# ENSM-SE and UJM at INEX 2010: Scoring with Proximity and Tag Weights

Michel Beigbeder[1], Mathias Géry[2], Christine Largeron[2], and Howard Seck[3]

[1] École Nationale Supérieure des Mines de Saint-Étienne
michel.beigbeder@emse.fr
[2] Université de Lyon, Saint-Étienne, France
{Mathias.Gery,Christine.Largeron}@univ-st-etienne.fr
[3] Université Paris-Dauphine, Paris, France
bseck@olaneo.fr

**Abstract.** This paper presents our participation in the *Relevant in Context task* (ad-hoc track) during the 2010 INEX competition, and a posterior analysis. Two models presented in previous editions of INEX by the authors were merged for our 2010 participation. The first one is based on the proximity of the query terms in the documents [1] and the second one is based on learnt tag weights [2]. The results demonstrate the improvement of focused information retrieval, thanks to the integration of the tag weights in the approach based on proximity.

## 1 Introduction

INEX Ad-Hoc track aims at evaluating focused XML information retrieval on large collections of structured documents in order to retrieve small units of information, smaller than document. Indeed, the structure allows to divide a document into elements which can be returned to a user instead of the whole document. But, as the tags may be used to emphasize words, the structure can also be used to improve the detection of relevant information. Thus, a word is probably more important if it is marked by certain tags, for instance if it appears in a particular font or if it appears within certain parts of a document (e.g. a title). In order to exploit this hypothesis, we proposed an extension of the BM25 weighting function, called BM25t [2], which takes the tags found in XML documents into account. In this model, a weight is estimated for each tag during a learning stage. This weight measures the capacity of the tag to emphasize terms which appear in relevant passages. This model was evaluated in a previous INEX campaign [3].

However, other approaches than the probabilistic model seemed promising in the context of focused information retrieval, notably those based on the proximity of the query terms in the documents. The use of the term positions first appeared in some implementations of the boolean model [4]. However, this model did not allow ranking of documents but this limitation was removed in subsequent works [5,6,7]. This approach has proved effective in the INEX 2009 campaign [8]. For this reason, the model based on the proximity of the query terms

S. Geva et al. (Eds.): INEX 2010, LNCS 6932, pp. 44–53, 2011.

in structured documents [1] and the BM25t model [2] were merged for our 2010 participation. This led us to study the way to take into account the structure into the model based on the proximity. So, the four runs labelled with "Emse" in the 2010 INEX campaign were done by a team both from the *École Nationale Supérieure des Mines de Saint-Étienne* and the *Université de Lyon - Saint-Étienne*.

The proximity based model uses the positions of the occurrences of the query terms in the documents to score them. More precisely, we define a text area around each occurrence of a query term. The positions belonging to this text area are influenced by this occurrence of the query term. We quantify this influence with a function, called *influence function*. Then, the influence functions of the query terms are combined in order to score the documents. A more formal presentation of these notions appears in section 2. The section 3 will be dedicated to the learning of the tag weights according to their capacity to mark relevant passages on a training collection.

The integration of the stucture into the proximity based model was done by modifying the shape of the influence functions according to the weight of the tags. In other words, in our model, the values of the function computed for an occurrence of each query term take into account the weight of the tag of the element in which this occurrence appears. Finally in section 4, we present how elements are scored with these weighted influence functions and how our runs were built with these scores. We also present runs which mix our proximity scores with the INEX Reference run.

Moreover, experiments posterior to the INEX campaign are also presented in this last section. Indeed, as it was pointed out during the INEX workshop, one limit of the evaluation in the INEX campaign of the model based on the proximity lies in the fact that this model requires boolean queries which do not exist in the evaluation framework. In order to avoid manual intervention to build boolean queries, the topic title fields were automatically transformed into boolean queries and the results obtained using these automatically generated queries are presented in this last section.

## 2 Influence Functions

### 2.1 Structure, Elements and Logical Elements

An XML document is composed of *elements*, each of them is delimited by an *opening tag* and a *closing tag*. Given an XML collection, we consider a partition of the set of tags, $B$, that appears in the collection with three subsets:

- $B_l$: the *logical tags* (or *section-like tags*, e.g. `ss1`, `ss2`, `ss3`, `ss4`);
- $B_t$: the *title tags*;
- $\overline{B_l \cup B_t}$ (the complement of the set $B_l \cup B_t$): the set of tags that are neither logical tags, nor title tags.

The structure is exploited at two levels. Firstly, the logical tags belonging to $B_l$ are used to determine the elements which can be returned to the user. Secondly,

Document $d_1$

```
<article>Document
  <ss1><st>Caesar in title</st>The <em>section</em> which <em>deals</em>
with Caesar</ss1>
  Following of the document.
</article>
```

$T = \{caesar, deals, document, following, in, of, section, the, title, which, with\}$
$E(d_1) = \{$`d1/article[1]`, `d1/article[1]/ss1[1]`, `d1/article[1]/ss1[1]/st[1]`,
`d1/article[1]/ss1[1]/em[1]`, `d1/article[1]/ss1[1]/em[2]`$\}$
$B = \{$`article`, `em`, `ss1`, `st`$\}$
$B_l = \{$`article`, `ss1`$\}$
$B_t = \{$`st`$\}$
$d_1(0) = document$
$d_1(1) = caesar$
$d_1(2) = in$
...
$d_1(9) = caesar$
...
$|d_1| = 14$
$d_1^{-1}(caesar) = \{1, 9\}$
$x_1($`d1/article[1]/ss1[1]`$) = 1$
$x_2($`d1/article[1]/ss1[1]`$) = 9$
$e(5) =$ `d1/article[1]/ss1[1]/em[1]`
$e_l(5) =$ `d1/article[1]/ss1[1]`
$b(5) =$ `em`
$M_{\texttt{st}}($`d1/article[1]`$) = \{1, 2, 3\}$
$M_{\texttt{em}}($`d1/article[1]`$) = \{5, 7\}$

**Fig. 1.** Collection example with one document

the tags belonging to $B$, including the logical tags, are used to estimate the relevance of an element as detailed in section 3.

When the vector space model considers the number of occurrences of the terms in the documents (through the term frequency or the inverse document frequency), the proximity based model, introduced by [7] takes also into account their positions in the document. Thus, a document is defined as a function which associates a term $t \in T$ to each position in the document:

$$\begin{aligned} d : \mathbb{N} &\rightarrow T \\ x &\mapsto d(x) \end{aligned} \tag{1}$$

Given a position $x$ in a document, $e(x)$ is the deepest element (in the XML tree) that surrounds the position $x$, and $e_l(x)$ is the deepest logical element that surrounds the position $x$; $b(x)$ is the tag of the element $e(x)$. Given an element $e$, $x_1(e)$ (resp. $x_2(e)$) denotes the position of its firts (resp. last) term. Figure 1 shows a sample document and illustrates all these notations. For instance, for the fifth position, corresponding to the word *section*, the deepest element is $e(5) =$ `d1/article[1]/ss1[1]/em[1]` while the deepest logical element is $e_l(5) =$ `d1/article[1]/ss1[1]`.

In order to compute the score $s(q, e)$ of an element $e$, given a query $q$, this model introduces the influence function of a term to a position and the influence

of a query to a position. These notions are briefly presented in the following sections. An extended presentation can be found in [1].

### 2.2 Influence Function of a Term to a Position

Firstly, we modelize the influence of one occurrence of term $t$ at position $i$ on one position $x$ in a document $d$ with an *influence function*. Any function with the three following properties is acceptable and modelizes the proximity idea:

- symmetric around $i$,
- decreasing with the distance to $i$,
- maximum (value 1) reached at $i$.

The simplest one is a linearly decreasing function centered around $i$: $x \mapsto \max(\frac{k-|x-i|}{k}, 0)$ where k is a parameter which controls the size of the influence area. The graphs of such functions have a triangular shape, so we call *triangle functions* these functions. When the distance between $x$ and $i$ is greater than $k$, the influence is zero – that's to say that the occurrence of term $t$ at position $i$ is too far from position $x$ to influence it. Moreover the influence is limited to the logical element $e_l(i)$ that surrounds the position $i$ of the occurrence of the query term $t$. To do that we take the product of the triangle function by the characteristic function $\mathbf{1}_{e_l(i)}$ of the position range that belongs to the logical element $e_l(i)$. Lastly, the influence should be that of the nearest occurrence of the term $t$, which can be obtained with $\max_{i \in d^{-1}(t)}$ because the influence function are symmetric and decreasing with the distance[1].

So the influence $p_t^d(x)$ of term $t$ to the position $x$ in the document $d$ is defined by:

$$p_t^d(x) = \max_{i \in d^{-1}(t)} \left( \mathbf{1}_{e_l(i)} \cdot \max\left(0, \frac{k - |x-i|}{k}\right)\right) \tag{2}$$

Though when $e(i)$ is a title-like element, the triangle function is replaced by the constant function 1. Thus one occurrence of a query term in a title spreads its influence over the whole surrounding logical element.

### 2.3 Influence Function of a Query to a Position

As explained previously, the influence function of a term to a position is used to compute the influence of a query to a position which is used itself to compute the score of an element for this query. This influence function of a query to a position is defined as follows: in the simplest case where a query $q$ contains only one term $t \in T$, the influence of the query to a position $x$ equals the influence of the term $t$ to the position $x$:

$$p_q^d(x) = p_t^d(x) \tag{3}$$

[1] The notation $d^{-1}(t)$ denotes the set of positions in the document $d$ where one occurrence of term $t$ does appear.

In the other cases, as a boolean query, the query $q$ is a tree with conjunctive and disjunctive nodes. To define the influence on a conjunctive node $q_1$ AND $q_2$ the minimum is taken over the influence functions of its children:

$$p^d_{q_1 \text{ AND } q_2}(x) = \min(p^d_{q_1}(x), p^d_{q_2}(x)) \tag{4}$$

Similarly, the influence on a disjunctive node $q_1$ OR $q_2$ is defined as the maximum over the influence functions of its children:

$$p^d_{q_1 \text{ OR } q_2}(x) = \max(p^d_{q_1}(x), p^d_{q_2}(x)) \tag{5}$$

These formulas are recursively used during a post order traversal of the query tree to compute the influence function at the root of the tree, that's to say the influence function of the query itself.

### 2.4 Score of an Element

Given the influence function of a document $d$ to a query $q$ that maps the positions in the document $d$ to [0,1] with $p^d_q(x)$, the score $s(q, e)$ of an element $e$ is computed with the following formula:

$$s(q, e) = \frac{\sum_{x_1(e) \leq x \leq x_2(e)} p^d_q(x)}{x_2(e) - x_1(e) + 1} \tag{6}$$

where $x_1(e)$ (resp. $x_2(e)$) is the first position (resp. the last position) of the textual content of the element $e$.

## 3 Weighting Tags and Modulating Influence Function Shapes

As explained previously, we suppose that the tags may be used to emphasize words. So, the structure can be used to improve the detection of relevant information. A weight is estimated for each tag using a training set. This weight measures the capacity of the tag to emphasize terms in relevant or in non relevant passages.

### 3.1 Weighting Tags

A weight is computed for each tag $b \in B$, following the learning method introduced by [2]. It estimates the probability that $b$ marks a relevant term or an irrelevant one. This weight is afterwards used to modulate the influence function of the term occurrences that appear in the elements of type $b$.

The set of assessments from INEX 2009 is used as a learning set. In the contingency table of Table 1, $R_q(e)$ is the set of the relevant positions in the element $e \in E$ for the topic $q \in Q$, and $M_b(e)$ is the set of the positions of $e$ marked by the tag $b \in B$.

**Table 1.** Contingency table for the query $q$ and for the tag $b$

| | $R_q(e)$ | $\overline{R_q(e)}$ |
|---|---|---|
| $M_b(e)$ | $t_{rm}(b,q)$ | $t_{\overline{r}m}(b,q)$ |
| $\overline{M_b(e)}$ | $t_{r\overline{m}}(b,q)$ | $t_{\overline{rm}}(b,q)$ |
| Total | $t_r^{coll}(q)$ | $t_{\overline{r}}^{coll}(q)$ |

The weight $w_b(q)$ of a tag $b$ for a query $q$ is defined by:

$$w_b(q) = \frac{\frac{t_{rm}(b,q)+s}{t_{rm}(b,q)+t_{r\overline{m}}(b,q)+s}}{\frac{t_{\overline{r}m}(b,q)+s}{t_{\overline{r}m}(b,q)+t_{\overline{rm}}(b,q)+s}} \tag{7}$$

with:

- $t_{rm}(b,q) = \sum_{e \in E} |R_q(e) \cap M_b(e)|$: number of relevant positions for the query $q$ marked by the tag $b$;
- $t_{r\overline{m}}(b,q) = \sum_{e \in E} |R_q(e) \cap \overline{M_b(e)}|$: number of relevant positions for the query $q$ not marked by the tag $b$;
- $t_{\overline{r}m}(b,q) = \sum_{e \in E} |\overline{R_q(e)} \cap M_b(e)|$: number of irrelevant positions for the query $q$ marked by the tag $b$;
- $t_{\overline{rm}}(b,q) = \sum_{e \in E} |\overline{R_q(e)} \cap \overline{M_b(e)}|$: number of irrelevant positions for the query $q$ not marked by the tag $b$.

The parameter $s$ is a smoothing parameter, which was fixed to 0.5 in our experiments.

In fact, we believe that the capacity of a tag to highlight relevant terms (or on the contrary those that are not relevant) is intrinsic to the tag itself and is not dependant on the query. Thus, we estimate the weight $w_b$ for each tag $b$ instead of a weight for each pair (tag $b$, query $q$). The weight $w_b$ of a tag $b$ is averaged using the set of 68 evaluated queries from INEX 2009, using the formula:

$$w_b = \frac{1}{|Q|} \sum_{q \in Q} w_b(q) \tag{8}$$

### 3.2 Modulating Influence Function Shapes

Then the weights of the tags are integrated into the score of an element. More precisely, the weights of the tags are used to modulate the influence function of the query term occurrences with two methods. In the first one, the height of the triangle is modified and the resulting influence function of a term is:

$$ph_t^d(x) = \max_{i \in d^{-1}(t)} \left( \mathbf{1}_{e_l(i)} \cdot \max \left( 0, w_{b(i)} \cdot \frac{k - |x-i|}{k} \right) \right) \tag{9}$$

and in the second one, both the height and the width of the triangle are modified and the resulting influence function of a term is:

$$phw_t^d(x) = \max_{i \in d^{-1}(t)} \left( \mathbf{1}_{e_l(i)} \cdot \max \left( 0, \frac{w_{b(i)} \cdot k - |x-i|}{k} \right) \right) \tag{10}$$

# 4 Experiments

## 4.1 Building Runs

For the experiments, we used the following sets of logical tags and title tags:

$$B_l = \{\texttt{article}, \texttt{sec}, \texttt{section}, \texttt{ss1}, \texttt{ss2}, \texttt{ss3}, \texttt{ss4}, \texttt{ss5}\}$$

$$B_t = \{\texttt{title}, \texttt{st}\}$$

We submitted four official runs (Emse301, Emse301R, Emse303 and Emse303R) at the *Relevance in Context* task. For the runs Emse301 and Emse301R, the influence function of a query term is *phw*, and for the runs Emse303 and Emse303R, the influence function is *ph*.

As the proximity based model requires boolean queries, for these runs the topic title fields were manually transformed into boolean queries during the competition. We call *Extended queries* this set of queries. After the competition, we conducted posterior experiments in order to obtain a system which is completely automatic in regards to the data currently available in the topics. In this posterior analysis, the following rules were applied to transform the title field into boolean queries:

- removing of the '+' operator
- replacement of the '-' operator by the NOT ('!') operator
- the remaining items (either simple terms or phrases) are connected by the AND operator.

We call *Title queries* this set of queries. Then we used the same settings used in our official runs to build another four runs which we named with the same name completed with an 'A'. Thus, for instance, the sole difference between our official run Emse301 and the run Emse301A is the set of queries used.

Table 2 recaps the settings for the runs. Letter 'R' means that the Reference run was used as described latter.

**Table 2.** Settings for our four official runs and for the four subsequent runs

| | Extended queries (INEX 2010) | Title queries (post-INEX) |
|---|---|---|
| Height modulation *ph* | Emse303, Emse303R | Emse303A, Emse303RA |
| Height and width modulation *phw* | Emse301, Emse301R | Emse301A, Emse301RA |

As each document is analyzed, a score is computed for each logical element according to formula 6. A score is computed for a document as the maximum of the scores of its descendants.

To choose some elements within a document, the scores of the elements of the document are sorted in decreasing order in a ranked list. The top ranked element is inserted in the result list. To fulfill the non overlapping requirement, at the

same time every descendants and every ascendants of this element are removed from the ranked list. This process is repeated util the ranked list is empty.

For the runs Emse301 and Emse303 and their automatic versions Emse301A and Emse303A, the elements are sorted:

1. firstly, by document score;
2. then, by document id;
3. and finally, by element score.

For the 'R' versions (Emse301R, Emse303R, Emse301RA and Emse303RA) the same sorting keys are used but the Reference run is also used. The elements are returned using the following method: the element of the documents that appear both in our results list and in the Reference run are firstly returned in the order of the Reference run, then the elements of the documents that appear only in our list and finally, the documents that only appear in the Reference run.

### 4.2 Results

The results obtained by our model during the 2010 INEX competition are presented in Table 3, together with the results obtained during our posterior analysis experiments. The results were computed using the INEX software: *inex_eval* 3.0.

**Table 3.** *MAgP* results of our four official runs and the four subsequent runs

| | Extended queries (INEX 2010) | | Title queries (post-INEX) | |
|---|---|---|---|---|
| | without R | with R | without R | with R |
| Height modulation *ph* | Emse303 0.1163 | Emse303R 0.1977 | Emse303A 0.0760 | Emse303RA 0.1591 |
| Height and width modulation *phw* | Emse301 0.1207 | Emse301R 0.1967 | Emse301A 0.0751 | Emse301RA 0.1596 |
| Baseline | Reference run 0.1436 | | | |

The first conclusion concerning our experiments is that both the Reference run and our method get benefits from the other one: use of the Reference Run is very beneficial to every methods and reciprocally all the methods that use the Reference run are significantly better than the Reference run itself.

Furthermore, the experiments permit to compare the methods used to modulate the influence functions with the tag weights. Both strategies *ph* and *phw* improve the Reference run results, but it is not clear if modifying both the height and the width of the triangles is better than only modifying the height.

Finally, the results of the subsequent runs with title queries are not as good as those obtained with extended queries. However, they stay very good when the Reference run is used. Indeed, Table 4 shows that the runs Emse301RA and Emse303RA are ranked just after the runs from "Peking University" which were ranked from 3rd to 6th during the INEX competition.

**Table 4.** INEX 2010 results (Relevant in Context task)

| Rank | *MAgP* | Institute | Run |
|---|---|---|---|
| 1 | 0.1977 | ENSM-SE | Emse303R |
| 2 | 0.1967 | ENSM-SE | Emse301R |
| 3 | 0.1615 | Peking University | 32p167 |
| 4 | 0.1615 | Peking University | 36p167 |
| 5 | 0.1598 | Peking University | 31p167 |
| 6 | 0.1598 | Peking University | 37p167 |
| **new** | 0.1596 | ENSM-SE | Emse301RA |
| **new** | 0.1591 | ENSM-SE | Emse303RA |
| 7 | 0.1588 | LIA - U. of Avignon | I10LIA1FTri |
| 8 | 0.1587 | LIA - U. of Avignon | I10LIA1FUni |
| 9 | 0.1521 | Queensland U. of Technology | Reference |
| 10 | 0.1519 | Peking University | 22p167 |

## 5 Conclusion

This article reports the results of our experiments in the *Relevance in Context* task during the 2010 INEX competition and a posterior analysis. Our official INEX 2010 runs used manually built boolean queries. The posterior experiments use automatically built queries which are a conjunctive interpretation of the title topic fields.

Two models presented in previous editions of INEX by the authors were merged for our 2010 participation. The first one is based on the proximity of the query terms in the documents through the use of influence functions around each occurrence of the query terms in the documents [1] and the second one is based on learnt tags weights [2]. The results demonstrate that the proximity model which already proved effective in the previous INEX campaigns is enhanced by modulating the shape of the influence functions of the query terms by the tag weights. Is is also shown that the two phase retrieval process with Fetch and Browse gets much benefits from the use of the BM25 based Reference run. Though the best results are obtained with actual boolean queries rather than with the conjunctive interpretation of the title topic field.

**Acknowledgements.** This work has been partly funded by the Web Intelligence project (région Rhône-Alpes, cf. `http://www.web-intelligence-rhone-alpes.org`) and the Conseil Général de la Loire.

## References

1. Beigbeder, M.: Focused retrieval with proximity scoring. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC 2010, pp. 1755–1759. ACM, New York (2010)
2. Géry, M., Largeron, C., Thollard, F.: Integrating structure in the probabilistic model for information retrieval. In: Web Intelligence, pp. 763–769 (2008)

3. Géry, M., Largeron, C., Thollard, F.: UJM at INEX 2008: Pre-impacting of tags weights. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 46–53. Springer, Heidelberg (2009)
4. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval, ch. 2. McGraw-Hill, New York (1983)
5. Hawking, D., Thistlewaite, P.: Proximity operators - so near and yet so far. In: [9]
6. Clarke, C.L.A., Cormack, G.V., Burkowski, F.J.: Shortest substring ranking (multitext experiments for TREC-4) In: [9]
7. Beigbeder, M., Mercier, A.: An information retrieval model using the fuzzy proximity degree of term occurences. In: Proceedings of the 2005 ACM Symposium on Applied Computing, SAC 2005, pp. 1018–1022. ACM, New York (2005)
8. Beigbeder, M., Imafouo, A., Mercier, A.: ENSM-SE at INEX 2009: Scoring with proximity and semantic tag information. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2009. LNCS, vol. 6203, pp. 49–58. Springer, Heidelberg (2010)
9. Harman, D.K. (ed.): The Fourth Text REtrieval Conference (TREC-4), Department of Commerce, National Institute of Standards and Technology (1995)