in Mb) of the 1,500 documents.

3 System overview

Our system is based on the BM25 weighting function [9], that processes articles a_j a well as elements e_j:

$$w_{ji} = \frac{tf_{ji} \times (k_1 + 1)}{k_1 \times ((1 - b) + (b * ndl)) + tf_{ji}} \times \log \frac{N - df_i + 0.5}{df_i + 0.5} \tag{1}$$

with:

- tf_{ji}: the frequency of t_i in article a_j (resp. element e_j).
- N: the number of articles (resp. elements) in the collection.
- df_i: the number of articles (resp. elements) containing the term t_i.
- ndl: the ratio between the length of articles a_j (resp. elements e_j) and the average article (resp. element) length (*i.e.* its number of terms occurrences).
- k_1 and b: the classical BM25 parameters.

Parameter k_1 allows to control the term frequency saturation. Parameter b allows to set the importance of ndl, *i.e.* the importance of document length normalization. This is particularly important in focused IR as the length variation for elements is greater than that of articles, as each article is fragmented into elements (the largest article contains about 35,000 words).

Our system also considers some other parameters, *e.g.*:

- *logical_tags*: list of XML tags which the system will consider either at indexing and querying step (the system will therefore not be able to return an element that does not belong to this list);

- $minimum\_size$: minimum size of documents (articles/elements) (# of terms);
- $level_{max}$: maximum depth of documents (depth of XML tree);
- df: df_i value for each term, computed on articles (INEX 2008: $max(df_i) = 659,388$); or on elements (INEX 2008: $max(df_i)$ between 1 and 52 millions);
- stop words: using a stop words list;
- parameters concerning queries handling: mandatory or banned query terms (+/- operators), etc.

4 Parameters tuning (INEX 2008)

4.1 System settings

All our runs have been obtained automatically, and using only the query terms (*i.e* the *title* field of INEX topics). We thus do not use the fields *description*, *narrative* nor *castitle*. Several parameters have been studied for article retrieval as well as for element retrieval. Some parameters were set after a few preliminary experiments, *e.g.*:

- $logical\_tags$ for article retrieval: *article*;
- $logical\_tags$ for element retrieval: *article, li, row, template, cadre, normallist, section, title, indentation1, numberlist, table, item, p, td, tr*;
- $minimum\_size_{terms}$: 10 terms. Some analysis on the INEX 2008 assessments (not presented here) have shown that it is not useful to consider elements smaller than 10 terms, because these small elements are either non-relevant or their father is 100% relevant, and in this case it is better to return the father. Note that [5] has shown, using former INEX 2002 collection, that an optimal value for this parameter is to be set around 40;
- $level_{max}$: 1 for article retrieval, 23 for element retrieval;
- df: computed on articles (resp. elements) while indexing articles (resp. elements), instead of computing an overall df (*e.g.* at article level) used while indexing articles as well as elements. Note that [11] compute an overall df.
- stop words: 319 words from Glasgow Information Retrieval Group[2],

Two important parameters were studied more thoroughly: b and k_1, using a 2D grid: b varying from 0.1 to 1, with 0.1 steps, and k_1 varying from 0.2 to 3.8 with 0.2 graduations), thus a total of 380 runs (article and element retrieval).

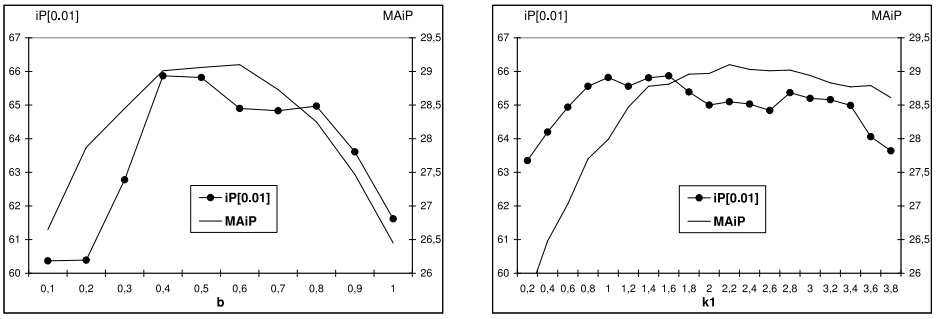
4.2 INEX 2008 tuning results

The results presented in this section were computed after INEX 2008 using the official evaluation program *inex-eval* (version 1.0).

Figure 1 presents the behavior of article retrieval, showing the MAiP and $iP[0.01]$ changes according to b (resp. k_1). For a given b (resp. k_1), the $iP[0.01]$ and $MAiP$ measures drawn are obtained using the optimal k_1 (resp. b).

[2] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

Fig. 1. Article retrieval in function of b and k_1



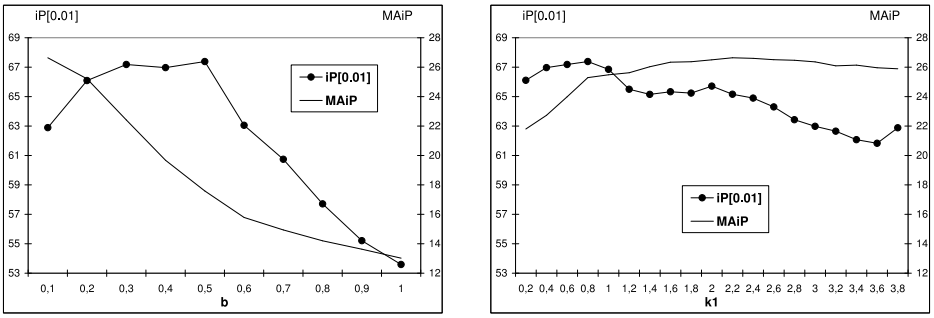
The best (b, k_1) values for article retrieval are slightly higher for MAiP ($(b, k_1) = (0.6, 2.2)$) than for iP[0.01] ($(b, k_1) = (0.4, 1.6)$). These values are not far from the classical values in the literature (*e.g.* (0.7, 1.2)). The results obtained with these optimal parameters are presented in table 1.

Table 1. Optimal b and k_1 parameters for article retrieval (iP[0.01] and $MAiP$)

Run	Granularity	b	k_1	Optimized results	#doc	#art	R[1500]	S[1500]
R1	Articles	0.4	1.6	iP[0.01] = 0.6587	1,457	1,457	**0.8422**	8.22
R2	Articles	0.6	2.2	$MAiP$ = **0.2910**	1,457	1,457	0.8216	6.15

Figure 2 presents the behavior of the BM25 model in focused IR.

Fig. 2. Focused IR in function of b and k_1



The best (b, k_1) values are different for MAiP ($(b, k_1) = (0.1, 2.2)$) than for iP[0.01] ($(b, k_1) = (0.5, 0.8)$). The best MAiP is reached with the minimum value $b = 0.1$. The length normalization of BM25 (through b) seems to be counter-productive when optimizing recall in focused IR. But on the other hand, it is still useful in order to optimize precision (best value: $b = 0.5$). The tf saturation (through k_1) seems to be less important for focused IR: both iP[0.01] and $MAiP$ slightly fluctuate with k_1. The results obtained with these optimal parameters are presented in table 2.

Table 2. Optimal b and k_1 parameters for focused IR ($iP[0.01]$ and $MAiP$)

Run	Granularity	b	k_1	Optimized results	#doc	#art	R[1500]	S[1500]
R3	Elements	0.5	0.8	$iP[0.01] = \mathbf{0.6738}$	1,463	1,257	0.4134	**1.65**
R4	Elements	0.1	2.2	$MAiP = 0.2664$	1,459	1,408	0.7476	5.24

5 INEX 2009 results

We present in this section the official results obtained during INEX 2009. We submitted 17 runs: 5 runs to the Focused task and 4 runs to the Best In Context, the Relevant In Context and the Thorough tasks. One run per task is based on the BM25 reference run (article-level ranking) given by the INEX organizers in order to facilitate cross-system comparisons [4].

5.1 System settings

All our runs have been obtained automatically, and using only the *title* field of INEX topics. Most of the settings given in section 4.1 have been reused for our INEX 2009 runs, except:

- $logical\_tags$ for element retrieval: article, list, p, reflist, sec, ss1, ss2, ss3, ss4, table, template (manually chosen);
- b and k_1: 0.6 and 2.2 for article retrieval (in order to maximize $MAiP$);
- b and k_1: 0.5 and 0.8 for element retrieval (in order to maximize $iP[0.01]$);
- $level_{max}$: 1 for article retrieval; 100 for element retrieval;
- df: computed on articles while indexing articles ($max(df_i) = 2,666,190$) and computed on elements while indexing elements ($max(df_i) = 444,540,453$).

5.2 Results: Focused task

Table 3 presents the official results of our runs, compared to UWFERBM25F2 (Waterloo University) which was the winning run of the Focused task.

Table 3. Official "Focused" task results (57 runs)

Run	Granularity	Reference run	b	k_1	$iP[0.01]$	Rank
UWFERBM25F2	Element	-	-	-	**0.6333**	1
UJM_15525	Article	-	0.6	2.2	0.6060	6
UJM_15479	Article	-	0.6	2.2	0.6054	7
UJM_15518	Element	INEX organizers	0.5	0.8	0.5136	36
UJM_15484	Element	-	0.5	0.8	0.4296	45

Our system gives very interesting results compared to the best INEX systems. Article retrieval, i.e. the BM25 model applied on full articles, achieves the best results in terms of precision: $iP[0.01] = 0.6060$ by UJM_15525 (differences between UJM_15525 and UJM_15479 settings are not significant). The article

retrieval runs outperform our focused IR run: $iP[0.01] = 0.4296$ (UJM_15484), despite the fact that BM25 parameter `ndl` is designed to take into account different documents lengths and thus documents granularities. This confirms the results obtained during INEX 2008. Note that our focused IR is improved when an article-level run (the reference run) is used as a pre-filter: $iP[0.01] = 0.5136$ by UJM_15518.

5.3 Relevant In Context (RIC), Best In Context (BIC), Thorough

Our BIC, RIC and Thorough runs have not been computed specifically. In order to respect the order and coverage rules of the RIC, BIC and Thorough tasks, our "focused" runs were reranked and filtered, using the same parameters settings than for the run UJM_15518. These results are presented in tables 4, 5 and 6.

Table 4. Official "Best In Context" task results (37 runs)

Run	Granularity	Reference run	b	k_1	$MAgP$	Rank
BM25bepBIC	Element	-	-	-	**0.1711**	1
UJM_15490	Element	UJM_15479	0.5	0.8	0.0917	28
UJM_15506	Element	UJM_15479	0.5	0.8	0.0904	30
UJM_15508	Element	INEX organizers	0.5	0.8	0.0795	34

Table 5. Official "Relevant In Context" results (33 runs)

Run	Granularity	Reference run	b	k_1	$MAgP$	Rank
BM25RangeRIC	Element	-	-	-	**0.1885**	1
UJM_15502	Element	UJM_15479	0.5	0.8	0.1075	21
UJM_15503	Element	INEX organizers	0.5	0.8	0.1020	26
UJM_15488	Element	UJM_15479	0.5	0.8	0.0985	27

Table 6. Official "Thorough" task results (30 runs)

Run	Granularity	Reference run	b	k_1	$MAiP$	Rank
LIG-2009-thorough-3T	Element	-	-	-	**0.2855**	1
UJM_15494	Element	INEX organizers	0.5	0.8	0.2435	9
UJM_15500	Element	UJM_15479	0.5	0.8	0.2362	12
UJM_15486	Element	-	0.5	0.8	0.1994	17

Runs UJM_15488 and UJM_15490 have been **filtered** with our best article run (UJM_15479), while UJM_15500, UJM_15502 and UJM_15506 have been **filtered and re-ranked** with the same article run (UJM_15479).

6 Conclusion

Our run UJM_15525 is ranked sixth of the competition according to the $iP[0.01]$ ranking. That means that a basic BM25 article retrieval run (article retrieval) gives better "precision" results ($iP[0.01]$) than BM25 element retrieval (focused IR), and should also give better "recall" results ($MAiP$).

These results confirm that article retrieval gives very good results against focus retrieval (as in INEX 2008 [3] [2]), even considering precision (that was not the case in 2008). However, we don't know if it comes from BM25, which is perhaps not suitable for elements indexing, or if it comes from a non optimal parameters settings. It is perhaps not so easy to reuse settings of parameters tuned on a different collection. We have to experiment more deeply on 2009 collection, using the same 2D grid for b and k_1, but also varying other parameters in order to better understand these results.

References

1. L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.
2. M. Géry, C. Largeron, and F. Thollard. Integrating structure in the probabilistic model for information retrieval. In *Web Intelligence*, pages 763–769, Sydney, Australia, December 2008.
3. M. Géry, C. Largeron, and F. Thollard. UJM at INEX 2008: Pre-impacting of tags weights. In *7th Workshop of the Initiative for the Evaluation of XML Retrieval*, volume 5631 of *LNCS*, pages 46–53, Dagstuhl Castle, Germany, December 2009.
4. S. Geva, J. Kamps, M. Lethonen, R. Schenkel, J. A. Thom, and A. Trotman. Overview of the INEX 2009 Ad Hoc track. In *INEX 2009 Workshop Preproceedings*, pages 16–50, Australia, December 2009.
5. J. Kamps, M. de Rijke, and B. Sigurbjörnsson. The importance of length normalization for XML retrieval. *Inf. Retr.*, 8(4):631–654, 2005.
6. J. Kamps, S. Geva, A. Trotman, A. Woodley, and M. Koolen. Overview of the INEX 2008 Ad Hoc track. In *7th Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 1–28, Dagstuhl Castle, Germany, December 2009. Springer-Verlag.
7. J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 evaluation measures. In *Focused access to XML documents, Workshop of the Initiative for the Evaluation of XML Retrieval*, 2007.
8. W. Lu, S. Robertson, and A. MacFarlane. Field-weighted XML retrieval based on BM25. In *Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 161–171, 2005.
9. S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *JASIST*, 27(3):129–146, 1976.
10. R. Schenkel, F. Suchanek, and G. Kasneci. Yawn: A semantically annotated wikipedia xml corpus. In *GI-Fachtagung fur Datenbanksysteme in Business, Technologie und Web*, volume 103 of *LNI*, pages 277–291. GI, 2007.
11. M. Taylor, H. Zaragoza, N. Craswell, S. Robertson, and C. Burges. Optimisation methods for ranking functions with multiple parameters. In *15th ACM conference on Information and knowledge management (CIKM '06)*, pages 585–593, New York, NY, USA, 2006.