XFIRM at INEX 2006. Ad-hoc, Relevance Feedback and MultiMedia tracks

Lobna Hlaoua, Mouna Torjmen, Karen Pinel-Sauvagnat, and Mohand Boughanem

SIG-RFI, IRIT, Toulouse, France

Abstract. This paper describes experiments carried out with the XFIRM system in the INEX 2006 framework. The XFIRM system uses a relevance propagation method to answer CO and CO+S queries. Runs were submitted to the ad-hoc, relevance feedback and multimedia tracks.

1 Introduction

In this paper, we decribe the IRIT/SIG-RFI participation in the INEX 2006 ad-hoc, relevance feedback and multimedia tracks. Our participation is based on the XFIRM system [10], which uses a relevance propagation method.
In the *ad-hoc track* (section 2), we propose to evaluate the algorithms proposed part years [10] on a new collection (the Wikipedia collection) and on new searching tasks (Best in Context and All in Context tasks). We also compare results obtained using structural hints of queries with results obtained with simple keywords terms.
For the *relevance feedback track* and for our second participation to the track (section 3), we propose to apply a Content-Oriented approach inspired from the probabilistic theory and to improve the Structure-Oriented approach presented in [10].
At last, for our first participation to the *multimedia track* (section 4), we describe a context-based approach for multimedia retrieval. This approach uses the text surrounding images and structure of documents to judge the relevance of images or XML components.

2 Ad-hoc track

The aim of our participation to the ad-hoc track is to evaluate on the wikipedia collection [2] the algorithms we proposed at INEX 2005 [10]. Results presented here are mainly official results and further experiments need to be done to confirm results obtained with this preliminary work.

2.1 The XFIRM model

The model is based on a relevance propagation method. During query processing, relevance scores are first computed at leaf nodes level. Then, inner nodes relevance is evaluated by doing a propagation of leaf nodes scores through the document tree. An ordered list of subtrees is then returned to the user.

Processing of Content-Only queries Let $q = t_1, \ldots, t_n$ be a content-only query. Relevance values are computed using the similarity function $RSV(q, ln)$.

$$RSV(q, ln) = \sum_{i=1}^{n} w_i^q * w_i^{ln}, \quad where \quad w_i^q = tf_i^q \quad and \quad w_i^{ln} = tf_i^{ln} * idf_i * ief_i \quad (1)$$

Where w_i^q and w_i^{ln} are the weights of term i in query q and leaf node ln respectively. tf_i^q and tf_i^{ln} are the frequency of i in q and ln respectively, $idf_i = log(|D|/(|di|+1))+1$, with $|D|$ the total number of documents in the collection, and $|di|$ the number of documents containing i, and ief_i is the inverse element frequency of term i, i.e. $log(|N|/|nf_i|+1)+1$, where $|nf_i|$ is the number of leaf nodes containing i and $|N|$ is the total number of leaf nodes in the collection.

Each node in the document tree is then assigned a relevance score which is function of the relevance scores of the leaf nodes it contains and of the relevance value of the whole document.

$$r_n = \rho * |L_n^r| . \sum_{ln_k \in L_n} \alpha^{dist(n, ln_k)-1} * RSV(q, ln_k) + (1-\rho) * r_{root} \qquad (2)$$

$dist(n, ln_k)$ is the distance between node n and leaf node ln_k in the document tree, i.e. the number of arcs that are necessary to join n and ln_k, and $\alpha \in]0..1]$ allows to adapt the importance of the $dist$ parameter. $|L_n^r|$ is the number of leaf nodes being descendant of n and having a non-zero relevance value (according to equation 1). $\rho \in]0..1]$, inspired from work presented in [6], allows the introduction of document relevance in inner nodes relevance evaluation, and r_{root} is the relevance score of the *root* element, i.e. the relevance score of the whole document, evaluated with equation 2 with $\rho = 1$.

Processing of CO+S queries The evaluation of a CO+S query is carried out with the following steps:

1. INEX (NEXI) queries are translated into XFIRM queries
2. XFIRM queries are decomposed into sub-queries SQ and elementary sub-queries ESQ, which are of the form: $ESQ = tg[q]$, where tg is a tag name, i.e. a structure constraint, and $q = t_1, ..., t_n$ is a content constraint composed of simple keywords terms.
3. Relevance values are then evaluated between leaf nodes and the content conditions of elementary sub-queries

4. Relevance values are propagated in the document tree to answer to the structure conditions of elementary sub-queries
5. Sub-queries are processed using the results of elementary sub-queries
6. Original queries are evaluated by doing an upwards and downwards propagation of the relevance weights

Step 3 is processed thanks to formula 1. In step 4, the relevance value r_n of a node n to an elementary subquery $ESQ = tg[q]$ is computed according the following formula:

$$r_n = \begin{cases} \sum_{ln_k \in L_n} \alpha^{dist(n,ln_k)-1} * RSV(q, ln_k) \; if \; n \; \in construct(tg) \\ 0 \; otherwise \end{cases} \qquad (3)$$

where the $construct(tg)$ function allows the creation of set composed of nodes having tg as tag name, and $RSV(q, ln_k)$ is evaluated during step 2 with formula 1. The $construct(tg)$ function uses a *Dictionary* Index, which provides for a given tag tg the tags that are considered as equivalent. This index is built manually.

More details about CO and CO+S queries processing can be found in [10].

2.2 Runs and results

For all retrieval strategies, runs were submitted using content-only conditions of topics (CO runs) and content and structure conditions (CO+S runs). Results are presented with the official metrics.

Thorough task. For the Thorough task, all nodes having a non-zero relevance value are returned by the XFIRM system. Results are described in Table 1.

Table 1. Thorough task. Generalised quantisation function with filtered assessments. Official runs are in bold.

Queries	Parameters	nxCG[5]	nxCG[10]	nxCG[25]	nxCG[50]	ep/gr MAP
CO	$\alpha = 0.6, \rho = 1,$ **eq. 2**	**0.2161**	**0.1822**	**0.1470**	**0.1235**	**0.01831**
CO	$\alpha = 0.6, \rho = 0.9,$ **eq. 2**	**0.2167**	**0.1885**	**0.1581**	**0.1526**	**0.0241**
CO	$\alpha = 0.1, \rho = 1,$ eq. 2	0.228	0.2002	0.1703	0.1367	0.0230
CO	$\alpha = 0.1, \rho = 0.9,$ eq. 2	0.2289	0.2005	0.1763	0.1447	0.0266
CO	$\alpha = 0.2, \rho = 1,$ eq. 2	0.2318	0.2117	0.1766	0.1458	0.02352
CO	$\alpha = 0.2, \rho = 0.9,$ eq. 2	0.2413	0.2178	0.1857	0.1555	0.02779
CO+S	$\alpha = 0.9,$ **eq. 3**					**0.0089**

2.3 Focussed task.

In order to reduce/remove nodes overlap, for each relevant path, we keep the most relevant node in the path. The results set is then parsed again, to eliminate any possible overlap among ideal components. Results are described in Table 2.

Table 2. Focussed task. Generalised quantisation function with filtered assessments. All runs are official.

Queries	Parameters	nxCG[5]	nxCG[10]	nxCG[25]	nxCG[50]
CO	$\alpha = 0.6, \rho = 1$, eq. 2	0.1333	0.1104	0.0853	0.0656
CO+S	$\alpha = 0.9$, eq. 3	0.1839	0.1456	0.1015	0.0879
CO+S	$\alpha = 0.8$, eq. 3	0.1820	0.147	0.1088	0.0906

2.4 All in Context task.

We use two different retrieval strategies:

1. Relevance values are computed for each node of the collection according to the focussed stategy and nodes are then ranked by article (we rank first the top ranked node according to equation 2 and then all the nodes having a non-zero relevance value belonging to the same document, and so on)
2. Nodes are first ranked by the relevance of the document they belong to (according the the Mercure system [1]), and then by their own relevance (according to equation 2 used in a focussed strategy).

Official results are described in Table 3, and additional results at element level are provided in Table 4.

Table 3. All in Context task. Generalised quantisation function. All runs are official.

Queries	Parameters	MAgP	gP[5]	gP[10]	gP[25]	gP[50]
CO	$\alpha = 0.6, \rho = 1$, eq. 2, 2nd strategy	0.0683	0.2136	0.1810	0.1359	0.0950
CO	$\alpha = 0.1$, eq. 3, 1st strategy	0.0388	0.1159	0.1042	0.0804	0.0633

Table 4. All in Context task. Element level. All runs are official.

Queries	Parameters	F-avg INTERSECTION	F-avg UNION
CO	$\alpha = 0.6, \rho = 1$, eq. 2, 2nd strategy	0.5050	0.1632
CO	$\alpha = 0.1$, eq. 3, 1st strategy	0.4066	0.1723
CO+S	$\alpha = 0.9$, eq. 3, 2nd strategy	0.4473	0.1632

Best in Context task. For the Best in Context task, nodes relevance is first computed in the same way than for the thorough strategy. We then only keep the most relevant node per article. Official results are described in table 5.

Table 5. Best in Context task. All runs are official.

Queries	Parameters	BEPD A=0.01	BEPD A=0.1	BEPD A=1	BEPD A=10	BEPD A=100
CO	$\alpha = 0.6, \rho = 1$, eq. 2	0.1082	0.1685	0.2505	0.3586	0.4466
CO	$\alpha = 0.9$, eq. 3	0.0723	0.1215	0.2037	0.3441	0.5086
Queries	Parameters	EPRUM A=0.01	EPRUM A=0.1	EPRUM A=1	EPRUM A=10	EPRUM A=100
CO	$\alpha = 0.6, \rho = 1$, eq. 2	0.0120	0.0269	0.0518	0.0843	0.1230
CO	$\alpha = 0.9$, eq. 3	0.0091	0.0177	0.0340	0.0661	0.1202

Discussion For all strategies, further experimentations are needed, in order to see whether conclusions drawn with the IEEE collection and past metrics are validated or not. However, we can already say by analysing additional results of the Thorough strategy that:

- document relevance seems to be important when evaluating nodes relevance,
- and very specific nodes are preferred by users,

this validates past years results.
The use of structural constraints in queries do not improve results, this contradicts results found in [9]. This can be explained by the small equivalencies dictionary used for tags: it should be extended before drawing conclusions.
Results obtained with the Focussed strategy and Best In Context tasks are not as good as expected and parameters tuning need to be done.
At last, although our system was ranked first at element level, we were disappointed with the results obtained using the official metrics for the All In Context task. We need to further investigate this matter.

3 Relevance feedback track

In our previous works, we proposed two main approaches for relevance feedback:

- a content-oriented approach, which expands the query by adding keywords terms,
- a structure-oriented approach, which expands the query by adding structural conditions.

For the 2006 RF track, we applied a Content-Oriented approach inspired from the probabilistic theory and to improve our Structure-Oriented approach already presented in [10].

3.1 Content-Oriented Relevance Feedback

Probabilistic method
Simple term extraction (without query terms re-weighting) using the Rocchio's

algorithm [7] has already been explored and did not show any improvement [10]. The main question is still how to extract and to weight the best terms that will be added to the query. Our approach in this paper is inspired from another very known way to do RF, the probabilistic model [3].
Let us consider E^r as a set of relevant elements; we define the probability of term expressiveness by $P(t_i/E^r)$. We estimate this probability according to a simplified version of the Robertson's formula [3]:

$$P(t_i/E^r) = r_e/R_e \tag{4}$$

where r_e is the number of elements in E^r containing term t_i and $R_e(= ||E^r||)$ is the number of relevant elements.
The new query is finally composed of terms ranked in the top k according to the above formula, that are added to the original query terms.
If original query terms appear in the top k, we do not add them again.

Re-weighting keywords using the probabilistic method
In previous work, we considered term relevance as a binary measure (relevant/not relevant). No preference between terms could be expressed and the user need was not really refined. Therefore, we propose here to add weights to query terms, that represent their degree of importance. Weight values vary in]0,1]. We use the scores calculated according to the probabilistic method described above. The higher weight is assigned to original query terms (weight=1).
For example, let $Q = t_1, ..., t_n$ be the initial query. If we choose to add 3 relevant terms to the initial query, the new query will be:
Q'=$(t_1, 1), ..., (t_n, 1), (t_o, w_o), (t_p, w_p), (t_r, w_r)$ where t_o, t_p and t_r are the added terms and w_o, w_p and w_r their associated weights.

3.2 Combined approach: content-and-structure RF

In the combined approach, both structural constraints and keywords terms are added to the initial query, i.e. we combine the content-oriented and the structure-oriented approaches. The principle of the structure-oriented approach is reminded in the following paragraph.

Structure-oriented approach
Our hypothesis in the structure-oriented approach is that for a given query, the user may only be interested to some types of elements (like for example *section*, *article*, *images*,...). Our approach consists in refining the initial query by adding some structures, extracted from the set of judged elements that could contain the information needed by the user. The idea behind structure-oriented RF is therefore to find for each query the appropriate generic structures, called here the generative structures, shared by the greatest amount of relevant elements. The generative structures are extracted as follows. Let:
- e_i be an element $\in E^r$; e_i is characterized by a path p_i and a score w_i initialized to 1 at the beginning of the algorithm. p_i is only composed of tag names, like

the following example: */article/bdy/sec*.
- CS be a set of Common Structures, obtained as algorithm output.
For each $(e_i, e_j) \in E^r \times E^r, i \neq j$, we apply the SCA algorithm, which allows the identification of the smallest common ancestor of e_i and e_j. The path of this smallest common ancestor is then added to the set of common structures CS. The SCA algorithm is processed for each pair of E^r elements, and is described below.

```
SCA(e_i,e_j) return boolean
Begin
if p_i.first=p_j.first then
   if p_i.last=p_j.last then
         if ∃e_p(p_p, w_p) ∈ CS/p_p = p_i then
                  w_p ← w_p + w_j
                  return true
         else CS ← e_j(p_j, w_j + w_i)
                  return true
   else
         if head(p_j) ≠ null then
                  p'_j ← head(p_j)
                  w'_j ← w_j/2
                  SCA (e_i(p_i, w_i), e'_j(p_j, w_j))
         else SCA(e_j, e_i)
return false
End
```

$p.last$ and $p.first$ are respectively the last and the first tag of the path p and $head(p)$ is a function allowing to reduce the path p, by removing the last tag of the path. For example, *head(/article/bdy/section)=/article/bdy*.
In our approach, we are only interested in element tags and we choose to only compare the $p.last$ tags of paths (even if elements having the same $p.last$ can have different paths). When no result is added in the CS set by the $SCA(e_i, e_j)$ algorithm, we try to run $SCA(e_j, e_i)$.
In order to express the new (CO+S) query, we then extract the k top ranked structures according to their score w_i in the CS set. The selected structures are then used in their simple form (i.e. the last tag).
Let $Q1 = t_1, t_2, ...t_n$ be a query composed of n keywords (CO query) and $S1$, $S2$ be two generative structures of the set of Common Structures CS. The new query derived from Q1 using our structure-oriented relevance feedback method will be: $Q1' = S1[t_1, t_2, ...t_n]\ OR\ S2[t_1, t_2, ...t_n]$.

Combined approach
As explained before, the combined approach consists in adding both content and structure constraints to the initial query. The new query (that will be a CO+S query), is thus composed of the most appropriate generic structures and of the k

best terms according to formula 4. Terms are added to the original query terms with their associated weights.

Example
Let $Q1 = t_1, t_2, ..., t_n$ be a query composed of n keywords (CO query) and $S1$ and $S2$ be two generative structures extracted from the set of Common Structures. The new query derived from $Q1$ using our combined Relevance Feedback method will be (we choose to add for example 2 generative structures and 2 relevant terms t_o, and t_p): $Q1' = S1[(t_1, 1), ..., (t_n, 1), (t_o, w_o), (t_p, w_p)]$ OR $S2$ $[(t_1, 1), ..., (t_n, 1), (t_o, w_o), (t_p, w_p)]$

3.3 Runs

CO.thorough task
For official runs and according to previous experiments, we use

- the Content-Oriented RF approach by adding 3 or 10 relevant terms to the initial query, respectively named in Table 6 by *CO-C3* and *CO-C10*.
- the combined approach with 3 expressive terms and 3 generative structures selected according to the SCA algorithm, and named by *CO-C3S3* in Table6.

The top 20 elements of each query is used to select relevant terms / relevant structures. We use the base run of CO queries obtained with XFIRM using $\alpha = 0.6$ and $\rho = 1$.
We present in the following table the Absolute Improvement (AI) and the Relative Improvements (RI) according the MAep metric. In Relevance Feedback track, all evaluations are preliminary; official evaluations are not yet available.

Table 6. Impact of Content and Combines-Oriented RF in CO queries

	CO-C3	CO-C10	CO-C3S3
AI(MAep)	-0.00645	-0.00653	-0.00332
RI-(MAep)	-64.67%	-65.48 %	-33.34%

We notice in Table 6 that there is no improvement (negative values of RI). When comparing the columns we can see that the number of added expressive terms does not have a significant impact (CO-C3 and CO-C10). This can be explained by the fact that the suitable number of terms to be added depends on the query length [8].
Non-standard methods to select elements used to choose relevant terms and relevant structures are also tested. We propose to evaluate the impact of the number of judged elements used to extend queries, by using elements in the top 10 and top 40 designated respectively in Table 7 by *CO-J10* and *CO-J40*. We also propose to consider a fixed number of relevant elements to extend queries (4 strictly

relevant elements are used in this paper) designated by *CO-R4*. We use the base run of CO queries using $\alpha = 0.6$ $\rho = 0.9$. We apply the Content-Oriented RF with adding 3 expressive terms. According to the Table 7, we do not see any

Table 7. Impact of RF strategies in CO queries

	CO-J10	CO-J40	CO-R4
AI(MAep)	-0.0087	-0.00734	-0.00722
RI-(MAep)	-0.5363%	-48.74%	-47.98%

improvement for all runs. However, we still think that it is important to choose appropriate strategies of Relevance Feedback. Indeed, we obtained contradictory results in previous experiments, where a fixed number of relevant elements improved the effectiveness of the Retrieval system [5]. For example, on the INEX 2005 collection, the Relative Improvement for the MAep strict metric was about 25%, whereas no improvement can be observed with the traditional strategy.

CO+S thorough task
For this task we evaluated the content-oriented RF using 3 and 5 terms and the combined approach using 3 expressive terms and 1 generative structure respectively designated by *COS-C3*, *COS-C5* and *COS-C3S* in Table 8. We used the base run of CO+S queries with $\alpha = 0.9$ In this task, we notice that the two ap-

Table 8. Impact of Content and Combines-Oriented RF in CO+S queries

	COS-C3	COS-C5	COS-C3S
AI(MAep)	0.0008	0.00118	0.00205
RI-(MAep)	18.27%	24.42%	42.36%

proaches of Relevance feedback are efficient and the improvement is about 48% when we apply the Combined approach. Moreover, we notice that the addition of 5 terms is more efficient than the addition of 3 terms to initial query.

4 Multimedia track

Two types of topics are explored in the INEX Multimedia Track: MMFragments and MMImages.
A MMFragments topic is a request which objective is to find relevant XML fragments given a multimedia information need. Here, the topic asks for multimedia fragments,*i.e* fragments composed of text and /or images.
A MMImages topic is a request which objective is to find relevant images given

an information need. Here, the type of the target element is defined as an 'image'. This is basically image retrieval, rather than XML element retrieval. Some topics use image as query: the user indicates by this way that results should have images similar to the given example.

In Image Retrieval, there are two main approaches [11] : (1) Context Based Image Retrieval and (2) Content Based Image Retrieval:

1. The context of an image is all information about the image issued from other sources than the image itself. At the moment, only the textual information is used as context. The main problem of this approach is that documents can use different words to describe the same image or can use the same words to describe different concepts.
2. Content Based Image Retrieval (CBIR) systems use low-level image features to return images similar to an image used as example. The main problem of this approach is that visual similarity does not correspond to semantic similarity (for example a CBIR system can return a picture of blue sky when the example image is a blue car).

The work presented in this paper belongs to the first approach.

4.1 Runs

MMImages topics processing (MMI method) : Our method uses the text surrounding images and structure of document to judge the relevance of images. A first step is to search relevant nodes according to the XFIRM Content Only method. Then, we only use documents having a score > 0 and we reduce our retrieval domain to both relevant nodes and images nodes belonging to relevant documents. For each image, we use the closest nodes to judge its relevance. The used nodes are: the descendant nodes, the ancestor nodes and the brother nodes (Figure 1).

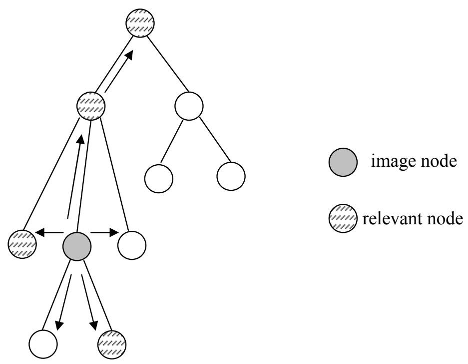

Fig. 1. Use of ancestor, brother and descendant nodes to evaluate images relevance

An image score corresponding to each of the preceding sources of evidence is computed:

- W_d^{im} is the image score computed using descendant nodes,
- W_b^{im} is the image score computed using brother nodes,
- W_a^{im} is the image score computed using ancestor nodes,

The total image score is then expressed as follows:

$$W_{im} = p_1.W_d^{im} + p_2.W_b^{im} + p_3.W_a^{im} \tag{5}$$

where p_1, p_2 and p_3 are parameters used to emphasize some weights types and $p_1 + p_2 + p_3 = 1$.
With this method, all the images of the relevant documents are evaluated and will have a score > 0. Indeed, they will inherit at least of the root node score W_a^{im}. We detail the evaluation of each score in the following paragraphs.
To evaluate the score of an image using its descendant nodes, we use the score of each relevant descendant node obtained by the XFIRM-CO method (W_{rdi}), the number of relevant descendant nodes according to the XFIRM model ($|d|$) and the number of non-relevant descendant nodes ($|\overline{d}|$).

$$W_d^{im} = f(W_{rdi}, |d|, |\overline{d}|) \tag{6}$$

If the number of relevant descendant nodes is greater than the number of non-relevant descendant nodes then they will have more importance in the score evaluation. Using this intuition, we apply the following formula in our experiments.

$$W_d^{im} = (\frac{|d| + 1}{|\overline{d}| + 1}) * \sum_{i=1}^{|d|} W_{rdi} \tag{7}$$

To evaluate the score of an image using its brother nodes, we use the score of each relevant brother node obtained by the XFIRM-CO method (W_{rbi}), the distance between the image node and each brother node ($dist(im, b_i)$): the larger the distance of the brother node from the image node is, the less it contributes to the image relevance. Finally, we use the number of relevant brother nodes $|b|$ and the number of non-relevant brother nodes $|\overline{b}|$

$$W_b^{im} = f(W_{rbi}, dist(im, b_i), |b|, |\overline{b}|) \tag{8}$$

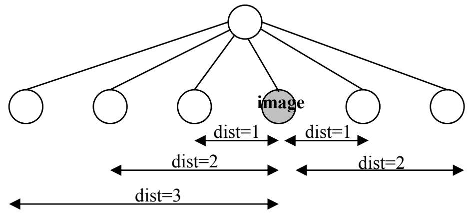

Fig. 2. distance between image node and brother nodes

Figure 2 shows how distances between an image node and its brother nodes are calculated:

The formula used in experiments presented here is :

$$W_b^{im} = (\frac{|b|+1}{|\overline{b}|+1}) * (\sum_{i=1}^{|b|} \frac{W_{rbi}}{dist(im, b_i)}) \tag{9}$$

To evaluate the score of an image using its ancestor nodes, we add the scores of relevant ancestor nodes obtained with the XFIRM-CO method (W_{rai}).
The XFIRM-CO method uses the distance between the relevant node and its ancestors to evaluate the ancestors scores of an element: the larger the distance of a node from its ancestor is, the less it contributes to the relevance of its ancestor. Our method also uses the distance $dist(im, a_i)$ between the image node and its ancestors: the larger the distance from an ancestor node is, the less it contributes to the relevance of the image node. We used the following formula:

$$W_a^{im} = \sum_{i=1}^{|a|} \frac{log(W_{rai}+1)}{dist(im, a_i)+1} \tag{10}$$

where $|a|$ is the number of relevant ancestor nodes according to the XFIRM model.

MMFragments topics processing (MMF method) : For MMFragments topics, we adapted the XFIRM CO+S method: we decomposed the query into sub-queries (figure 3). For each sub-query, if its element is different from "image", we applied the XFIRM COS method and if the subquery element is "image", we applied the MMI method. Then, we propagated scores of sub-queries to the target element using the XFIRM CO+S method.

4.2 Results

MMFragments task results are based on 9 topics, whereas MMImages task results are based on 13 topics.

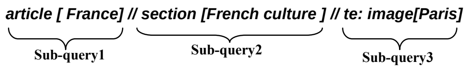

Fig. 3. Query decomposition in sub-queries

MMImages task results Two metrics are used to evaluate MMImages topics: MAP (Mean Average Precision) and BPREF (Binary PREFerence). Results are presented in table 9:
We used 4 methods: XFIRM CO method, XFIRM CO+S method, MMI method

Table 9. MMImages task results

Method	α	ρ	p_1	p_2	p_3	MAP	BPREF
CO	0.6	1	-	-	-	0.1140	0.1394
CO+S	0.9	-	-	-	-	**0.2254**	0.2060
MMI	0.6	1	0.33	0.33	0.33	0.2122	0.2065
MMI	0.6	0.9	0.33	0.33	0.33	0.2159	0.2078
MMF	0.6	-	0.33	0.33	0.33	0.2114	**0.2221**
MMF	0.6	-	0	1	0	0.2142	0.2225
MMI	0.6	0.9	0	0	1	0.2112	0.2078

and MMF method. Grayed boxes are results of our official runs. Best MAP is 0.2254 using XFIRM COS method with parameters α=0.9 and "image" as query target element.
Results for the MAP metric with the XFIRM CO+S method are better than results with the MMI method. This can be explained by the structure of the images collection (Figure4). The MMI method is based on the place of textual information in the structure (ascendants, descendants, brothers), whereas in the MMImages collection, images do not have decendant nodes and the ancestor nodes scores are only calculated with brother nodes. All the textual content of the document is thus in the brother nodes.

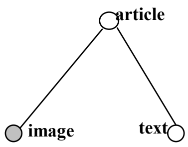

Fig. 4. MMImages collection structure

Best BPREF is 0.2225 using MMF method with parameter α=0.6, $p_1 = 0 p_2 = 1 p_3 = 0$ and "image" as target element.

MMFragments task results Table 10 shows results of the MMFragments task. They are based on the Mean Average Precision (MAP) metric. We tested Multimedia methods and XFIRM methods where no specification of images is done.

Table 10. MMFragments task results

Method	α	ρ	dict	p_1	p_2	p_3	MAP
CO	0.6	1	2	-	-	-	0.008599
CO	0.6	0.9	2	-	-	-	0.015488
CO+S	0.9	-	1	-	-	-	0.010142
CO+S	0.9	-	2	-	-	-	0.01079
MMF	0.5		2	0.33	0.33	0.33	0.01596
MMI	0.1	0.9	2	0.33	0.33	0.33	**0.01765**
MMI	0.1	0.9	2	1	0	0	0.000124
MMI	0.1	0.9	2	0	1	0	0.0084
MMI	0.1	0.9	2	0	0	1	0.01575

Grayed boxes are results of our official runs. Runs using the CO+S query processing model of the XFIRM system use two different dictionary indexes: *dict*1 contains simple tag equivalencies (for example, *image* and *figure* are considered as equivalent tags), whereas *dict*2 contains very extended equivalencies between tags.

The best MAP is obtained with the MMI method, where the target element is always an image node. We tested some values of the 3 parameters: $p_1, p_2 et p_3$. We can observe that using only descendant nodes gives worst results whereas using only ancestor nodes gives good results. All document context seems thus to contribute to the image relevance. This can be explained by the wealth of other nodes information belonging to the document that are likely to share the same subject as images.

In future work, we plan to:

- differentiate methods used to assign weights to image fragments and text fragments.
- process queries by example by using text surrounding images used as examples
- add additional sources of information to process queries of the MMImages task: Images classification scores, Images features vectors and a CBIR system,...

References

1. M. Boughanem, T. Dkaki, J. Mothe, and C. Soule-Dupuy. Mercure at TREC-7. In *Proceedings of TREC-7*, 1998.
2. L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

3. S. E.Robertson and J. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, pages 129–146, 1976.
4. N. Fuhr, M. Lalmas, S. Malik, and G. Kazai. INEX 2005 workshop proceedings, 2005.
5. L. Hlaoua, K. Pinel-Sauvagnat, and M. Boughanem. Relevance feedback for xml retrieval: using structure and content to expand queries. In *Proceedings of RCIS 2007, Ouarzazate, MOROCCO, to appear*, april 2007.
6. Y. Mass and M. Mandelbrod. Experimenting various user models for XML retrieval. In *[4]*, 2005.
7. J. Rocchio. *Relevance feedback in information retrieval.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.
8. I. Ruthven and M. Lalmas. Selective relevance feedback using term characteristics. *CoLIS 3, Proceedings of the Third International Conference on Conceptions of Library and Information Science.*, 1999.
9. K. Sauvagnat, M. Boughanem, and C. Chrisment. Why using structural hints in XML retrieval ? . In *Flexible Query Answering (FQAS) 2006 , Milano, Italia*. Advances in Artificial Intelligence, 7-10 june 2006.
10. K. Sauvagnat, L. Hlaoua, and M. Boughanem. Xfirm at inex 2005: ad-hoc and relevance feedback track. In *INEX 2005 Workshop proceedings*, pages 88–103, 2005.
11. W. Thijs. Image retrieval: Content versus context. In *RIAO*, pages 276–284, 2000.