BM25t: a BM25 extension for focused information retrieval

Mathias Géry, Christine Largeron

Knowledge and Information Systems manuscript No.
(will be inserted by the editor)

BM25t: a BM25 extension for focused information retrieval

Mathias Géry · Christine Largeron

Received: Jul 21, 2010 / Revised: Mar 15, 2011 / Accepted: Apr 23, 2011

Abstract This paper addresses the integration of XML tags into a term-weighting function for focused XML Information Retrieval (IR). Our model allows us to consider a certain kind of structural information: tags that represent a logical structure (*e.g. title*, *section*, *paragraph*, etc.) as well as other tags (*e.g. bold*, *italic*, *center*, etc.). We take into account the influence of a tag by estimating the probability for this tag to distinguish relevant terms from the others. Then, these weights are integrated in a term-weighting function. Experiments on a large collection from the INEX 2008 XML IR evaluation campaign showed improvements on focused XML retrieval.

Keywords Probabilistic Information Retrieval model · Structured Information Retrieval · XML · Tags · Weighting scheme · BM25

1 Introduction

With the development of markup languages, most of the information available on the Internet has become very structured. This has launched the development of focused information retrieval (focused IR) which aims to provide fragments of documents rather than whole documents as in classic information retrieval. The information specifically relevant to the user's needs is then identified directly within the documents. This is useful especially when the documents are long or in the context of mobile computing (*e.g.* smartphones, tablets, etc.). Depending on whether the list which is returned contains passages or XML elements, we may then speak more specifically of either passage retrieval or XML retrieval (XML IR) [26, 27]. The field of XML IR has been encouraged over the past few years through the organization of workshops and competitions such as INEX [1, 3, 8, 19, 21, 46]. However, markup languages such as XML do not only allow a document to be broken up into elements; they can also be used to annotate the text with tags so that the structure (logical, layout, formatting, links, etc.) may be described independently from the content itself.

Mathias Géry · Christine Largeron
Université de Lyon, F-42023, Saint-Etienne, France;
CNRS, UMR 5516, Laboratoire Hubert Curien, F-42000, Saint-Etienne, France;
Université de Saint-Etienne, Jean-Monnet, F-42000, Saint-Etienne, France;
E-mail: {mathias.gery, christine.largeron}@univ-st-etienne.fr
Tel: (+33)4 77 91 57 56; Fax:(+33)4 77 91 57 81

Hence, studies in XML IR have not only been concerned with the retrieval of more concise units of information but also with the better exploitation of these tags in order to improve the detection of relevant information.

For these purposes, two types of approach have been adopted. The first one, which is user-oriented, is concerned with the development of interfaces for the visualization and navigation within results and also query-languages such as W3QS [23], XIRQL [5], NEXI [44, 45], Bricks [51] or BusEngine-L [40] which take into account the structure. However, the use of such query-languages remains limited, because few users are able to formulate their needs with complex queries[1]. Most of the time, those queries are expressed with a few keywords [17, 22, 31].

The second type of approach improves the classic models and suggests a scheme for structural weighting [10, 25]. Indeed, as pointed out by Tamine-Lechani *et al.* or by Zhu *et al.* the retrieval accuracy can be improved in taking into account the structure to represent documents [41, 50]. Within such a weighting scheme, the weight assigned to a word is not only based upon its frequency within the document and within the collection, but also upon its position within the document. The tags are used to define these positions. Hence, the ranking of a document depends not only upon the existence of a term within a document but also upon the tags which mark the term. Different tags can be considered, including tags related to formatting (*e.g. bold*, *italic*, *center*, etc.) and logical tags that define an internal structure (*e.g. title*, *section*, *paragraph*, etc.).

In the XML document presented in figure 1, there are logical tags like *article*, *p* and other tags like *strong*, *emph3*, *collectionlink* which might emphasize the important terms.

```
<?xml version="1.0" encoding="UTF-8"?>
<article>
    <name id="5432">Economy of Cambodia</name>
    <emph3>
        <collectionlink href="9223.xml">Economy</collectionlink> - overview:
    </emph3>
    <p>
        During <collectionlink href="34658.xml">1995</collectionlink>, the <strong>Cambodia
government</strong> implemented firm stabilization policies under difficult circumstances. Overall,
<strong>macroeconomic performance was good</strong>. Growth in 1995 was estimated at 7% because of improved agricultural production (<collectionlink href="36979.xml">rice</collectionlink>
in particular). Strong growth in <collectionlink href="239038.xml">construction</collectionlink>
and services continued. Inflation dropped from 26% in 1994 to only 6% in 1995. Imports increased as a
result of the availability of external financing. Exports also increased, due to an increase in log exports.
With regard to the budget, both the current and overall deficits were lower than originally targeted.
    </p>
    <p>
        After four years of solid macroeconomic performance, Cambodia's economy slowed dramatically in <collectionlink xlink:href="34601.xml">1997</collectionlink>.
    </p>
</article>
```

Fig. 1 Tags in XML article "Economy of Cambodia" from Wikipedia

In this article, we consider the problem of extending the probabilistic model [29, 34] that aims to estimate the relevance of a document for a given query through two probabilities: the probability of finding relevant information and the probability of finding non-relevant

[1] Example: "I am looking for a paragraph about running, taken from an article about marathons containing a photograph of a marathon runner."

information. The model is extended using all kind of XML tags. We suppose that both types of tags may be used to emphasize words:

- A word is undoubtedly more important if it appears within certain sections of a document (a title, a caption to a figure, a paragraph, etc.)
- In the same way, a word does not carry the same emphasis if it is marked by any kind of tags, especially if it appears in a particular font (bold, italic, etc.).

Consequently, in the model which we propose, the document structure is integrated at two levels. At the first level, the logical structure is used in order to determine the granularity of the indexing, and thereby the granularity of the elements with which the system is likely to provide to the user. Therefore, the relevance is not evaluated, as is usually the case, at the document level, but rather at the level of the XML elements. Then at the second level, the logical structure and other kinds of structures are integrated in the weighting scheme. During the learning stage, a weight is given to each tag. This weight is based upon the probability that this tag highlights either a relevant term or a non-relevant term. The underlying idea is the same as in the probabilistic model [34] which estimates the probability that a term appears in a relevant (or non-relevant) document, from a test collection where the relevance of the documents is available. At the query stage, the probability that an element might be relevant is estimated by combining the weight of the terms that it contains with the weight of their tags.

The main contribution of this paper is the following:

- A theoretical framework which explicitly takes the logical tags and other tags as found in XML documents into account, by removing the limitations on the number of tags which are considered, unlike Robertson *et al.* [33].
- A learning stage to estimate of the weight of each tag which measures its capacity to emphasize terms in relevant passages or in non relevant passages. As the weights may have a negative impact, this stage may also be considered as a stage where the tags are automatically selected.
- The extension of the BM25 weighting function [15, 16, 35] through the integration of automatically learned tag weights.
- The retrieval of elements whose granularity is better adapted, especially useful for mobile applications, unlike approaches which aim to improve the retrieval of whole documents.
- An evaluation of this model upon a wide collection of documents (the INEX[2] collection)

This model will be described in the following section. The experiments are presented in the section 3 and the results are reported in section 4 and in section 5. A state of the art is given in section 6 before the conclusion.

2 A probabilistic model for the representation of structured documents

2.1 Notations

Let $\mathcal{D}$ be a set of structured documents. Each tag describing logical structure (*article*, *section*, *p*, *table*, etc.) defines a logical element that corresponds to a part of a document. Therefore, each logical element will be represented by a set of terms and will be indexed. These elements will be returned by the system.

[2] INitiative for Evaluation of XML Retrieval. See `http://www.inex.otago.ac.nz`

On the other hand, some tags are not considered as logical tags, as for example the formatting tag . This tag does not define an element to be indexed. However, it belongs to the user to define the list of logical tags and the minimum elements size.

We note:

- $E = \{e_1, ..., e_j, ...e_l\}$, the set of the logical elements available in the collection;
- $T = \{t_1, ..., t_i, ...t_n\}$, a term index built from E;
- $B = \{b_1, ..., b_k, ..., b_m\}$, the set of tags.
- $B_l \subset B$, the set of logical tags;
- $\overline{B_l}$, the set of all the other tags.

In the following sections, the representation of an element e_j is noted x_j when only the terms are considered, and m_j when both terms and tags are taken into account. The term based score and the tag based score are respectively detailed in the next sections.

2.2 Example

The following example presents three documents d_0, d_1 and d_2. Considering a query q, we suppose that the seven underlined terms are relevant:

d_0	d_1	d_2
<article>	<article>	<article>
$\underline{t_1 t_2} t_3$	<section>	<section>
<section>	$t_1 t_4$	t_5
$\underline{t_1} t_4$	$t_2 t_5$	$\underline{t_3 t_4}$
$\underline{t_2 t_5}$	</section>	$t_3 t_5$
</section>	$t_2 t_1$	</section>
</article>	</article>	</article>

We assume that in this example the set of logical tags $B_l = \{article, section, p\}$. The document d_2 is thus indexed by five elements, as presented in figure 2: an *article* (tag <article>), a *section* (tag <section>) and three *paragraphs* (tag). This example shows that the content of elements included in larger ones are indexed several times, e.g. each paragraph is indexed itself as element, and its content is also included in the index of both elements <article> and <section>.

2.3 Term based relevance score for an XML element

The relevance of an element for a given query Q depends upon the weights of the matching terms (*i.e.* terms of the query contained in the element). The weight of the term t_i in the element x_j is noted w_{ji}.

Formally, we define X_j a vector of random variables and $x_j = (x_{j1}, ..., x_{ji}, ..., x_{jn})$ a realization of the vector X_j, with $x_{ji} = 1$ (resp. 0) if terms t_i appears (resp. does not appear) in element e_j.

Given these notations, the term based relevance f_{term} of x_j is given by the score:

$$f_{term}(x_j) = \sum_{t_i \in T \cap Q} x_{ji} \times w_{ji} \quad (1)$$

$T = \{t_1, t_2, t_3, t_4, t_5\}$
$E = \{$`d0/article[1]`, `d0/article[1]/p[1]`, `d0/article[1]/section[1]`,
`d0/article[1]/section[1]/p[1]`, `d0/article[1]/section[1]/p[2]`,
`d1/article[1]`, `d1/article[1]/section[1]`, `d1/article[1]/section[1]/p[1]`,
`d1/article[1]/section[1]/p[2]`, `d1/article[1]/p[1]`,
`d2/article[1]`, `d2/article[1]/section[1]`, `d2/article[1]/section[1]/p[1]`,
`d2/article[1]/section[1]/p[1]/b[1]`, `d1/article[1]/section[1]/p[2]`,
`d2/article[1]/section[1]/p[3]`$\}$
$B = \{$`article`, `section`, `p`, `b`$\}$
$B_l = \{$`article`, `section`, `p`$\}$
$\overline{B_l} = \{$`b`$\}$
$|d_0| = 7; |d_1| = 6; |d_2| = 5;$
...
$|d_2/article[1]| = 5$
$|d_2/article[1]/section[1]- = 5$
$|d_2/article[1]/section[1]/p[1]| = 1$
$|d_2/article[1]/section[1]/p[2]| = 2$
$|d_2/article[1]/section[1]/p[3]| = 2$

Fig. 2 Modelling documents d_0, d_1 and d_2

As mentioned by Robertson *et al.* [33], this general dot-product form covers different ranking functions, for example the functions `ltn` and `ltc` implemented by SMART system [36], or the well known `BM25` function [15, 16, 35].

Preliminary experiments using `ltn` and `ltc` have led to weak results [11], thus we will henceforth only consider `BM25` [15, 16, 35]:

$$w_{ji} = \frac{tf_{ji} \times (k_1 + 1)}{k_1 \times ((1 - b) + (b * ndl)) + tf_{ji}} \times \log \frac{N - df_i + 0.5}{df_i + 0.5} \tag{2}$$

with:

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

Parameter k_1 allows setting the frequency saturation. Parameter b allows setting the importance of ndl, *i.e.* the importance of length normalization.

We can note that parameters k_1 and b allow modifying the bend of the curve and in some way the non linearity of the function. As an example, when k_1 is set to 1.1, a term frequency of 10 will lead to almost the same value in the tf component of the BM25 weighting function, than a term frequency of 25.

This non linearity property of weighting functions is very important for our purpose. Indeed, impact of tag weight on w_{ji} is very different than on the tf_{ji}. Like Robertson *et al.* [33], we think that the impact on the term weighting function should not break the non linearity property, and thus we have compared the pre-impact of tag weights (*i.e.* impact on tf_{ji} in the TTF strategy) as well as the post-impact (*i.e.* impact on w_{ji} in the CLAW strategy). These different strategies are detailed in following sections.

2.4 Tag based relevance score for an XML element

Similarly to the previous section, we define M_j as a vector of random variables[3] T_{ik} in $\{0,1\}$:

$$M_j = (T_{10}, ..., T_{1k}, ..., T_{1m}, ..., T_{n0}, ..., T_{nk}, .., T_{nm})$$

with
$T_{ik} = 1$ if term t_i appears in this element marked by b_k
$T_{ik} = 0$ if term t_i does not appear marked by b_k
$T_{i0} = 1$ if term t_i appears without being marked by a tag in B
$T_{i0} = 0$ if term t_i does not appear without being marked

We note $m_j = (t_{10}, ..., t_{1k}, ..., t_{1m}, ..., t_{n0}, ..., t_{nk}, .., t_{nm})$ a realization of the random variable M_j.

In the running example given above, we have b_1 = *article*, b_2 = *section*, b_3 = *p*, b_4 = *b* and T = $\{t_1,..,t_5\}$. The element: e_j = t_1 t_2 t_3 of d_0 can be represented by the vector:

$$\begin{aligned} m_1 &= \{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{20}, t_{21}, ..., t_{53}, t_{54}\} \\ &= \{0, 1, 0, 1, 0, 0, 1, ..., 0, 0\} \end{aligned}$$

as the term t_1 is marked 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 $t_{10} = 0$ as the term does not appear without tag.

In this section, we adapt the model introduced by Robertson *et al.* [34] in order to take into account the documents structure described previously (cf. section 2.1). To do so, we not only use terms weights w_{ji}, but also tag weights.
In an information retrieval context, we wish to estimate the relevance of an XML element e_j (modelled by the vector m_j) for a given query. We thus want to estimate:

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

Let $f_1(m_j)$ be a document-ranking function:

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

The higher $f_1(m_j)$, the more relevant the information represented by m_j. Using Bayes formula, we get:

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

The term $\frac{P(R)}{P(NR)}$ being constant throughout 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:

[3] M for Mark up. Random variables M_j and its realizations m_j represent structured elements.

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

Using the Binary Independence Model assumption, we have:

$$\begin{aligned} P(M_j = m_j|R) &= \prod_{t_{ik} \in m_j} P(T_{ik} = t_{ik}|R) \\ &= \prod_{t_{ik} \in m_j} P(T_{ik} = 1|R)^{t_{ik}} P(T_{ik} = 0|R)^{1-t_{ik}} \end{aligned} \tag{3}$$

In the same way, we get:

$$P(M_j = m_j|NR) = \prod_{t_{ik} \in m_j} P(T_{ik} = 1|NR)^{t_{ik}} P(T_{ik} = 0|NR)^{1-t_{ik}} \tag{4}$$

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

$p_{i0} = P(T_{i0} = 0|R)$: the probability that t_i does not appear without being marked, given a relevant element.

$p_{ik} = P(T_{ik} = 1|R)$: the probability that t_i appears marked by the tag k, given a relevant element.

$q_{i0} = P(T_{i0} = 0|NR)$: the probability that t_i does not appear without being marked, given a non relevant element.

$q_{ik} = P(T_{ik} = 1|NR)$: the probability that t_i appears marked by the tag k, given a non relevant element.

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

$$\begin{aligned} P(m_j|R) &= \prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}, \\ P(m_j|NR) &= \prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}. \end{aligned}$$

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

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

The *log* function being monotone increasing, taking the logarithm of the ranking function will not change the ranking. This leads to the function f_3:

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

As before, the term $\sum_{t_{ik} \in m_j} \log(\frac{1-p_{ik}}{1-q_{ik}})$ is constant in respect to the collection (independently from t_{ik}). Not considering it, will lead to the ranking function $f_3(m_j)$:

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} * \log\left(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}\right) \quad (5)$$

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

$$w'_{ik} = \log(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}) \quad (6)$$

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

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} \times w'_{ik} \quad (7)$$

This formula is similar to the classical term weighting function seen in equation 1, except that tag weights are considered instead of term weights.

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 this purpose, we propose to use a learning set LS in which element 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 marked by b_k:

Table 1 Contingency table for the term t_i and for the tag b_k

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

with:

- r_{ik}: the number of times term t_i marked by b_k is relevant in LS;
- $\sum_i r_{ik}$: the number of relevant terms marked by b_k in LS.
- n_{ik}: the number of times term t_i is marked by b_k in LS;
- $nr_{ik} = n_{ik} - r_{ik}$: the number of times term t_i marked by b_k is not relevant in LS;
- $R = \sum_{ik} r_{ik}$: the number of relevant terms in LS;
- $|NR| = N - R$: 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:

$$w'_{ik} = \log \frac{\frac{r_{ik}}{R}(1 - \frac{n_{ik} - r_{ik}}{N - R})}{\frac{n_{ik} - r_{ik}}{N - R}(1 - \frac{r_{ik}}{R})} \quad (8)$$

$$= \log \frac{r_{ik} \times (N - n_{ik} - R + r_{ik})}{(n_{ik} - r_{ik}) * (R - r_{ik})}$$
$$= \log \frac{r_{ik} \times (|NR| - nr_{ik})}{nr_{ik} \times (R - r_{ik})}$$

This weighting function evaluates, for a given tag, the probability of being able to distinguish between relevant and non-relevant terms: it increases according to the tag ability to distinguish a relevant term. In practice, the learning set contains a set of queries and consequently the tag weights are evaluated with an aggregation of the contingency table over all the queries.

Considering the tag $b_2 = section$ in our example, we obtain the following contingency table for the query q and the term t_1:

Table 2 Example: contingency table for the term t_1 and for the tag b_2 = section

	R	NR	$LS = R \cup NR$
$t_{1,2} \in m_j$	$r_{1,2} = 1$	$nr_{1,2} = 1$	$n_{1,2} = 2$
$t_{1,2} \notin m_j$	$R - r_{1,2} = 6$	$N - n_{1,2} - R + r_{1,2} = 10$	$N - n_{1,2} = 16$
Total	$R = 7$	$\|NR\| = 11$	$N = 18$

Then, we can calculate the weight $w'_{1,2}$ of p related to the term t_1 as follows:

$$w'_{1,2} = \log \frac{1 \times (11 - 1)}{1 \times (7 - 1)} = \log \frac{5}{3} \quad (9)$$

Our model needs a learning set, as the probabilistic model needs. Obviously, such a learning set is not always available, and it can be challenging to build. In the INEX campaign for instance, the participants use an interface to highlight the relevant passages in function of their query. In this way, a training set of queries is composed and another set of queries is proposed as a test set. Its robustness has to be studied when the documents collection changes. However, when such a learning set exists, the probabilistic model can be used, and it has shown its effectiveness.

It should be noted however, that the estimation of probabilities may comprises some smoothing when the learning set is limited in size. This has not been useful for our experiments, thanks to the estimation of one weight for each tag instead of one weight for each pair (tag, term) (as explained in the next section).

2.5 Estimation of tag weights

From a theoretical point of view, we may estimate a weight for each pair (term, tag) (cf. equation 8), in other words the capacity of a tag to reinforce a relevant term (or on the contrary, to mitigate a non-relevant term). However, we aim to create a model for tag-impact, not in relation to a particular term, but generally. Indeed, we believe that the capacity of a tag to highlight relevant terms (or on the contrary, to reduce their visibility) is intrinsic to the tag itself and is therefore not dependent on the terms. The objective then is, for instance, to evaluate whether or not a word featuring in a title is more important than a word taken

from a section/paragraph, regardless of the word itself. We are therefore not interested in the weight of each pair (term-tag) but instead, in the tag-weight, regardless of the terms which it labels. Thus, we estimate a weight w'_k for each tag b_k instead of a weight for each couple (term t_i, tag b_k):

$$w'_k = \frac{\sum_{t_i \in T} w'_{ik}}{|T|} \tag{10}$$

2.6 Estimation of the global XML element score

Having term and tag weights, a global score can be computed for ranking the elements. We propose two strategies to integrate the tag-weights into the BM25 weighting scheme:

- CLAW[4] is an a posteriori impact-strategy in the results of BM25.
- TTF[5] is an early impact-strategy, integrating tag-weights into the BM25 function.

In the CLAW strategy, in order to take into account all the tags that mark one given term, we propose to combine linearly the average of their weights with the weight of the term itself. This combining function, noted f_{claw}, is defined by:

$$f_{claw}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} w_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{11}$$

where w_{ji} is the t_i term's weight in document m_j computed by the `BM25` function (see eq.2).

Géry *et al.* showed that the use of the tag weights raises the recall, although the improvement is not significant [11]. On the other hand, the `BM25` is non linear, the non linearity being controlled by the term frequency saturation parameter k_1 (cf. section 2.3). For this reason the integration of tag weights at a global level (*i.e.* on the w_{ji} weight) is very different than their integration into the term frequency (*i.e.* on tf_{ji}). Like Robertson *et al.* [33], we adopt an early strategy which consist in integrate the tag-weights directly into tf_{ji}. In this way, we take advantage of the non-linearity of the `BM25` function. The new term weight (tf_{ji} multiplied by the average of the weights of the tags that mark the term t_i), noted ttf, will replace the regular tf in the `BM25` function defined in equation 2.

$$ttf_{ji} = tf_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{12}$$

3 Experiments

The framework for our experiments is INEX[6], the international XML IR competition which is presented in the next section. The results obtained by our model at INEX 2008 showed the advantage of taking tags into account. These results will be presented briefly in section 4. We then conducted some more in-depth experiments to study the impact of various

[4] f_{claw}: Combining Linearly Average tag-Weights.
[5] TTF: Tagged Term Frequency.
[6] INEX: Initiative for the Evaluation of XML Retrieval

parameters on the behaviour of our model, for classical IR which aims to provide full articles (granularity: full articles) as well as for Focused IR which returns element (granularity: XML elements). The results of these experiments are presented in section 5.

3.1 INEX collection

For our experiments, we used the INEX Ad-Hoc 2008 collection, extracted from the English Wikipedia XML corpus [4]. This collection contains a significant amount of structured XML data. It also contains relevance assessments, allowing us to evaluate the quality of Focused XML IR systems.

The corpus includes 659,388 articles from the Wikipedia encyclopaedia. The original Wiki syntax was converted into XML, using both general tags for the logical structure (*e.g. article*, *section*, *paragraph*, *title*, *list*, *item*, etc.), formatting tags (*e.g. bold*, *italic*, *small*, etc.) and link tags (*e.g. collection-link*, etc.). The documents are strongly structured as they are composed of 52 million XML elements. There is no DTD fixing the available tags. Consequently, there exist 1,244 different tags in the collection, although most of them appear in very few articles. Each XML article can be viewed as a tree containing on average 79 elements and having, on average, a depth of 6.72. Moreover, whole articles (textual content + XML structure) represent 4.5 Gb of the data whereas the textual content represents only 1.6 Gb. Thus, the structural information is twice as large as the textual information.

3.2 INEX evaluation measures

The evaluation measures are based on the *precision* and *recall*, defined by Swets [39]. $iP[x]$ is the precision value at recall x. The *average interpolated precision* (AiP) combines *precision* and *recall*, calculating the average of $iP[x]$ on 101 recall points (x = 0.00; 0.01; 0.02; ... 0.99; 1.00). This measure provides an evaluation of the system's quality for each query. By averaging the AiP values over the set of queries, an overall measure of performance is defined [20]. This average is called *mean average interpolated precision* ($MAiP$).

Given that every experiment is submitted to INEX in the form of a ranked list of a maximum of 1,500 XML elements for each query, these measures, in terms of recall, are in favour of the experiments for which whole articles are retrieved (thereby providing a greater quantity of information for 1,500 documents). This is problematic in the case of Focused IR because focused answers may be penalized even though the very purpose of Focused IR is to be able to retrieve short answers (in the form of relevant elements, reduced relatively to the whole article). Taking this into account, we also calculated $R[1500]$, the recall rate for 1,500 documents, and $S[1500]$, the size (in MB) of the 1,500 documents which were found.

It should be noted that the main ranking of the INEX competition is based on $iP[0.01]$ instead of the overall MAiP measure, in order to take into account the importance of precision at low recall levels. Thus, Focused IR is evaluated according to precision rather than recall.

All the results presented here, including those of INEX systems, were computed using the INEX 2008 evaluation programs: *inex_eval*, version 1.0.

3.3 Experimental protocol

All the experiments have been carried out, based on the BM25 model [15, 16] which has been applied either in a classical way (*i.e.* indexing at the article level) or in a focused way (*i.e.* indexing at the element level). It would be interesting to evaluate our model against some other models that consider tags, as for example BM25F or BM25E, but implementing these models is very difficult upon documents collections that use hundreds of tags, as pointed out their authors (cf. section 6). We investigate the impact of tags on both levels: article and element.

3.3.1 INEX 2008 runs

In the learning stage, the 2006 INEX collection, composed of 659,388 articles, 114 queries and associated relevance judgements, was firstly used as a learning set in order to estimate tag-weights w'_k.

Following this, our indexing and querying experiments were carried out on the same 659,388 articles but using the 70 new queries from the 2008 edition of the INEX Ad-Hoc. Thus, the set of queries from INEX 2006 is used as a training set to learn the tag weights while the new set of queries from INEX 2008 is used as a testing set. Therefore, even if the same collection of documents is used in both stages: when estimating tag-weights (*i.e.* the learning stage), and during IR experiments (*i.e.* testing stage), they represent in fact two distinct collections from a IR point of view, thanks to the two different sets of queries. The problem of overfitting is thus avoided.

Only the key-words in each query were used (*title* field for INEX queries). We did not use the fields *description*, *narrative* or *castitle* (structured part of the query).

We experimented our model (CLAW and TTF) on a classical IR task, where the granularity for the answers is the whole article, as well as on a Focused IR task, where the granularity of the answers is the XML element. These experiments, presented in section 4 enabled us to demonstrate the advantage of taking tags into account for Focused IR within the context of our participation in INEX 2008 [12].

3.4 Parameter settings

Depending on the model used, different parameters need to be set. Some of them were chosen and fixed during the experiments, and other, more important ones, were studied more exhaustively:

- Fixed parameters: weighting function (BM25 [15, 16]), minimum length of returned elements, minimum length of terms, maximum depth of returned elements, stop words, *andish* mode, mandatory or banned query terms (+/- operators), set of weighted tags.
- Granularity-based parameters, which are fixed on one hand for classical IR (articles), and on the other hand for focused IR (elements): a set of logical tags (*i.e.* the kind of element the system is able to return[7]), calculation of df.
- Studied parameters: tag impact (no tags, CLAW or TTF), BM25 b and k_1.

It should be noted that a study [42] on the evaluation of parameters suggests that a training corpus composed of 100 queries and their assessments is enough to estimate the 9 parameters of the `BM25F` model that was successfully used in the TREC competition [49].

[7] The set of logical tags is reduced to only one tag, i.e. *article*, in the case of classical IR.

As previously mentioned, all the experiments were carried out with the same stop word dictionary[8], and with the same processing of the queries (*e.g.* considering query operators + (mandatory terms) and - (banned terms)).

We launched some preliminary experiments in order to estimate some important parameters. Concerning the elements, the first question we face is to define the length of the smallest element the system will be able to return. As the process of conversion from the Wikipedia to INEX corpus was automatic, some very small elements are not interesting, as they cannot contain enough information. This is the case, for example, with the XML *languagelink* elements. Moreover, some analyses on the 2006 and 2007 assessments (not presented here) showed that it is not necessary 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, which is bigger and thus easier to index). We can note that Kamps *et al.* showed that the optimal value for this parameter is around 40 [18].

Another parameter which needs to be considered, as discussed by Mass *et al.* [30], is the computation of the df: should we compute the df at an element level or should we compute an overall df (*e.g.* at article level) without regard to the element length? Computing df at the element level will introduce a great variance in the df, and each term may be considered several times (*i.e.* each time it appears in an element from the same article). On the other hand, computing the df at the document level does not allow to distinguish some elements considering also the terms distribution inside articles.

We chose to calculate the df values at article level as well as at element level, *i.e.* the df is different for elements and for articles. Note that Taylor *et al.* compute an overall df [42], and Mass *et al.* compute a df at six different levels [30].

3.5 Tag Selection

Another important parameter is the list of logical tags, *i.e.* the XML elements which will be considered by the system either at indexing time or during the query step. The system will therefore not be able to return an element that does not belong to this list. Some simple statistics about the 70 tags appearing in the relevant passages in INEX 2007 assessments helped us to select the set of 16 logical tags (B_l, cf. table 3). We first removed all tags occurrences when their father is more than 80% relevant. Then, to be selected, a logical tag should fulfill the following thresholds in the relevant passages:

- Number of occurrences ≥ 5.
- Average length ≥ 25 characters.
- Average relevance $\geq 10\%$.
- Total relevance length $\geq$ 5,000 characters.

Table 3 Set of logical tags B_l

$B_l = \{article, cadre, indentation1, item, li, normallist, numberlist, p, row, section, table, td, template, title, th, tr\}$

This set of tags is more or less the same for each INEX participant.

The 59 tags used in the weighting function are those whose occurrences exceed 300, chosen amongst 1,244 different tags appearing in the 659,388 documents (cf. table 4). This is

[8] Stop words list: 319 words from Glasgow Information Retrieval Group, cf. http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

a compromise between considering a lot of tags, which is heavy to process, and the coverage of the documents collection by these tags. Indeed, this set of tags includes all the important tags appearing in the training collection, as presented in the figure 3 showing that the tags frequencie decreases dramatically. Our set of 59 tags represents 99.99% of the 51,042,202 tags occurrencies in the collection. Only 110 tags appear more than 10 times.

Table 4 Tags frequencies (20 most frequent tags and the 7 less frequent from our set of 59 tags)

Tag	#occurrences	Logical tag	Used for weighting
collectionlink	17,018,017	-	X
item	5,684,158	X	X
unknownlink	3,946,351	-	X
cell	3,770,465	-	X
p	2,752,835	X	X
emph2	2,722,784	-	X
template	2,427,454	X	X
section	1,610,183	X	X
title	1,592,672	X	X
emph3	1,481,088	-	X
normallist	1,110,280	X	X
row	939,665	X	X
outsidelink	858,944	-	X
languagelink	739,391	-	X
name	659,406	-	X
body	659,394	-	X
article	659,388	X	X
br	383,706	-	X
td	370,975	X	X
caption	350,858	-	X
...	...	...	...
gallery	2,527	-	X
cite	2,153	-	X
indentation3	1,993	-	X
emph4	940	-	X
em	608	-	X
strong	351	-	X
h4	307	-	X

Afterwards, we manually removed 5 tags out of the 59: *article* and *body* because they mark the whole articles, *br*, *s* and *value* because they are without content. Then, the logical tags are firstly used during the indexing step to define the elements which may be returned by the system. They are secondly considered during the query step as weighting tags which impact the weight of terms. For this reason, they are marked in the third and in the fourth column in table 4, while the other tags are considered only as weighting tags and are only marked in the fourth column.

3.6 Tag Weighting

The weights of the 54 remaining tags, including 14 on 16 logical tags (as said before, *article* and *body* where removed), were computed according to equation 10. Table 5 presents the

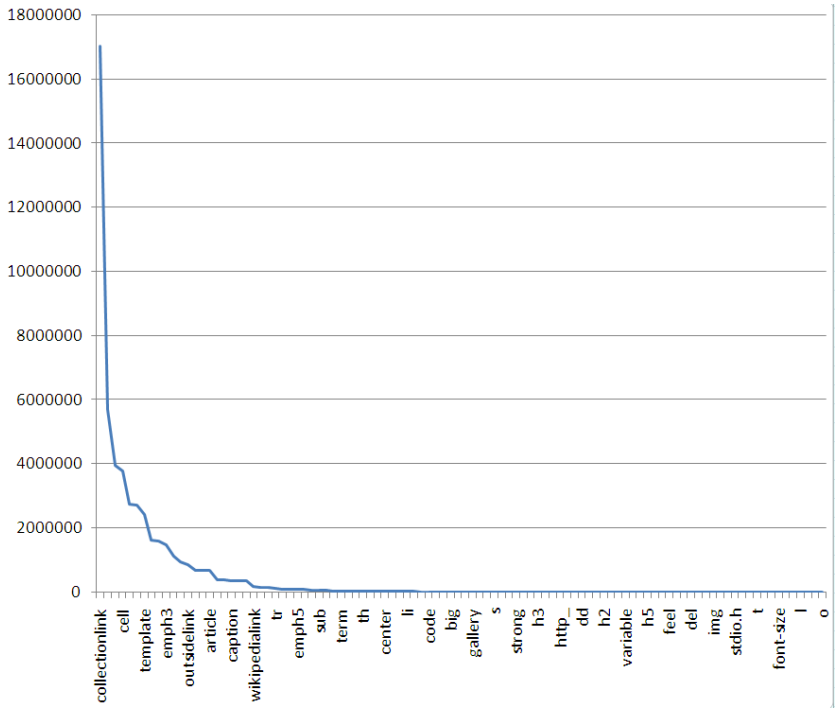

Fig. 3 Tags frequencies (top 100)

top 10 tags and their weights, together with the 10 weakest ones and their weights. Their frequencies in the whole collection are also given.

Table 5 Weight w'_k of the 10 strongest and 10 weakest tags

Weight w'_k of the 6 strongest				Weight w'_k of the 6 weakest tags			
#	Tag	Weight	#occs	#	Tag	Weight	#occs
1	h4	12.32	307	45	u	0.24	3,527
2	ul	2.70	3,050	46	i	0.23	17,935
3	sub	2.38	54,922	47	code	0.15	5,955
4	indentation1	2.04	135,420	48	span	0.15	2,592
5	section	2.01	1,610,183	49	tt	0.14	6,841
6	blockquote	1.98	4,830	50	b	0.13	11,297
7	strong	1.97	351	51	em	0.11	608
8	small	1.97	61,132	52	big	0.08	3,213
9	cadre	1.91	149,002	63	font	0.07	27,117
10	indentation2	1.82	14,065	54	emph4	0.06	940

Although tags *h4* and *strong* have a high score, their impact will be very low because these tags appear only around 300 times in the corpus. A formatting tag that might have a significant impact is the tag *small* which appears 61,132 times. We can nevertheless notice the presence of *ul*, *section* and *cadre* in the top 10 tags. The tag *section* appears more than one million times in the corpus. Its impact will thus be very important. Actually, the logical tags have more impact on the term weights than the formatting tags.

4 INEX results

We will now present the results obtained by our model during the 2008 INEX competition. We submitted three experiments (*Foc-1*, *Foc-2*, *Foc-3*) at the "Focused" Ad-hoc task. In this task, a set of 1,500 non-overlapping XML elements had to be returned. Our first objective was to obtain an efficient baseline, then to evaluate our model for classical IR and Focused IR to study the impact of having taken tag-weights into account in the BM25 function [15, 16].

Table 6 presents the mean of the evaluation mesures on the set of queries for these three experiments. Our runs were compared to FOERStep (Waterloo University, [14]) which was the winner of the Focused task. Structure is not taken into account either in Foc-1, where whole articles were returned (granularity: articles), or in Foc-2, where it is the elements which were returned (granularity: elements). However, in Foc-3, the tag-weights are integrated into BM25 for Focused IR (granularity: elements, TTF). Only 3 runs per task can be submitted to INEX. That is the reason why it was not possible to present a run Foc-4, integrating tag-weights into BM25 (granularity: articles, TTF). Nevertheless, this method was experimented later (cf. section 5). In order to take into account the non-overlapping constraint of the "Focused" task, the list of elements returned by our system were filtered by removing all the elements that overlap with another element which is better ranked. The best results (winner's iP[0.01], our best iP[0.01], MAiP, R[1500] and S[1500]) are emphasized with a bold font.

Table 6 Evaluation of 61 "Focused" task experiments

Run	Granularity	Tags	$iP[0.01]$	Rang	$MAiP$	Rang	R[1500]	S[1500]
FOERStep	Element	-	**0.6897**	1	0.2071	27	0.4494	1.11
Foc-1	Article	-	0.6412	13	**0.2791**	6	**0.7897**	5.57
Foc-2	Element	-	0.5688	37	0.1206	45	0.2775	**0.73**
Foc-3	Element	TTF	**0.6640**	7	0.2342	19	0.6110	3.34

The first experiment, *Foc-1*, in classical IR, ranked 13^{th} out of 61. The second experiment *Foc-2*, in Focused IR, did not do as well: 37^{th} out of 61. The early integration of the *Foc-3* tag-weights (TTF strategy), in Focused IR, gave very good results and ranked 7^{th} out of 61, which is better than for classical IR (*Foc-1*) and improves the Focused IR whose recall rates were weak (from 0.5688 to 0.6640 according to the $iP[0.01]$ criterion).

This result tends to confirm the advantage of Focused IR (*Foc-3*) when compared to classical IR (*Foc-1*). This also show the advantage of taking structural information into account (*Foc-2* vs. *Foc-3*). Furthermore, FOERStep produces better results with low recall rates (*i.e.* 0.01). However, the *Foc-1* experiment gives better results with recall rates at 0.05, and *Foc-1* and *Foc-3* give very good results in terms of recall: $MAiP$ at 0.2791 (resp. 0.2342) and $R[1500]$ at 0.7897 (resp. 0.6110).

We tested whether there is a significant difference between *Foc-1* and *Foc-3* using a paired t-test (one tailed) at 95 %. The performance (measured by *MAiP*) of *Foc-1* is significantly better than *Foc-3* but the size of the documents returned in *Foc-3* is significantly lower. Indeed, at 1,500 documents, the recall decreases to 16% between *Foc-1* and *Foc-3* whilst the size in MB of these documents decreases by 40%. This shows that the Focused IR eliminates a greater number of non-relevant elements. In other respects, the difference (measured by *iP[0.01]*) is not significant. As the size of the documents returned in Focused

IR (*Foc-3*) is lower, these experiments confirm the interest of our model in the context of mobile applications.

We also tested whether there is a significant difference between *Foc-2* and *Foc-3* and concluded that the performance (measured by *MAiP* and *iP[0.01]*) of *Foc-3* is better than *Foc-2*. Consequently, we arrived, with a different collection, at the same conclusion as Robertson *et al.* [33]: taking tags into account in the BM25 weighting scheme is interesting if this is done early (TTF strategy, *Foc-3* run), thereby maintaining the BM25's non-linearity, rather than taking the tags into account later, directly on the final term-weights [11]. This point was confirmed by other experiments in which the pre-impact strategy (TTF) and the post-impact strategy (CLAW) are compared. These experiments are presented in the next section.

5 Posterior analysis

The performances of the weighting function BM25 depend a lot on the tuning of its parameters, especially those related to the documents length normalization and to the tf saturation (b and k_1). Subsequently, we conducted in-depth experiments to study the impact of certain parameters on the model as exhaustively as possible and to analyse its behaviour when the parameters are finely tuned. We therefore carried out several experiments using six models: articles, articles + CLAW, articles + TTF, elements, elements + CLAW, elements + TTF.

Some parameters were setted after a few initial experiments (cf. section 3.4), and two important parameters were studied more thoroughly so that we might understand their influence on focused IR and to study the stability of our model. We have used a 2D grid for the parameters b (varying from 0 to 1, with 0.1 steps) and k_1 (varying from 0.2 to 3.8 with 0.2 graduations), thus a total of $6 * 11 * 19 = 1,254$ experiments.

For these posterior analyses, a different queries set was used during the learning stage (INEX 2006) than during the IR stage (INEX 2008). There is a risk of over-fitting in these "Posterior analysis" experiments, due to the tuning of the parameters using the 2008 INEX collection, which we also used to evaluate our model. This is a well-known problem in the context of IR competitions (TREC, INEX, etc.) [33, 42]. We believe however, just as Robertson *et al.* [33], that it is relevant to proceed like this. Indeed we aim to determine the potential of our model by seeking the best b and k_1 parameters, bearing in mind that, in real-life conditions, we will need to discover the values for b and k_1 using a learning collection.

5.1 Summary of the results

In table 7, the results obtained with the optimal parameter configuration according to the $iP[0.01]$ criterion are presented, while they are presented according to the $MAiP$ criterion in table 8. Tables 7 and 8 present the mean of the evaluation measures on the set of queries.

In order to situate the quality of the runs which were studied, we must add that the R7 run would have been ranked 4^{th} in the 2008 INEX in terms of $MAiP$ (the winner having achieved 0.3065) while the R6 run would have been ranked 4^{th} in terms of $iP[0.01]$ (the winner having achieved 0.6897).

Figure 4 presents the recall / precision curves of the 4 experiments with the best results according to the $iP[0.01]$ criterion ($R1, R3, R4, R6$ runs), excluding the CLAW runs ($R2, R5$) which are outperformed by TTF ($R3, R6$). This figure shows that the elements runs

Table 7 Evaluation of 1,254 runs with the $iP[0.01]$ criterion

Run	Granularity	Tags	b	k_1	$iP[0.01]$	#doc	#art	R[1500]	S[1500]
R1	Articles	-	0.4	1.6	0.6587	1,457	1,457	**0.8422**	8.22 MB
R2	Articles	CLAW	1.0	3.8	0.6278	1,457	1,457	0.7424	4.26 MB
R3	Articles	TTF	0.6	1.6	0.6654	1,457	1,457	0.8214	7.69 MB
R4	Elements	-	0.5	0.8	0.6738	1,463	1,257	0.4134	**1.65 MB**
R5	Elements	CLAW	0.2	3	0.6061	1,461	1,280	0.5730	2.83 MB
R6	Elements	TTF	0.3	0.8	**0.6837**	1,461	1,294	0.5180	2.98 MB

Table 8 Evaluation of 1,254 runs with the $MAiP$ criterion

Run	Granularity	Tags	b	k_1	$MAiP$	#doc	#art	R[1500]	S[1500]
R7	Articles	-	0.6	2.2	**0.2910**	1,457	1,457	0.8216	6.15 MB
R8	Articles	CLAW	0.8	2.4	0.2522	1,457	1,457	0.8004	6.24 MB
R9	Articles	TTF	0.6	2.6	0.2860	1,457	1,457	**0.8299**	7.09 MB
R10	Elements	-	0.1	2.2	0.2664	1,459	1,408	0.7476	5.24 MB
R11	Elements	CLAW	0.1	3.8	0.2137	1,459	1,356	0.6985	**5.00 MB**
R12	Elements	TTF	0.1	2.8	0.2576	1,459	1,389	0.7285	5.37 MB

(R4 and R6) seem to be outperformed by the article runs (R1 and R3). However, this is not true at low recall rates (recall $<= 0.05$). This is very interesting, as these precision-oriented rates are the most important in focused IR[9].

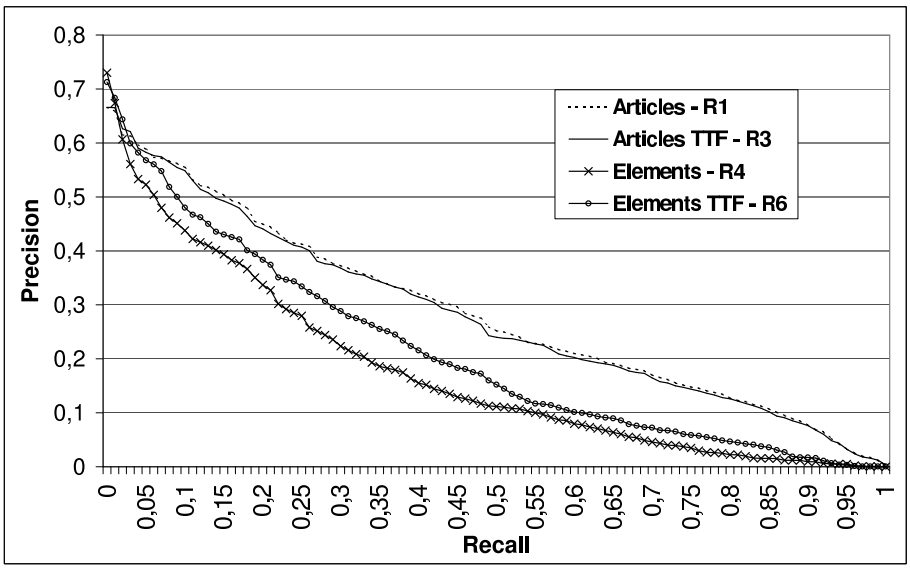

Fig. 4 Recall / Precision of the runs with the best results.

[9] The the main INEX 2008 measure is $iP[0.01]$

5.2 On the impact of b and k_1 parameters

Let us now study the influence of b and k_1 parameters on the results:

The b parameter: The role of b is to control the document length normalization (cf. equation 2). 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 [10].

The k_1 parameter: The role of k_1 is to control the term frequency saturation rate, which is very important for TTF strategy, as TTF modifies directly the tf.

The CLAW strategy does not perform well (either in the competition framework or in posterior analysis). This is due to the later integration of the tags weights. Indeed, as explained previously, in the CLAW strategy the tags are introduced directly in the final term-weights, while they are introduced early in the weighting scheme in the TTF strategy. The pre-impact (TTF-strategy) permits to maintain the non-linearity of the BM25 function and thus to provide better results.

5.2.1 Classical IR

Figure 5 presents the values of MAiP and $iP[0.01]$ according to b on the left (resp. k_1, on the right) in the context of classical IR. The $iP[0.01]$ and $MAiP$ measures presented for b (resp. k_1) are the ones obtained using the optimal value for k_1 (resp. b).

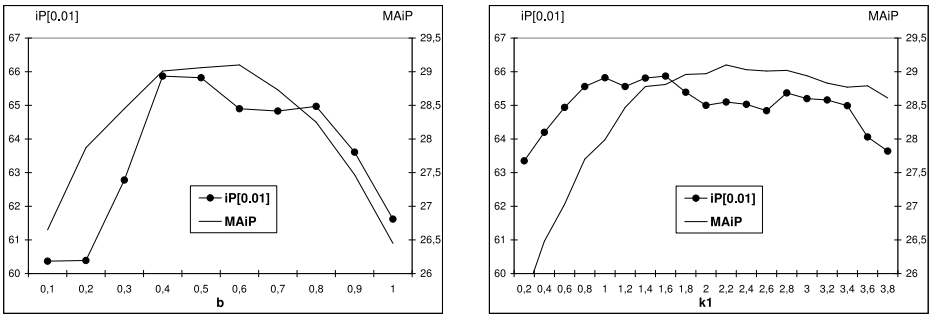

Fig. 5 Classical IR according to b and k_1

The best (b, k_1) values 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 proposed in the literature (*e.g.* (0.7, 1.2): for such values, the system performs an $iP[0.01]$ of 0.6352).

5.2.2 Focused IR

Figure 6 presents the behaviour of the BM25 model in focused IR.

The best (b, k_1) values are quite 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 seems to be counterproductive for optimizing recall in

[10] In our experiments, we set the minimum element length at 10 words and the largest article contains 35,000 words

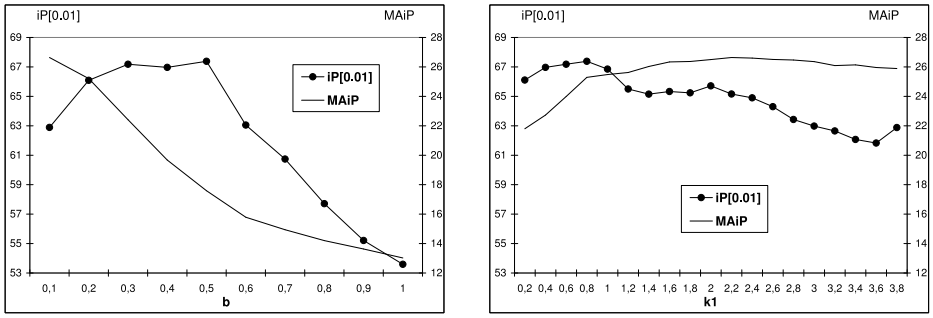

Fig. 6 Focused IR according to b and k_1

focused IR. On the other hand, it seems useful to optimize precision (best value: $b = 0.5$). The k_1 (tf saturation) seems to be less important for focused IR: iP[0.01] and MAiP fluctuates slightly with k_1.

5.2.3 Focused IR and BM25t (TTF strategy)

Figure 7 presents the behaviour of the BM25t model (TTF strategy) in focused IR.

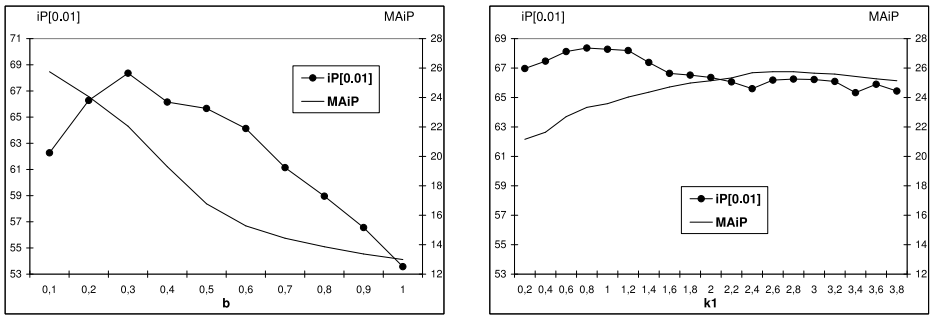

Fig. 7 Focused IR + TTF according to b and k_1

Again, the best (b, k_1) values are different for MAiP ($(b, k_1) = (0.1, 2.8)$) than for iP[0.01] ($(b, k_1) = (0.3, 0.8)$). As in the case of focused IR without TTF strategy, the best MAiP is reached with the minimum value of $b = 0.1$. The behaviour of focused IR is similar with and without TTF strategy.

6 Related works

The relevance of a document for a given query depends generally on the weight of the words in the query. This weight itself depends upon its frequency within the document and within the collection. When compared to this model, a weighting scheme permits the structure be taking into account in assigning a weight to the tags relative to their importance. This weight is then combined with those of the terms in order to determine the relevance of a document.

Thus, the relevance of a document does not only depend upon the frequency with which the terms in the query occur in the document, but it also depends upon their position within that document and these positions are defined by the tags with which they have been marked.

This principle has already been widely studied in the context of classic IR [25]. The tags which are considered such as their weights, may be chosen empirically. For example in [32], the tag *title* has a weight of 2 and the tag abstract is set to 1.5. One obvious limit to this approach lies in the difficulty in chosing the values for the weights.

For this reason, another way consists in learning automatically the weights assigned to the tags by using for instance genetic algorithms [38, 43] or by optimizing techniques based upon simulated annealing [2]. However, if the weights are not fixed by the user, again the set of tags is chosen empirically. Moreover, in these previous works, due to the computational cost, the number of tags used is very limited. In our model, the tags as well as the weights are determined automatically.

Once the weights of the tags have been determined, they should be combined with those of the words.

When logical tags are used their weight may be integrated ad hoc. In this case, the document may be divided into as many elements as it has parts (heading, abstract, main text, etc.) defined by these tags, and each part may be processed independently. Then, a linear combination of the scores obtained for each part can be computed. However, in the case of the BM25 model [15, 16], Robertson *et al.* demonstrated the advantages of duplicating the parts as many times as suggested by the weights [33]. For instance, a structured document with a title-weight equal to two becomes an unstructured document with the content of the *title* repeated twice. The unstructured document obtained is then processed in the usual way. The experimental evaluation carried out by Robertson *et al.* confirms that this approach (named BM25F weighting scheme) provides better results than a linear combination of scores computed for each part [33], the advantage being that it retains the non-linearity of the BM25 weighting scheme. However, in the studies mentioned previously, the system returns complete documents and its ability to extract parts of a document has not been evaluated. This is the aim of our work. Indeed, in this article, we propose a framework for focused XML retrieval, which becomes very important with the development of mobile applications and access.

In the context of focused IR, the weighting schemes have also been studied. Once the tag-weights have been fixed, a simple strategy that can be used in order to integrate them is based on a scalar product: the tag-weights are used as multiplying factors for the term weights within their scope. This approach was used to improve the probabilistic model [28, 48] as well as the vector space model [47]. However, in these works the tag-weights were arbitrarily chosen.

Other studies aim to exploit the XML tree-representation of documents [24, 37, 43]. Each XML element, corresponding to a tree-node, is characterized by a path leading from the tree-root to this node. The structure is taken into account at the term-level by considering the path of the element it contains. With this representation, Schlieder *et al.* introduced an extended vector model [37]. They computed the inverse document frequency for a term of each type of logical document found within the collection. Their system however, requires the user to give a structured query; which is not always possible. Kotsakis associates a weight directly with each path [24]. For example, a term located on the path *journal/issue/article/title* has a larger weight than a term on the path *journal/ issue/article/abstract*. The final weight of a term is made up of two components. The first one is computed with the classical *tf.idf* formula, while the second one is the weight associated with the position of the word in the tree (*i.e.* to the path of this node). The question of how these structural weights are computed

is not discussed in [24], whereas Trotman suggests estimating the weight of each XML node using a genetic algorithm [43]. This weight is then combined with the tf in various weighting schemes. These experiments showed some improvements when using vector space model or probability model, but no significant improvement was observed upon the results provided by BM25.

On the other hand, BM25E, which was introduced by Lu *et al.* gave promising experimental results within a focused IR framework [28]. It is probably the model closest to the one which we are proposing in that it assigns a score to an element by affecting an early combination of the weights given to the terms in the query and those of their tags. However, in BM25E, the tag-weights are determined empirically. Furthermore, as with the majority of approaches previously cited, very few tags are taken into consideration (generally less than 5) and their choice often requires manual handling. Also, as the authors of this model pointed out *"the creation of a practical algorithm to generate values for tuning parameters at the element level is a challenging task"*. This article tries to provide parts of an answer to this question. Unfortunately, it was not possible to compare our model, which can handle several hundred tags, against BM25E, which is "challenging" to implement in such case as mentioned by its authors [28].

7 Conclusion and perspectives

In this article, we have presented a new approach for taking into account the XML structure for focused IR. This approach is inspired by probabilistic models of IR. For this purpose we propose to look at both the logical structure and all the other structures. Logical structure is used during the indexing stage in order to define the type of elements which are indexed and potentially returned by the system. The logical structure and the other structures are then integrated into the document model. During the learning stage, a weight is calculated for each tag, based upon the probability that the tags will be able to distinguish between relevant and non-relevant terms. During the query stage, calculating the relevance of an XML element for a query combines the textual content (terms) with the structure (tags which label terms).

The main contribution of this paper is a modelling of the tags ability to highlight terms according to the principles of the probabilistic IR model. Thus, the tag weights are automatically adjusted. Because the late integration of tag weights into the term weighting function only showed a slight improvement in the results (CLAW strategy), we proposed an early integration (TTF strategy) which presents the advantage to maintain the non-linearity of the BM25 function and produces much better results.

The second contribution of this work is an extensive experimentation of the BM25 model [15, 16] in the context of XML retrieval. We evaluated our model through participation in the INEX competition. Our first experiment in classical IR (Foc-1 corresponding to a granularity of articles) came 13^{th} out of 61. Our second experiment (Foc-2 corresponding to a granularity of XML elements) achieved a lower ranking: 37^{th} out of 61. The early integration of tags in the model for focused IR (Foc-3) achieved very good results, very close to those of the best INEX systems, thereby demonstrating the advantages of focused IR (Foc-3) when compared with classical IR (Foc-1), or with structural information retrieval (Foc-2). We think that it could be possible to improve Foc-3 results in order to reach those of the best INEX systems, for example by using some query processing heuristics as some of INEX systems do.

Even if the collections which were used are very different, we have reached the same conclusions as Robertson *et al.* [33]: it is worthwhile taking the tags in the BM25 weighting function [15, 16] into account, if this is done early on. But, in the context of XML retrieval, the number of tags is very large. So it is not possible to optimize dedicated parameters b and k_1 for each tag, as is done by BM25f [33] for each field. Nevertheless, we hypothesize that the tag weights used by TTF strategy can also somehow replace this fine-tuning: indeed, tag weights have an influence on the tf_{ji} just like the k_1 parameter.

The last contribution of this article is a quite exhaustive study of the influence of the BM25's parameters b and k_1 on the $iP[0.01]$ measure and on the $MAiP$. The first result provided by this study is the smooth variation of the model quality with respect to parameter changes. This is important as it shows that few experiments are useful to set up the model parameters correctly. Moreover, we can expect good behaviour in generalization. This explains the good results obtained by the system during the INEX competition. The tuning of the system done on a collection leads to good parameters to analyse a new set of queries. With the $MAiP$ as the evaluation measure, the best model is the classical BM25 model. This can probably be explained by the fact that the measures based on recall favour systems which return large elements (granularity article). Finally, the best performances observed at low recall points, are achieved by the tag-enriched model BM25t which returns elements.

Several perspectives remain open.

First of all, the TTF strategy implements a simple average of the tag-weights. Previous experiments, which are not reported in this article, showed that this method achieves better results than other functions (multiplication of weights, taking only the closest tag into account, etc.). This point should be analyzed theoretically. The average used in these experiments places all of the tags attached to a given term on the same level. A non-uniform weighting of tag-weights, for instance, according to the distance between the term and the tag, may prove more efficient.

From the experimental point of view, we focused our attention on parameters b and k_1 of the BM25 model. Other parameters also need to be studied. In particular, the way the df is computed may have great importance. Further, work is still needed in order to properly take into account the great variation of documents length in the context of focused IR. This can be done either by considering some normalization procedures as done in [18], or by a better computation of the parameter df.

Our model has been evaluated using the INEX 2008 collection (cf. section 4), and a posterior analysis is presented but with an overfitting risk. Our model should be evaluated on other collections. The INEX collection, composed by Wikipedia XML documents, is strongly structured. We hope that our model could perform well on any collection of structured documents, as for example the INEX 2010 collection, using the INEX 2009 set of queries as a training collection. This collection is also strongly structured, even more than the INEX 2008 collection, indeed a new kind of structure is introduced by the use of semantic tags. We also hope that our model could perform well on any Web collection, as for example the TREC Web collection, composed by HTML documents.

8 Acknowledgements

The authors would like to thank Franck Thollard who greatly contributed to this work. The work was supported by the Web Intelligence project (région Rhône-Alpes, cf. `http://www.web-intelligence-rhone-alpes.org`).

References

1. Baeza-Yates R, Fuhr N, Maarek Y (2006) Introduction to the special issue on XML retrieval. ACM Transactions on Information Systems 24:405–406
2. Boyan J, Freitag D, Joachims T (1996) A machine learning architecture for optimizing web search engines. In: AAAI Workshop on Internet-based Information Systems
3. Carmel D, Maarek Y, Soffer A (2001) XML and information retrieval: a SIGIR 2000 workshop. SIGMOD Record 30:62–65
4. Denoyer L, Gallinari P (2006) The Wikipedia XML corpus. SIGIR Forum 40:64–69
5. Fuhr N, Großjohann K (2001) XIRQL: a query language for information retrieval in XML documents. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, SIGIR'01, pp 172–180
6. Fuhr N, Lalmas M, Malik S (eds) (2004) Proceedings of the Second Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2003, Schloss Dagstuhl, Germany, December 15-17, 2003
7. Fuhr N, Lalmas M, Malik S, Szlávik Z (eds) (2005) Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Lecture Notes in Computer Science, vol 3493, Springer
8. Fuhr N, Kamps J, Lalmas M, Malik S, Trotman A (2008) Overview of the INEX 2007 Ad Hoc Track. In: [9], pp 1–23
9. Fuhr N, Kamps J, Lalmas M, Trotman A (eds) (2008) Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007, Lecture Notes in Computer Science, vol 4862, Springer
10. Fuller M, Mackie E, Sacks-Davis R, Wilkinson R (1993) Coherent answers for a large structured document collection. In: Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA, USA, pp 204–213
11. Géry M, Largeron C, Thollard F (2008) Integrating structure in the probabilistic model for information retrieval. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, Washington, DC, USA, pp 763–769
12. Géry M, Largeron C, Thollard F (2009) UJM at INEX 2008: Pre-impacting of Tags Weights. In: [13], pp 46–53
13. Geva S, Kamps J, Trotman A (eds) (2009) Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008, Lecture Notes in Computer Science, vol 5631, Springer
14. Itakura KY, Clarke CLA (2009) University of Waterloo at INEX 2008: Adhoc, Book, and Link-the-Wiki tracks. In: [13], pp 132–139
15. Jones KS, Walker S, Robertson SE (2000) A probabilistic model of information retrieval: development and comparative experiments - part 1. Information Processing and Management 36:779–808
16. Jones KS, Walker S, Robertson SE (2000) A probabilistic model of information retrieval: development and comparative experiments - part 2. Information Processing and Management 36:809–840

17. Kamps J, Marx M, de Rijke M, Sigurbjörnsson B (2005) Structured queries in XML retrieval. In: Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, CIKM'05, pp 4–11
18. Kamps J, Rijke MD, Sigurbjörnsson B (2005) The importance of length normalization for XML retrieval. Information Retrieval 8:631–654
19. Kamps J, Geva S, Trotman A (2008) Report on the SIGIR 2008 workshop on focused retrieval. SIGIR Forum 42:59–65
20. Kamps J, Pehcevski J, Kazai G, Lalmas M, Robertson S (2008) INEX 2007 Evaluation Measures. In: [9], pp 24–33
21. Kamps J, Geva S, Trotman A, Woodley A, Koolen M (2009) Overview of the INEX 2008 Ad Hoc track. In: [13], pp 1–28
22. Kazai G, Trotman A (2007) Users' perspectives on the usefulness of structure for XML information retrieval. In: Proceedings of the 1st International Conference on the Theory of Information Retrieval, pp 247–260
23. Konopnicki D, Shmueli O (1995) W3QS: A query system for the world-wide web. In: Proceedings of the 21th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers, San Francisco, CA, USA, VLDB'95, pp 54–65
24. Kotsakis E (2002) Structured information retrieval in XML documents. In: Proceedings of the 2002 ACM symposium on Applied computing, ACM, New York, NY, USA, SAC'02, pp 663–667
25. Lalmas M (2009) Structure weight. In: Liu L, Özsu MT (eds) Encyclopedia of Database Systems, Springer, p 2862
26. Lalmas M (2009) XML Retrieval. Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers
27. Lalmas M, Trotman A (2009) XML retrieval. In: Liu L, Özsu MT (eds) Encyclopedia of Database Systems, Springer US, pp 3616–3621
28. Lu W, Robertson SE, MacFarlane A (2006) Field-Weighted XML Retrieval Based on BM25. In: Fuhr N, Lalmas M, Malik S, Kazai G (eds) Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, Springer, Dagstuhl Castle, Germany, Lecture Notes in Computer Science, vol 3977, pp 161–171
29. Maron ME, Kuhns JL (1960) On relevance, probabilistic indexing and information retrieval. Journal of the ACM 7:216–244
30. Mass Y, Mandelbrod M (2004) Retrieving the most relevant XML components. In: [6], pp 53–58
31. O'Keefe RA, Trotman A (2004) The simplest query language that could possibly work. In: [6], pp 167–174
32. Rapela J (2001) Automatically combining ranking heuristics for HTML documents. In: Proceedings of the 3rd international workshop on Web information and data management, ACM, New York, NY, USA, WIDM'01, pp 61–67
33. Robertson S, Zaragoza H, Taylor M (2004) Simple BM25 extension to multiple weighted fields. In: Proceedings of the 13th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, CIKM'04, pp 42–49
34. Robertson SE, Jones KS (1976) Relevance weighting of search terms. Journal of the American Society for Information Science 27(3):129–146
35. Robertson SE, Walker S, Jones S, Hancock-Beaulieu M, Gatford M (1995) Okapi at TREC-3. In: Harman DK (ed) Proceedings of the third Text REtrieval Conference (TREC-3), pp 109–126

36. Salton G, McGill MJ (1986) Introduction to modern Information Retrieval. McGraw-Hill, New York, NY, USA
37. Schlieder T, Meuss H (2002) Querying and ranking XML documents. Journal of the American Society for Information Science and Technology 53:489–503
38. Sun YHK, Kim S, hong Eom J, tak Zhang B (2000) SCAI experiments on TREC-9. In: Proceedings of the 9th Text REtrieval Conference (TREC-9), pp 392–399
39. Swets JA (1963) Information retrieval systems. Science 141:245–250
40. Taha K, Elmasri R (2010) BusSEngine: a business search engine. Knowledge and Information Systems 23(2):153–197
41. Tamine-Lechani L, Boughanem M, Daoud M (2010) Evaluation of contextual information retrieval effectiveness: overview of issues and research. Knowledge and Information Systems 24(1):1–34
42. Taylor M, Zaragoza H, Craswell N, Robertson S, Burges C (2006) Optimisation methods for ranking functions with multiple parameters. In: Proceedings of the 15th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, pp 585–593
43. Trotman A (2005) Choosing document structure weights. Information Processing and Management 41(2):243–264
44. Trotman A, Sigurbjörnsson B (2005) Narrowed Extended XPath I (NEXI). In: [7], pp 16–40
45. Trotman A, Sigurbjörnsson B (2005) NEXI, now and next. In: [7], pp 41–53
46. Trotman A, Geva S, Kamps J (2007) Report on the SIGIR 2007 workshop on focused retrieval. SIGIR Forum 41(2):97–103
47. Wilkinson R (1994) Effective retrieval of structured documents. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag, New York, NY, USA, SIGIR'94, pp 311–317
48. Wolff JE, Flörke H, Cremers AB (2000) Searching and browsing collections of structural information. In: Proceedings of the IEEE Advances in Digital Libraries 2000, IEEE Computer Society, Washington, DC, USA, pp 141–150
49. Zaragoza H, Craswell N, Taylor M, Saria S, Robertson S (2004) Microsoft cambridge at TREC 13: Web and hard track. In: Voorhees EM, Buckland LP (eds) Proceedings of the 13th Text REtrieval Conference (TREC 2004)
50. Zhu J, Huang X, Song D, Rüger SM (2010) Integrating multiple document features in language models for expert finding. Knowledge and Information Systems 23(1):29–54
51. van Zwol R, Baas J, van Oostendorp H, Wiering F (2006) Bricks: The building blocks to tackle query formulation in structured document retrieval. In: Lalmas M, MacFarlane A, Rüger SM, Tombros A, Tsikrika T, Yavlinsky A (eds) Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, Springer, Lecture Notes in Computer Science, vol 3936, pp 314–325

Author Biographies

Mathias Géry is Assistant Professor at the Computer Science Department of the University of Saint-Etienne (France) and the leader of the Data Mining and Information Retrieval group at the Hubert Curien Laboratory. He received his Ph.D. degree in Computer Science from the University of Grenoble I (France) in 2002. His current research focuses on Structured Information Retrieval, Social Information Retrieval and Social Networks Mining.

Christine Largeron is Professor at Jean Monnet University and she is member of the Machine Learning group at the Hubert Curien Laboratory. She received her Ph.D in computer science from Claude Bernard University (Lyon - France) in 1991 and then her HDR from Jean Monnet University in 2004. Her main interests include data mining and information retrieval and her current research focuses on developing methods to efficiently retrieve information in structured data such as XML documents or social networks. She has served as program committee member of international conferences and she has been invited as reviewers by several journals (Dawak, ASONAM, INEX, KAIS, Pattern Recognition Letters, WIAS).