UJM at INEX 2008: pre impacting of tags weights

Mathias Géry, Christine Largeron, Franck Thollard

UJM at INEX 2008: pre-impacting of tags weights

Mathias Géry, Christine Largeron and Franck Thollard

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, franck.thollard}@univ-st-etienne.fr

Abstract. This paper[1] addresses the impact of structure on terms weighting function in the context of focused Information Retrieval (IR). Our model considers a certain kind of structural information: tags that represent logical structure (title, section, paragraph, etc.) and tags related to formatting (bold, italic, center, etc.). We take into account the tags influence by estimating the probability that a tag distinguishes relevant terms. This weight is integrated in the terms weighting function. Experiments on a large collection during INEX 2008 IR competition showed improvements for focused retrieval.

1 Introduction

The focused information retrieval (IR) aims at exploiting the documents structure (e.g. HTML or XML markup) in order to retrieve the relevant elements (parts of documents) for a user information need. The structure can be used to emphasize some particular 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 terms versus text body).

Different approaches have been proposed to integrate the structure at the step of querying or at the step of indexing. Following [2], we propose to integrate the structure in the weighting function: the weights of terms are based not only on the terms frequencies in the documents and in the collection, but also on the terms position in the documents. This position can be defined by XML tags. This approach raises two questions: how to choose the structural weights? How to integrate them in the classical models?

Some works propose to choose empirically the tags and their weights [5] or to learn them automatically using genetic algorithms [8]. These approaches use generally less than five tags. We propose to learn automatically the tags weights, without limit on the number of tags.

Concerning the integration of the structure weights, Robertson et al. suggests to preserve the non linearity of the BM25 weighting function by pre-impacting

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

structure on the terms frequencies instead of impacting it directly on the global terms weights [6]. We propose to apply this approach in the context of focused XML IR.

The main contribution of this paper is a formal framework integrating structure, introduced in the next section. We present in section 3 our experiments and in section 4 our results in the INEX 2008 competition.

2 A structured document model

We consider in this paper the problem of extending the classical probabilistic model [7] that aims at estimating the relevance of a document for a given query through two probabilities: the probability of finding a relevant information and the probability of finding a non relevant information.

Our model takes into account the structure at two levels. Firstly, the logical structure (e.g. tags section, paragraph, table, etc.) is used in order to select the XML elements that are handled at the indexing step. These elements are the only ones that can be indexed, ranked and returned to the user. Secondly, the formatting structure (e.g. bold font, italic, etc.) and the logical structure are integrated into the terms weighting function. For a given tag we can estimate if it emphasizes terms in relevant documents or term in non relevant part of documents. A learning step computes a weight for each tag, based on the probability, to distinguish relevant terms and non relevant ones. At querying step, the relevance of an element is estimated based on the weights of the terms it contains, combined with the weights of the tags labeling those terms.

2.1 Term based score of XML elements

The relevance of an element e_j for a query Q is function of the weights of the query terms t_i that appear in the element. We use the weighting function BM25 [7]:

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

With tf_{ji}: frequency of t_i in e_j; N: number of elements in the collection; df_i: number of elements containing the term t_i; ndl: ratio between the length of e_j and the average element length; k_1 and b: classical BM25 parameters.

2.2 Tag based score of XML elements

The relevance of an element e_j relatively to the tags is based on the weights, noted w'_{ik}, of each term t_i labeled by a tag b_k. We used a learning set LS in which the relevant elements for a given query are known. Given the set R (resp. NR) that contains the relevant (resp. non relevant) elements, a contingency table can be built:

	R	NR	$LS = R \cup NR$
$t_{ik} \in e_j$ $t_{ik} \notin e_j$	r_{ik} $R - r_{ik}$	$nr_{ik} = n_{ik} - r_{ik}$ $N - n_{ik} - R + r_{ik}$	n_{ik} $N - n_{ik}$
Total	R	$NR = N - R$	N

With R: number of relevant terms; NR: number of non relevant terms. r_{ik}: number of times term t_i labeled by b_k is relevant; $\sum_i r_{ik}$: number of relevant terms labeled by b_k; n_{ik}: number of times term t_i is labeled by b_k; $nr_{ik} = n_{ik} - r_{ik}$: number of times term t_i labeled by b_k is not relevant.

Then, w'_{ik} can be used to distinguish relevant terms from non relevant ones according to the tags that mark them. This is closely related to probabilistic IR model, but in our approach tags are considered instead of terms and terms instead of documents.

$$w'_{ik} = \frac{P(t_{ik}|R)(1 - P(t_{ik}|NR))}{P(t_{ik}|NR)(1 - P(t_{ik}|R))} = \frac{r_{ik} \ \times (NR - nr_{ik})}{nr_{ik} \times (R - r_{ik})} \tag{2}$$

Moreover, we hypothesize that the property for a tag to distinguish relevant terms does not depend on terms, *i.e.* the weight of a tag b_k should be the same for all terms. We finally estimate for each tag b_k a weight w'_k:

$$w'_k = \frac{\sum_{t_i \in T} w'_{ik}}{|T|} \tag{3}$$

2.3 Global score of XML elements

In order to compute a global score, we propose a linear combination f_{claw}[2] between the weight w_{ji} of a term t_i and the average of the weights w'_k of the tags b_k that mark the term[3]:

$$f_{claw}(e_j) = \sum_{t_{ik} \in e_j / t_i \in Q} w_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{4}$$

In previous experiments [3], f_{claw} slightly improved recall but the results were not convincing. Even if the estimation of the tag weights must be carefully addressed, it appears that the way such weights are integrated into the final score is essential. Following [6], we take advantage of the non linearity of BM25 by pre-impacting the tags weights at the term frequency level. More precisely, tf is replaced by ttf[4]in BM25:

$$ttf_{ji} = tf_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{5}$$

[2] CLAW: Combining Linearly Average tag-Weights.

[3] w_{ji}: the BM25 weight of term t_i in element e_j, cf. eq. 1.

[4] TTF: Tagged Term Frequency. $t_{ik} = 1$ means that t_i is labelled by b_k.

3 Experiments

We have experimented these models during the INEX 2008 IR competition in a classic IR way (granularity: full articles) as well as in a focused IR way (granularity: XML elements). The English Wikipedia XML corpus [1] contains 659,388 strongly structured articles, which are composed of 52 millions of XML elements (i.e. 79 elements on average; with an average depth of 6.72). The whole articles (textual content + XML structure) represent 4.5 Gb while the textual content only 1.6 Gb. The original Wiki syntax has been converted into XML, using both general tags of the logical structure (article, section, paragraph, title, list and item), formating tags (like bold, emphatic) and frequently occurring link-tags.

3.1 Experimental protocol

The corpus enriched by the INEX 2006 assessments on 114 queries has been used as a training set in order to estimate the tags weights w'_k. We have evaluated our approach using the 70 queries of INEX 2008.

Our evaluation is based on the main INEX measures ($iP[x]$ the precision value at recall x, AiP the *interpolated average precision*, and $MAiP$ the *interpolated mean average precision* [4]). Note that the main ranking of INEX competition is based on $iP[0.01]$ instead of the overall measure $MAiP$, in order to take into account the importance of precision at low recall levels.

Each run submitted to INEX is a ranked list containing at most 1 500 XML elements for each query. Some runs retrieve all the relevant elements among the first 1 500 XML returned elements, and some others retrieve only part of them. Note that a limit based on a number of documents (instead of *e.g.* a number of bytes) allows to return more information and therefore favors runs composed by full articles. We have calculated $R[1500]$ (the recall at 1 500 elements) and $S[1500]$ (the size of these 1 500 elements in Mbytes).

3.2 Tags weighting

We have manually selected 16 tags (*article*, *cadre*, *indentation1*, *item*, *li*, *normallist*, *numberlist*, *p*, *row*, *section*, *table*, *td*, *template*, *th*, *title*, *tr*) in order to define the XML elements to consider. These logical structure tags will be considered during the indexing step and therefore those will define the elements the system will be able to return.

Regarding the other tags (namely the formatting tags), we first selected the 61 tags that appear more than 300 times in the 659,388 documents. We then manually removed 6 tags: article, body (they mark the whole information), br, hr, s and value (considered not relevant).

The weights of the 55 remaining tags were computed according to equation w'_k in equation 3. Table 1 presents the top 6 tags and their weights, together with the weakest 6 ones and their weights. Their frequencies in the whole collection is also given.

Table 1. Weight w'_k of the 6 strongest and 6 weakest tags

Top strongest weights			Top weakest weights		
tag	weight	freq.	tag	weight	freq.
h4	12,32	307	emph4	0,06	940
ul	2,70	3'050	font	0,07	27'117
sub	2,38	54'922	big	0,08	3'213
indentation1	2,04	135'420	em	0,11	608
section	2,01	1'610'183	b	0,13	11'297
blockquote	1,98	4'830	tt	0,14	6'841

4 Results: focused task

Our aim was firstly to obtain a strong baseline, secondly to experiment focused retrieval (i.e. elements granularity) against classic retrieval (i.e. full articles granularity), and thirdly to experiment the impact of tags weights in the BM25 weighting function. Table 2 presents the 3 runs that we have submitted to INEX 2008 Ad-Hoc in focused task. The structure is not taken into account in R1, where the documents are returned to the user (articles granularity) as well as in R2 where the elements are returned (elements granularity), while in R3 the tags weights are integrated in BM25 in a focused retrieval (elements granularity - TTF)

Table 2. Our 3 runs submitted to INEX 2008 Ad-Hoc, focused task

Run (name)	Granularity	Tags weights
R1 (JMU_expe_136)	articles	-
R2 (JMU_expe_141)	elements	-
R3 (JMU_expe_142)	elements	TTF

4.1 Parameters

The parameters of the chosen weighting functions (namely `BM25`) were tuned in order to improve classic retrieval (articles granularity) and focused retrieval (elements granularity). Among the parameters studied to improve the baseline, we can mention the use of a stoplist, the optimization of BM25 parameters ($k_1 = 1.1$ and $b = 0.75$), etc. Regarding the queries, we set up a better "andish" mode and consider *or* and *and*, *etc* . . . . Some specific parameters (*e.g.* the minimum size of the returned elements) were also tuned for focused retrieval.

Our baseline and all other runs have been obtained automatically, and using only the query terms (*i.e* the *title* field of INEX topics). We thus do not use fields *description*, *narrative* nor *castitle*.

4.2 INEX ranking: $iP[0.01]$

Our system gives very interesting results compared to the best INEX systems. Our runs are compared on the figure 1 against *FOERStep*, the best run submitted to INEX 2008 according to $iP[0.01]$ ranking, on 61 runs yet evaluated in the focused task. This run outperforms our runs at very low recall levels. Our run $R1$ gives the best results at recall levels higher than 0.05. This is also shown by the $MAiP$ presented in table 3.

Fig. 1. Recall / Precision of 3 runs on 61 runs yet evaluated in the focused task

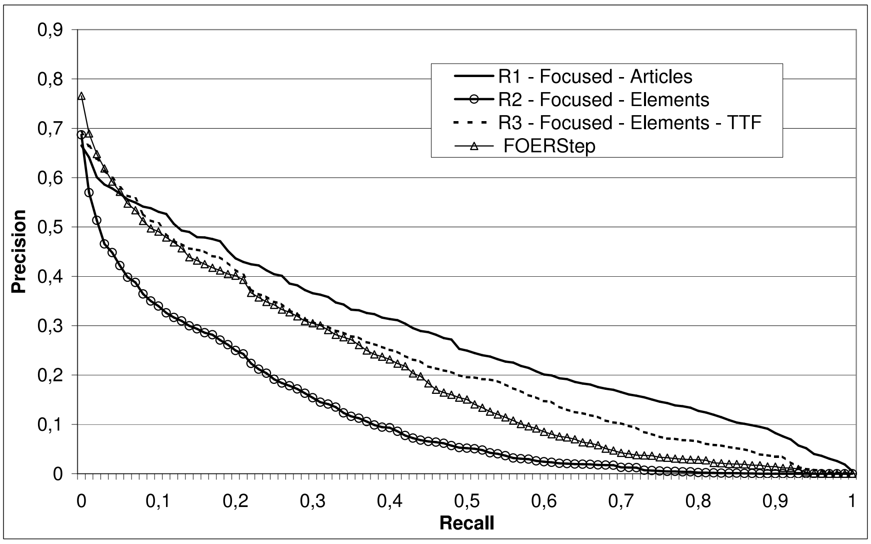

Table 3 presents the results of our 3 runs submitted to the track Ad-Hoc (focused task).

Table 3. Our 3 runs compared to 61 "focused" runs

Run (rank)	$iP[0.01]$	$MAiP$	$R[1500]$	$S[1500]$
FOERStep (winner)	**0.6897**	0.2076	0.4468	**78**
R1: articles (14)	0.6412	**0.2801**	**0.7871**	390
R2: elements (37)	0.5697	0.1208	0.2761	**51**
R3: elements+TTF (9)	**0.6640**	0.2347	0.6097	234

4.3 Articles versus elements

Our second aim was to compare classic retrieval of full articles versus focused retrieval of XML elements. We therefore indexed either the whole articles or the elements, and the parameters of the system were tuned also for focused retrieval.

It is interesting to notice that the BM25 model applied on full articles ($R1$) outperforms our focused retrieval results ($R2$) considering $MAiP$, despite the fact that BM25 parameter `ndl` is designed to take into account different documents lengths and thus documents granularities. Classic IR weighting functions, indexing and querying process, are undoubtedly not well adapted to focused retrieval. However, this is consistent with other results obtained during the INEX 2007 campaign where some top ranked systems only consider (and therefore return) full articles.

On the other hand, the focused run $R2$ returns a smallest quantity of information. Indeed, the total size of the 1 500 XML elements returned (for each query) is reduced to 51 Mb instead of 390 Mb for classic retrieval of full articles.

4.4 Pre-impacting of tags weights on terms weights

Finally, our third aim was to experiment the impact of tag weights in term weighting function in a focused retrieval scheme. In order to understand the pro and cons of our structured model, the weighting functions and the same parameters used for the baseline runs were also used with our structured model.

The figure 1 shows that our TTF strategy ($R3$) improves dramatically the focused retrieval at low recall levels (from 0.5697 to 0.6640 following $iP[0.01]$ ranking). However, it does not improve focused retrieval enough to reach better results than classic retrieval.

These results confirm also that, according to Robertson and *al.* [6], it is important to keep the non linearity of the BM25 weighting function by "pre-impacting" term position in the structure of document (in other terms, tags weights) on the terms frequencies (strategy TTF) instead of "post-impacting" it directly on the terms weights (strategy CLAW, cf. [3]).

5 Conclusion

We proposed in [3] a new way of integrating the XML structure in the classic probabilistic model. We consider both the logical structure and the formatting structure. The logical structure is used at indexing step to define elements that correspond to part of documents. These elements will be indexed and potentially returned to the user. The formatting structure is integrated in the document model itself. During a learning step using the INEX 2006 collection, a weight is computed for each formatting tag, based on the probability that this tag distinguishes relevant terms. During the querying step, the relevance of an element is evaluated using the weights of the terms it contains, but each term weight is modified by the weights of the tags that mark the term.

The baselines are rather strong as the score of the BM25 run on article (run $R1$) is ranked seven of the competition according to the $iAP[0.01]$ ranking.

Our strategy TTF gives better results than focused retrieval (*R2*) and classic retrieval (*R1*) at low recall levels ($iP[0.01]$). That shows the interest of focused IR (*R3* vs *R1*), and the interest of using structure (*R3* vs *R2*). Pre-impacting the structure on terms frequencies (TTF, *R3*) gives also better results than "post-impacting" it on final terms weights (CLAW, [3]). Actually, TTF changes significantly the performances of the methods when considering the $iP[0.01]$ or the $MAiP$ measure.

TTF (*R3*) gives also good recall results ($MAiP = 0.2347$; $R[1500] = 0.6097$). Focused IR eliminates more non relevant elements than relevant elements (*R3* vs *R1*): $R[1500]$ decreases by 16% while S decreases by 40%.

We have presented a document model integrating explicitly the structural information in the weighting function, and a learning process of tags weights. We reach the same conclusions than [6] about the interest of pre-impacting structure, with a very different collection, a more heterogeneous one that contains a much larger set of tags ($>$ 1 thousand).

In previous experiments, a basic average function, that considers all the tags equally (CLAW), gives better results than other combining functions (multiplication, only the closest tag, etc.). But, we think that a finest combining function (e.g. taking into account the distance between terms and tags) should improve the results.

References

1. Ludovic Denoyer and Patrick Gallinari. The wikipedia XML corpus. In *SIGIR forum*, volume 40, pages 64–69, 2006.
2. Michael Fuller, Eric Mackie, Ron Sacks-Davis, and Ross Wilkinson. Coherent answers for a large structured document collection. In *SIGIR*, pages 204–213, 1993.
3. Mathias Géry, Christine Largeron, and Franck Thollard. Integrating structure in the probabilistic model for information retrieval. In *Web Intelligence*, pages 763–769, 2008.
4. J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 evaluation measures. In *Focused access to XML documents, INEX*, 2007.
5. Joaquin Rapela. Automatically combining ranking heuristics for html documents. In *WIDM*, pages 61–67, 2001.
6. S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM*, pages 42–49, New York USA, 2004.
7. S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *JASIST*, 27(3):129–146, 1976.
8. Andrew Trotman. Choosing document structure weights. *IPM*, 41(2):243–264, 2005.