UJM at INEX 2007: Document Model Integrating XML Tags

Mathias Géry, Christine Largeron, and Franck Thollard

Jean Monnet University, Hubert Curien Lab, Saint-Étienne, France
{Mathias.Gery,Christine.Largeron,
Franck.Thollard}@univ-st-etienne.fr

Abstract. Different approaches have been used to represent textual documents, based on boolean model, vector space model or probabilistic models. In text mining as in information retrieval (IR), these models have shown good results about textual documents modeling. They nevertheless do not take into account documents structure. In many applications however, documents are inherently structured (e.g. XML documents).

In this article[1], we propose an extended probabilistic representation of documents in order to take into account a certain kind of structural information: logical tags that represent the different parts of the document and formatting tags used to emphasized text. Our approach includes a learning step that estimates the weight of each tag. This weight is related to the probability for a given tag to distinguish the relevant terms.

1 Introduction

In Information Retrieval as in text mining many approaches are used to model documents. As stated in [1], these approaches can be organized in three families: models based on boolean model (for example fuzzy or extended boolean model); models based on vector space model; probabilistic models. The latter holds Bayesian networks, inference networks or belief networks. All these models appear to be appropriate to represent textual documents. They were successfully applied in categorization task or in information retrieval task.

However they all present the drawback of not taking into account the structure of the documents. It appears nevertheless that most of the available information either on the Internet or in textual databases are strongly structured. This is for example the case for scientific articles in which a title, an abstract, keywords, introduction, conclusion and other sections do not have the same importance. This is also true for the documents available on the Internet as they are written in languages (e.g. HTML or XML) that explicitly describe either the logical structure of the document (section, paragraph,...) and the formatting structure (*e.g.* font, size, color, ...).

For all these documents, the information provided by structure can be useful to emphasize some part of the textual documents. Consequently a given word does not have the same importance depending on its position in the article (*e.g.* in the title or in the

[1] This work has been partly funded by the Web Intelligence project (région Rhône-Alpes).

N. Fuhr et al. (Eds.): INEX 2007, LNCS 4862, pp. 103–114, 2008.

body) or if it is emphasized (bold font, etc.). Indeed, if the author of a web page deliberately writes a given word in a particular font, it could be thought that a particular information can be associated with the term and therefore that the term should be considered differently.

For all these reasons, recent works in information retrieval as in text mining, takes into account the structure of documents. This leads, in particular, to content oriented XML information retrieval (IR) that aims at taking advantage of the structure provided by the XML tree. Taking into account the structure can be done either at the indexing step or at the querying one. In the former [2,9,7], a structured document is indexed using a tree of logical textual elements. The terms weight in a given element is propagated through the structural relation, *i.e.* from leafs to the root or from root to leafs. In the latter [5], SQL query language has been adapted to the structured context in order to allow queries like "I look for a paragraph dealing with running, included in an article that deals with the New-York marathon and in which a photo of a marathon-man is present". The INEX competition (INitiative for Evaluation of XML Retrieval[2]) provides, since 2002, large collections of structured documents. Systems are evaluated through their ability to find relevant part of documents associated with XML element rather than the whole documents.

In this article, we propose to extend the probabilistic model in order to take into account the document structure (the logical structure and the formatting structure). Our approach is made up of two steps: the first one is a learning step, in which a weight is computed for each tag. This weight is estimated, based on the probability that a given tag distinguishes relevant terms. In the second step, the above weights are used to better estimate the probability for a document to be relevant for a given query.

An overview of our model is presented in the next section. A more formal one follows in section 3. The results obtained on the INEX 2006 & 2007 collections are then presented in section 4.

2 Integrating Tags into Document Modeling

In Information Retrieval, the probabilistic model [6] 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.

These estimates are based on the probability for a given term in the document to appear in relevant (or in non relevant) documents. Given a training collection in which the documents relevance according to some query is available, one can estimate the probability for a given term to belong to a relevant document (respectively non relevant document), given its distribution in relevant documents (respectively non relevant documents).

This probabilistic model leads to good results in textual information retrieval. Our goal here is to extend this model by taking into account the documents structure. Different kinds of "structure" can be considered. As an example, Fourel defined physical structure, layout structure, linguistic structure, discursive structure and logical structure

[2] See `http://inex.is.informatik.uni-duisburg.de/2007/` for more details on the INEX competition.

[3]. In our model, we consider the structure defined through XML tags: logical structure (title, section, paragraph, etc.) and formatting structure (bold font, centered text, etc.).

Then, the structure is integrated in the probabilistic model at two levels:

1. The logical structure is used in order to select the XML elements (section, paragraph, table, etc.) that are considered at the indexing step. Given a query, these indexed elements are the only ones that can be ranked and returned to the user.
2. The formatting structure is then integrated into the probabilistic model, in order to improve terms weighting.

Integrating formatting tags needs a learning step in which a weight for each tag is computed. This weight is based on the probability, for a given tag, to distinguish relevant terms from non relevant ones. This is closely related to the classic probabilistic model, in which a weight for each term is estimated, based on the probability for the term to appear in relevant documents or in non relevant documents. But in our approach, tags are considered instead of terms and terms instead of documents. Thus the relevance is evaluated on documents parts (term by term) instead of whole documents, and the probability for a tag to distinguish relevant terms from non relevant ones is estimated. Accordingly, in the INEX collections, the relevance is defined on structural elements, i.e. XML elements and parts of them (i.e. sentences[3]).

During querying step, the probability for an element to be relevant is estimated based not only on the weights of the terms it contains, but also on the weights of the tags that labeled these terms.

A more formal presentation of our model is given in the next section.

3 A Probabilistic Model for the Representation of Structured Documents

3.1 Notations and Examples

Let $\mathcal{D}$ be a set of structured documents. We will consider here XML documents. Each logical element (article, section, paragraph, table, etc.) e_j of the XML tree will therefore be represented by a set of terms. We now present a running example in which three documents D_0, D_1 and D_2 are present:

D_0	D_1	D_2
<article>	<article>	<article>
$t_1t_2t_3$	<section>	<section>
<section>	t_2t_4	t_5
t_1t_4	t_2t_5	t_3t_4
t_2t_5	</section>	t_3t_5
</section>	t_2t_1	</section>
</article>	</article>	</article>

[3] In our model, we do not consider the relevance of sentences, but only the relevance of XML elements.

Each tag describing logical structure (*article*, *section*, *p*, etc.) defines elements that corresponds to a part of a document. Each element will be indexed. In the example, document D_2 is indexed by five elements: an *article* (tag <article>), a *section* (tag <section>) and three *paragraphs* (tag).

We note:

- $E = \{e_j, j = 1, ..., l\}$, the set of the logical elements available in the collection (*article*, *section*, *p*, etc.);
- $T = (t_1, ..., t_i, ...t_n)$, a term index built from E;
- $B = \{b_1, ..., b_k, ..., b_m\}$, the set of tags.

Let E_j be a vector of random variables T_{ij} in $\{0, 1\}$:

$$E_j = (T_{10}, ..., T_{1k}, ..., T_{1m}, ..., T_{i0}, ..., T_{ik}, .., T_{im}, ..., T_{n0}, ..., T_{nk}, .., T_{nm})$$

$$\text{with } \begin{cases} T_{ik} = 1 \text{ if the term } t_i \text{ appears in this element labeled by } b_k \\ T_{ik} = 0 \text{ if the term } t_i \text{ does not appear labeled by } b_k \\ T_{i0} = 1 \text{ if the term } t_i \text{ appears without being labeled by a tag in } B \\ T_{i0} = 0 \text{ if the term } t_i \text{ does not appear without being labeled} \end{cases}$$

We note $e_j = (t_{10}, ..., t_{1k}, ..., t_{1m}, t_{i0}, ..., t_{ik}, .., t_{im}, t_{n0}, ..., t_{nk}, .., t_{nm})$ a realization of the random variable E_j.

In the previous example with three documents, we have b_1 = *article*, b_2 = *section*, b_3 = *p*, b_4 = *b* and T = $\{t_1,..,t_5\}$.

The element e_1: t_1 t_2 t_3 of D_0 can be represented by the vector:

$$\{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{20}, t_{21}, ..., t_{53}, t_{54}\} = \{0, 1, 0, 1, 0, 0, 1, ..., 0, 0\}$$

since the term t_1 is labeled by *article* ($t_{11} = 1$), and *p* ($t_{13} = 1$) but neither by *section* ($t_{12} = 0$) nor by *b* ($t_{14} = 0$). We have $t_{10} = 0$ since the term does not appear without tag.

3.2 Term Based Score for an XML Element to Be Relevant

In the classic probabilistic model, the relevance of an element for a given query is function of the weights of the matching terms (*i.e.* terms of the query contained in the element). The weighting function BM25 [6], is broadly used to evaluate this weight, noted w_{ij}, of a term t_i in an element e_j. The term based relevance f_{term} of e_j is given by:

$$f_{term}(e_j) = \sum_{t_{ik} \in e_j} t_{ik} * w_{ij} \tag{1}$$

Given this classical model, the goal is now to propose an extension that will take into account the documents structure.

3.3 Tag Based Score for an XML Element to Be Relevant

In this section, we adapt the model introduced in [6] in order to take into account the documents structure described previously (cf. section 3.1). To do so, we not only consider term weights w_{ij}, but also tag weights.

In an information retrieval context, we want to estimate the relevance of an XML element e_j given a query. We thus want to estimate:

$P(R|e_j)$: the probability of finding a relevant information (R) given an element e_j and a query.
$P(NR|e_j)$: the probability of finding a non relevant information (NR) given an element e_j and a query.

Let $f_1(e_j)$ be a document ranking function:

$$f_1(e_j) = \frac{P(R|e_j)}{P(NR|e_j)}$$

The higher $f_1(e_j)$, the more relevant the information presented in e_j. Using Bayes formula, we get:

$$f_1(e_j) = \frac{P(e_j|R) \times P(R)}{P(e_j|NR) \times P(NR)}$$

The term $\frac{P(R)}{P(NR)}$ being constant over the collection for a given query, it will not change the ranking of the documents. We therefore define f_2 – which is proportional to f_1 – as:

$$f_2(e_j) = \frac{P(e_j|R)}{P(e_j|NR)}$$

Using the Binary Independence Model assumption, we have:

$$P(E_j = e_j|R) = \prod_{t_{ik} \in e_j} P(T_{ik} = t_{ik}|R) \quad (2)$$

$$= \prod_{t_{ik} \in e_j} P(T_{ik} = 1|R)^{t_{ik}} \times P(T_{ik} = 0|R)^{1-t_{ik}} \quad (3)$$

In the same way, we get :

$$P(E_j = e_j|NR) = \prod_{t_{ik} \in e_j} (P(T_{ik} = 1|NR))^{t_{ik}} \times (P(T_{ik} = 0|NR))^{1-t_{ik}} \quad (4)$$

For sake of notation simplification, we note, for a given XML element:

$p_0 = P(T_{i0} = 0|R)$**:** the probability that t_i does not appear without being labeled, given a relevant element.
$p_{ik} = P(T_{ik} = 1|R)$**:** the probability that t_i appears labeled by b_k, given a relevant element.
$q_0 = P(T_{i0} = 0|NR)$**:** the probability that t_i does not appear without being labeled, given a non relevant element.

$q_{ik} = P(T_{ik} = 1|NR)$**:** probability that t_i appears labeled by b_k, given a non relevant element.

Using these notations in equations 3 and 4, we get:

$$P(e_j|R) = \prod_{t_{ik} \in e_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}},$$
$$P(e_j|NR) = \prod_{t_{ik} \in e_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}.$$

The ranking function $f_2(e_j)$ can then be re-written:

$$f_2(e_j) = \frac{\prod_{t_{ik} \in e_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}}{\prod_{t_{ik} \in e_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}}$$

The *log* function being monotone increasing, taking the logarithm of the ranking function will not change the ranking. We can then define f_3 as:

$$\begin{aligned} f_3(e_j) &= \log(f_2(e_j)) \\ &= \sum_{t_{ik} \in e_j} (t_{ik} \log(p_{ik}) + (1 - t_{ik}) \log(1 - p_{ik}) - t_{ik} \log(q_{ik}) - (1 - t_{ik}) \log(1 - q_{ik}) \\ &= \sum_{t_{ik} \in e_j} t_{ik} \times \left(\log \left(\frac{p_{ik}}{1 - p_{ik}} \right) - \log(\frac{q_{ik}}{1 - q_{ik}}) \right) + \sum_{t_{ik} \in e_j} \log(\frac{1 - p_{ik}}{1 - q_{ik}}) \end{aligned}$$

As before, the term $\sum_{t_{ik} \in e_j} \log(\frac{1-p_{ik}}{1-q_{ik}})$ is constant with respect to the collection (independent of t_{ik}). Not considering it will not change the ranking provided by $f_3(e_j)$:

$$f_{tag}(e_j) = \sum_{t_{ik} \in e_j} t_{ik} \log \left(\frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \right) \tag{5}$$

Thus, we obtain in this ranking function, a weight for each term t_i and each tag b_k. The weight of a term t_i labeled by b_k will be written w'_{ik}:

$$w'_{ik} = \log(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})})$$

Finally, in our probabilistic model that takes into account the document structure, the relevance of an XML element e_j, relatively to the tags, is defined through $f_{tag}(e_j)$:

$$f_{tag}(e_j) = \sum_{t_{ik} \in e_j} t_{ik} \times w'_{ik}$$

In practice, we have to estimate the probabilities p_{ik} and q_{ik}, $i \in \{1, .., n\}$, $k \in \{0, .., m\}$ in order to evaluate the element relevance. For that purpose, we used a learning set LS in which elements relevance for a given query is known. Given the set R (respectively NR) that contains the relevant elements (respectively non relevant ones) a contingency table can be built for each term t_i labeled by b_k:

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

with:

- r_{ik}: the number of times term t_i labeled by b_k is relevant in LS;
- $\sum_i r_{ik}$: the number of relevant terms labeled by b_k in LS.
- n_{ik}: the number of times term t_i is labeled by b_k in LS;
- $r'_{ik} = n_{ik} - r_{ik}$: the number of times term t_i labeled by b_k is not relevant in LS;
- R = $\sum_{ik} r_{ik}$: the number of relevant terms in LS;
- N-R = $\sum_{ik} r'_{ik}$: the number of non relevant terms in LS.

We can now estimate $\begin{cases} p_{ik} = P(t_{ik} = 1|R) & = \frac{r_{ik}}{R} \\ q_{ik} = P(t_{ik} = 1|NR) & = \frac{n_{ik} - r_{ik}}{N - R} \end{cases}$

And w'_{ik} follows:

$$w'_{ik} = \log(\frac{\frac{r_{ik}}{R}(1 - \frac{n_{ik} - r_{ik}}{N - R})}{\frac{n_{ik} - r_{ik}}{N - R}(1 - \frac{r_{ik}}{R})}) = \log(\frac{r_{ik} * (N - n_{ik} - R + r_{ik})}{(n_{ik} - r_{ik}) * (R - r_{ik})}). \quad (6)$$

3.4 Combining Term Based and Tag Based Scores

In order to estimate the relevance of an element e_j given a query, a global ranking function $fc(e_j)$ combining terms weights used in $f_{term}(e_j)$ and tags weights used in $f_{tag}(e_j)$, is introduced:

$$fc(e_j) = \sum_{t_{ik} \in e_j} w_{ij} \times C_k(w'_{ik})$$

where C is the function used to combine terms weights and tags weights.

We experiment different ways of combining terms weights and tags weights, in other words several functions C.

In the first one, called PSPM, the weight w_{ij} of each term t_i in e_j is multiplied with the weights w'_{ik} of the tags that label this term. More formally:

$$f_{PSPM}(e_j) = \sum_{t_{ik} \in e_j} w_{ij} \times \prod_{k/t_{ik}=1} w'_{ik}$$

We can note that some tags will reinforce the weight of the term ($w'_{ik} > 1$) while other will weaken it ($w'_{ik} \leq 1$).

The second model, called CSPM (for Closest Structured Probabilistic Model), only considers the weight w'_{ic} of the tag b_c that tags the term t_i and that is the closest to t_i.

$$f_{CSPM}(e_j) = \sum_{t_{ik} \in e_j} w_{ij} \times w'_{ic}$$

In the third model, called ASPM (for Average Structured Probabilistic Model) the weight w_{ij} of each term t_i in e_j is multiplied with the average of the weights w'_{ik} of the tags that label this term.

$$f_{ASPM}(e_j) = \sum_{t_{ik} \in e_j} w_{ij} \times \frac{\sum_{k/t_{ik}=1} w'_{ik}}{|\{k/t_{ik} = 1\}|}$$

These strategies have been evaluated on the INEX 2006 & 2007 collections.

4 Experiments on INEX 2006 and 2007 Collection

4.1 INEX Collection

We used for our experimentations the INEX (Initiative for Evaluation of XML Retrieval) collection as it contains a significant amount of data together with the availability of relevant assessments.

The corpus contains 659,388 articles in English, from the free Wikipedia encyclopaedia. The documents are strongly structured as they are composed of 52 millions XML elements. Each XML article view as a tree contains, on average, 79 elements for an average depth of 6.72. Moreover, whole articles (textual content + XML structure) represent 4.5 Gb while the textual content weights only 1.6 Gb. The structural information thus represents more than twice the size of the textual one.

A set of queries is submitted by the participants during INEX 2006 competition (125 queries) and 2007 competition (130 queries). In order to evaluate information retrieval systems, the INEX campaign made available the relevance assessments corresponding to the 114 queries in 2006, and to the 107 queries in 2007.

4.2 Experimental Protocol

The corpus enriched by the INEX 2006 assessments is used as the LS training set in order to estimate the tags weights w'_{ik}.

The queries of INEX 2007 are then processed. The vector space model using BM25 weighting function is used as the baseline, without stemming nor stoplist. In order to understand the pro and cons of our structured document model, BM25 is also used as the term weighting function before integrating the tags weights.

We have evaluated our approach using the 107 assessed queries of INEX 2007. The evaluation measures used are the *precision* and *recall* measures as defined by [8].

The *interpolated average precision* (AiP), introduced by INEX, combines *precision* and *recall*, and provides an evaluation of the system results for each query. By averaging the AiP values on the set of queries, an overall measure of performance is defined [4]. This average is called *interpolated mean average precision* (MAiP).

4.3 Results

We have manually selected 14 tags in order to define the XML elements to consider. These logical structure tags will be the retrieval units, i.e. the tags considered during

Table 1. Tags frequencies (top 20)

collectionlink	16645121	normallist	1087545
item	5490943	row	954609
unknownlink	3847064	outsidelink	841443
cell	3814626	languagelink	739391
p	2689838	name	659405
emph2	2573195	body	659396
template	2396318	article	659389
section	1575519	conversionwarning	659388
title	1558235	br	378990
emph3	1484568	td	359908

the indexing step and therefore the tags the system will be able to return. These tags are *article, body, p, section, table, normallist, numberlist, title, row, td, tr, caption, definitionitem, th*.

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 (cf. table 1) and then manually removed the 6 we considered not relevant (*e.g.* br, hr, value, ...).

The weights of the 55 remaining tags were computed according to equation 6.

Table 2 presents the top 6 tags and their weights, together with the weakest 6 ones and their weights.

Table 2. Weight of the 6 strongest and 6 weakest tags

Top strongest weights		Top weakest weights	
h4	11,52	emph4	0,06
ul	2,92	tt	0,07
sub	2,34	font	0,08
small	2,21	big	0,08
strong	2,16	em	0,11
section	2,03	languagelink	0,12

We now compare the results obtained on the 107 queries of the INEX 2007 collection using our baseline and the three variants of our structured probabilistic model. Only BM25 and PSPM were submitted as official runs to INEX 2007, but all the results were computed using INEX evaluation programs (version 2, february 2008).

The results are synthesized either in table 3 or figure 1.

As can be seen, the BM25 baseline obtains a 5.32% MAiP, while PSPM obtains a 2.63% MAiP and ASPM 5.77% MAiP. The baseline is outperformed by our model ASPM, but produces better results than PSPM. Our interpretation of the latter is that multiplication impacts too strongly on small weights: two or three tags having small weights are enough to delete a term from the corpus, decreasing its weight to zero.

ASPM, that takes into account all tags by averaging their weights, performs slightly better than BM25 baseline. We can also notice that ASPM also performs better than

Table 3. MAiP of the three models evaluated on the 2007 collection

Model	@0.00	@0.01	@0.05	@0.10	@0.90	@1.00	MAiP	Rank
BM25 (baseline)	**0.4195**	**0.3221**	**0.2142**	**0.1530**	0.0004	0.0000	0.0532	63th
PSPM: all tags (weights product)	0.2266	0.1813	0.1100	0.0729	0.0000	0.0000	0.0263	72th
CSPM: closest tags only	0.1426	0.1426	0.1405	0.1271	0.0027	0.0000	0.0529	
ASPM: all tags (average weights)	0.1611	0.1611	0.1584	0.1455	**0.0027**	**0.00001**	**0.0577**	

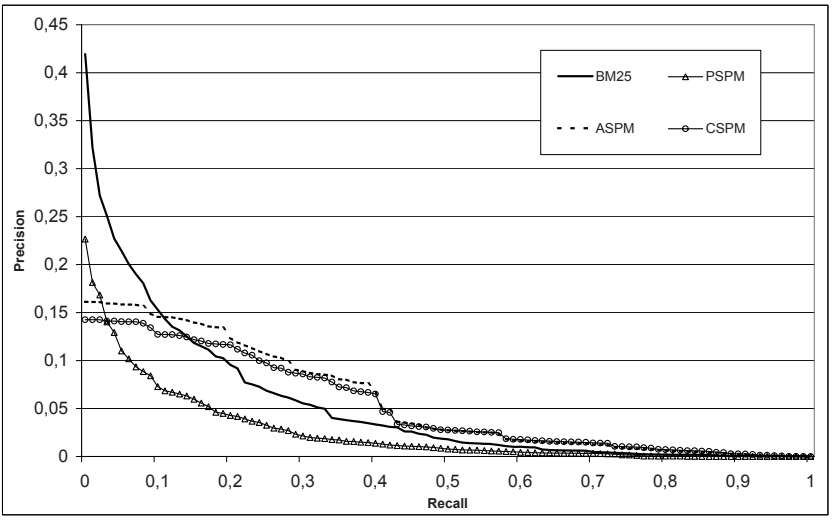

Fig. 1. MAiP of the three models evaluated on the 2007 collection

CSPM (method that only takes into account the closest tag). This is important since it shows that tags have somehow a long term dependency impact.

Our model is outperformed by the BM25 baseline at low recall levels (between P@0.00 and P@0.01), but it outperforms the BM25 baseline at other recall levels. In particular, we can see in table 3 that our model gives better results for P@0.90 and P@1.00. The precision of our model at P@1.00 is very low, but greater than zero. Each run submitted to INEX is composed by a ranked list of 1500 XML elements maximum. That means that our model is sometimes able to retrieve all the relevant elements among the first 1500 XML returned elements.

In order to better consider this fact in recall/precision curves, we have estimated $R[1500]$, the recall at 1500 elements, and we have used this score to normalize the recall-precision scores:

$$nMAiP = R[1500] * MAiP$$

These results show that our model outperforms the baseline (4.01% nMAiP versus 0.95% nMAiP), when exhaustivity (i.e. $R[1500]$) is considered (cf. table 4).

Table 4. R[1500] and nMAiP of the three models evaluated on the 2007 collection

Model	R[1500]	nMAiP
BM25 (baseline)	0.1778	0.0095
PSPM: all tags (weights product)	0.1255	0.0033
CSPM: closest tags only	**0.7026**	0.0372
ASPM: all tags (average weights)	0.6951	**0.0401**

4.4 Time Requirements

The learning step (resp. the indexing step using tags weights, and the querying step) took about 57 hours (resp. 55 hours and 3 hours), mainly due to XML Parsing. Each of these computations, parallelized on 18 PCs with 1.5Ghz-5Ghz processors and 512Mb-3Gb memory, took about 6 hours (resp. 5 hours and 12 minutes) of real time.

5 Conclusion

We proposed in this article a new way of integrating the XML structure in the classic probabilistic model. We consider both logical and 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 a weight is computed for each formatting tag, based on the probability that this tag caracterizes relevant term. During the querying step, the relevance for 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 label the term.

This model was evaluated on the INEX 2007 collection.

Our structured probabilistic model ASPM outperforms slightly a classical BM25 baseline. Moreover, experiments that takes into account the recall reached at 1500th rank ($R[1500]$) show that our model is better at high exhaustivity levels. We think that it is very important to integrate criteria like $R[1500]$ in evaluation measures. Indeed, in case of high exhaustivity is needed, it could be better to retrieve 69.51% of relevant information with 1500 elements (ASPM), than 17.78%, even with a better precision at low recall levels (BM25).

Beyond the fact that our structured probabilistic model ASPM outperforms a classical BM25 baseline, experiments with CSPM suggest that long term dependencies exist between tags and terms.

In a near future, we plan to analyze and take advantage of contextual information (*e.g.* long term-to-tag dependency, relationship of the tag in respect to other tags, etc.) and hope to obtain much better results. This can be done either from a practical point of view (*e.g.* using machine learning methods for modeling these relationships) or from theoretical point of view (*e.g.* adapting the aggregation between term and tag weight in the structured probabilistic model).

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison-Wesley, Reading (1999)
2. Defude, B.: Etude et réalisation d'un système intelligent de recherche d'informations: Le prototype IOTA. PhD thesis, Institut National Polytechnique de Grenoble (January 1986)
3. Fourel, F.: Modélisation, indexation et recherche de documents structurés. PhD thesis, Université de Grenoble 1, France (1998)
4. Kamps, J., Pehcevski, J., Kazai, G., Lalmas, M., Robertson, S.: INEX 2007 evaluation measures. In: Fuhr, N., Lalmas, M., Trotman, A., Kamps, J. (eds.) INEX 2006. LNCS, vol. 4518, Springer, Heidelberg (2007)
5. Konopnicki, D., Schmueli, O.: W3qs: A query system for the world-wide web. In: 21th International Conference on Very Large Data Bases (VLDB 1995), September 1995, pp. 54–65 (1995)
6. Robertson, S.E., Sparck Jones, K.: Relevance weighting of search terms. Journal of the American Society for Information Sciences 27(3), 129–146 (1976)
7. Sauvagnat, K., Hlaoua, L., Boughanem, M.: XFIRM at INEX 2005: Ad-hoc and relevance feedback tracks. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 88–103. Springer, Heidelberg (2006)
8. Swets, J.A.: Information retrieval systems. Science 141, 245–250 (1963)
9. Wilkinson, R.: Effective retrieval of structured documents. In: 17th ACM Conference on Research and Development in Information Retrieval (SIGIR 1994) (July 2007)