Sphere of influence model in information retrieval

Annabelle Mercier, Michel Beigbeder

Sphere of influence Model in Information retrieval

Annabelle Mercier, Michel Beigbeder
École Nationale Supérieure des Mines de Saint-Étienne
158, cours Fauriel F-42023 Saint-Étienne Cedex 2 – FRANCE
Email: Annabelle.Mercier, Michel.Beigbeder@emse.fr

***Abstract*— We present a new method based on fuzzy proximity for scoring and ranking relevant documents according to the user's information need. Our study takes place in the information retrieval domain and uses the terms localization in documents, we assume that the more the occurrences of the query terms are found close to each other in a document the more this document must be in the top of the retrieved document list. After a brief description of some traditional models and the few approaches which use proximity between the occurrences of terms, we present our "sphere of influence" model which scores documents according to the fuzzy proximity between the query terms in them. First, we detail this model and then we demonstrate that our model includes the traditional ones.**

I. Introduction

The information retrieval domain which is well known with the www search engines uses different models. These models give a way to select and rank documents answering user's information needs. There are three main kinds of models [1] : (i) set theoretic models (Boolean model, fuzzy set model and extended Boolean model) (ii) algebraic models (vector model and latent semantic indexing) and (iii) probabilistic models (inference network, bayesian network and belief network).

Our focus is on the two first kinds of models and on one of the first ideas about IR given by Luhn [2] : *"It is here proposed that the **frequency of word occurrences** in an article furnishes a useful measurement of word significance. It is further proposed that the **relative position within a sentence of words** having given values of significance furnishes a useful measurement for determining the significance of sentences. The significance factor of a sentence will therefore be based on a combination of these two measurement"*.

The first aspect, dealing with term frequency, was developed in the algebraic models but the second, about the relative position of words – thus the term occurrence proximity – wasn't deeply studied.

First of all, the section 2 reminds two classical information retrieval models and the few approaches using term proximity. Then, the section 3 presents our sphere of influence model. Afterwards, we explain how we can derive boolean and vector method in our model. Finally, the section 5 presents the conclusion.

II. State of the art

Information retrieval systems based on classical models use an indexing method to build the documents representation. The index is usually grounded on term occurrences. In this study, T is the set of terms and D the set of documents.

In the classical Boolean model, each document is represented by the set of terms occurring in its text and the query is a boolean expression. The query is represented by a tree whose leaves are terms and nodes are AND and OR operators. In this model, the relevance decision criterion is binary : the document score is in the set $\{0, 1\}$. So the main drawback is that the responses cannot be ranked. Nevertheless, the main advantage is that the query language is quite expressive.

Various models are based on term frequencies which catch the Luhn's first idea. In the vector model the weight $w(d, t)$ of the term t in the document d usually depends in increase order on term frequency and in decrease order on document frequency – that is to say the number of documents where a given term occurs. The document score can be computed with various formulas based on the $tf \cdot idf$ scheme (tf stands for term frequency and idf stands for inverse document frequency). Documents and queries are represented by term vectors. The similarity measure between a document and a query can be taken as the cosine of the angle between the document and the query vectors. The query model is a bag of words so it is simple but less expressive than the Boolean one. The main advantage of this model over the Boolean one is the capacity to arrange documents according to their score, so they can be shown in decrease system relevance order to the user. This ranking capacity is fundamental in information retrieval because the evaluation methods use this to compare systems. So, the lack of ranking in Boolean model led to the introduction of extended Boolean model [3] and fuzzy set models [4].

In order to graduate the score given by the basic Boolean model, different models based on fuzzy set theory were built [4]. In these models, a function μ_t is associated to each term $t \in T$. This function expresses the membership level of the documents to the fuzzy set corresponding to a term t. In this model, a query is also represented by a tree. The membership level to a node with the OR operator (resp. AND) is the maximum (resp. minimum) of its sons' membership level, that is to say this node is the fuzzy union (resp. intersection) of its sons' fuzzy subsets. Finally, we have a membership level at the query tree root. This permits to score a document with a value in the interval $[0, 1]$ which allows to rank documents on the contrary to the basic Boolean model.

These three latest models (Boolean, vector and fuzzy models) do not consider the position of term occurrences. A classic extension of Boolean model adds an operator to the query language to express proximity constraints. This operator is called NEAR, ADJacent, or WINDOW in the systems which implement it [5]. It works like the AND operator with a constraint on the position of term occurrences. It allows to specify a maximal distance between two term occurrences. For example, if we want to find A and B with a distance of 5 or less, we give the query: A NEAR 5 B. In our model, we do not use the NEAR operator because it can only be used with leaves and its generalization to sub-trees

is not consistant [6]. Some works deal with the implementation issues of these operators and their consequences on the index. Keen studied [7], [8] the recall and precision performances of different implementations of the proximity operator and showed that the NEAR operator improves the precision for the retrieved documents set. However, the relevance is still binary and the documents cannot be ranked.

Some recent approaches straightly use term proximity to score the documents [9], [10], [11]. These methods start with the selection of intervals which contain the query terms in the documents. Each method has its own rules to select the intervals. Clarke and al. method [9], [12], [13] selects the shortest intervals which contain all the query terms so that the selected intervals are not nested. In Hawking and al. method [10], for each term occurrence, the shortest interval which contains all the terms and beginning on this occurrence is selected. The Rasolfo and al. method [11] selects the intervals including two query terms and the interval length has to be less than 5 words. Then, each interval obtains a score (the shorter the interval, the higher its score) and contributes to the global score of the document. Finally, the document score is the sum of the selected intervals scores. The results with these methods are reported to be better than those obtained with the classical ones [14]. The next section present our model based on proximity.

III. Our model

On one hand, the Boolean model takes into account the membership of a term in a document to a term and on the other hand, the vector model takes into account the term frequency. These two models proceed with a **global** approach of the influence of any term, on the the score of the documents according to the query. This means that the relevance score does not depend on the position of terms occurrences. However, meaning of the sentences in a document doesn't only depend on the vocabulary used but on the arrangement of the vocabulary terms too. Thus, our model is **local** in the sense that we modelize the locality of term occurrences influence. At a given position, we can consider two notions of locality:

- **A term proximity** :
  At a precise text location, are we near a query term occurrence? This proximity will be graduated and so it will be a "fuzzy proximity".
- **A local relevance** :
  Is a precise text location relevant to a query term? The more occurrences are nearby this location, the higher it will be considered relevant.

In each case, an **influence function** is used to represent the influence of the words. Such a function is a relation from $\mathbb{R}$ to $[0, 1]$ with a finite support, increasing on $\mathbb{R}^-$, decreasing on $\mathbb{R}^+$. Various functions can be used. Firstly, it's possible to choose different function families (Hamming, Hanning or Gaussian functions, rectangular or triangular functions and so on). Then, in a family, it's possible to choose various values for the function parameters, for instance to obtain different influence function for the query terms. In particular, we call k the parameter which allows to control the support of the function and defines the influence area length. The triangular influence function can be expressed with :

$$f(x) = \max(\frac{k - |x|}{k}, 0).$$

For one term occurrence at position i, the translation $g(x) = f(x - i)$ of an influence function f , allows to modelize the fuzzy proximity to this occurrence (resp. the local relevance carried by this occurrence).

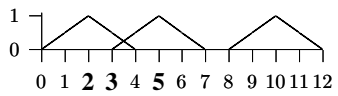

Fig. 1. At the position $x = 3$, the fuzzy proximity value is the maximum value between the fuzzy proximity values of occurrences in positions $x = 2$ and $x = 5$ thus this of occurrence at the position $x = 2$

In both models, the query evaluation for a given document is computed from the leaves. Firstly, the local relevance value functions (resp. fuzzy proximity) of query terms associated to the tree leaves are computed. These functions are defined at each position x in the documents. Then, going upward, these functions are combined at each tree level using the functional operators assigned to the OR and AND operators. Finally, we compute the document score by integrating the local relevance value function (resp. fuzzy proximity) obtained at the query tree root.

A. Fuzzy proximity model

It is natural to consider that the fuzzy proximity value $p_t^d(x)$ to a given term t at a position x of a document d is the proximity value of the nearest occurrence of the term t. For example, consider the document of figure 1 where the term t occurs at the positions 2, 5 and 10. The fuzzy proximity value at the position $x = 3$ is that of the nearest occurrence that is to say the occurrence at the position 2.

The influence functions defined before are decreasing with regard to the distance to 0, so the fuzzy proximity function to a term occurrence are decreasing with regard to the distance to the occurrence position. So obtaining the proximity value of the nearest occurrence consists in taking the maximum of the different fuzzy proximity functions of the different occurrences :

$$p_t^d(x) = \max_{i \in Occ(t,d)} f(x - i)$$

where $Occ(t, d)$ is the set of the term t occurrences positions in the document d and f is the chosen influence function.

These functions$x \mapsto p_t^d(x)$ are associated to the query tree leaves. We now want to define the fuzzy proximity value functions at the internal nodes of the query tree. Given a node, the function will be defined with the fuzzy proximity functions of its sons.

Consider two documents, the first one including the terms B an C at the positions 7 and 12 (cf. fig. 2) and the second one including two occurrences of the term B at the same positions (cf. fig. 3). The function p_B^d (resp. p_C^d) associates to each position in the document d the fuzzy proximity value to the term B (resp. C). Consider the query B OR C, for such an information need, finding B or C in the text is equivalent. Thus we want the same proximity

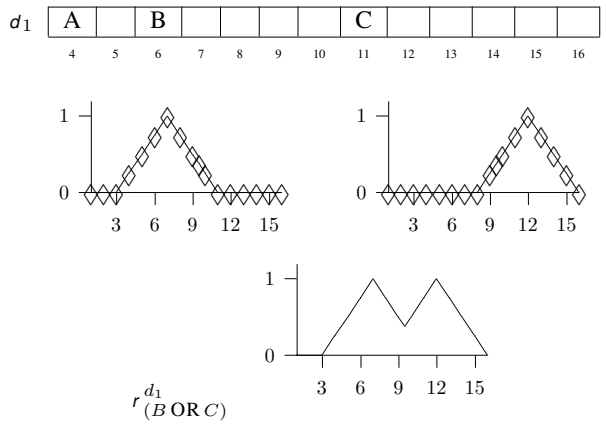

Fig. 2. Document 1 – fuzzy proximity. The first (resp. second) curve represents the fuzzy proximity to the term B (resp. term C), the last curve represents the fuzzy proximity to the query (B OR C).

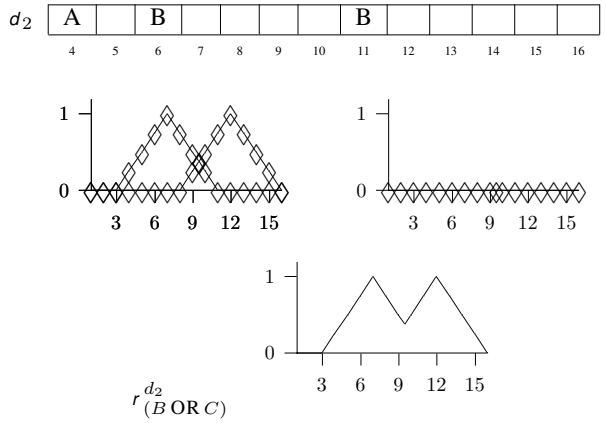

Fig. 3. Document 2 – fuzzy proximity. The first curve represents the influence functions of the two occurrences of the term B. The second function (influence function of the term C) is flat because the term C does not occur in the document d_2. The last curve represents the fuzzy proximity to the query (B OR C).

function for these 2 documents (cf. third curve in figures 2 and 3) and this can be obtained with

$$p^d_{B \text{ OR } C}(x) = \max(p^d_B(x), p^d_C(x)).$$

We generalize this by setting

$$p^d_{q \text{ OR } q'} = \max(p_q, p_{q'})$$

for a node which sons are not only terms but are the queries q and q'. This corresponds to the operation made in the fuzzy set model (section 2.3) for the union operation. With this interpretation, $q \text{ OR } q'$ appears as the (fuzzy) union of the fuzzy sets q and q'. By analogy for an AND operator, the fuzzy proximity at a node q AND q' is obtained with

$$p^d_{q \text{ AND } q'} = \min(p_q, p_{q'})$$

which corresponds to the fuzzy intersection. Moreover, our model can easily use other functions applied in fuzzy logic for the AND and OR operators [15], [16]. For example, the OR operator can be associated to the function $\max(x + y - 1, 0)$ and the AND

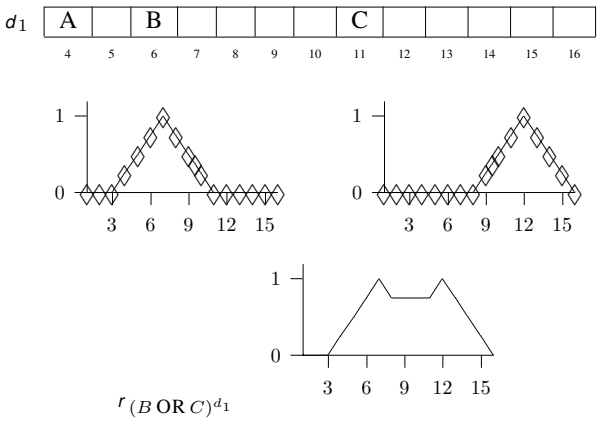

Fig. 4. Document 1 – local relevance (signal). The first (resp. second) curve represents the local relevance of the term B (resp. term C), the last curve represents the local relevance for the query (B OR C).

operator to the function $\min(x + y, 1)$ with x and y taken in two different fuzzy sets.

The last step is to determine the relevance score $s(q, d)$ for the document d according to the query q. In the vector model, the relevance score values are either inner products or cosines. Both of them are sums that can be interpreted as accumulations of pieces of relevance. The integral calculus methods capture this accumulation idea by computing the surface below a curve. So, we express the score as

$$s(q, d) = \int_{-\infty}^{+\infty} p^d_q(x) \, \mathrm{d}x \, .$$

An approximation consists in adding the fuzzy proximity values at each position x of the document, the result is a score in $\mathbb{R}^+$. This score depends on the fuzzy proximity of each term. We have already experimented this alternative (fuzzy proximity) of our sphere of influence model, some results are available in [17].

B. Local relevance model

In this approach, we consider that the term occurrences give a piece of local relevance around their positions. The relevance signal is represented by an influence function like those described before. We use these functions to compute the document relevance score. First of all, we consider the signals for each query term occurrence. Then given a term t and a document d, in order to gather the different pieces of relevance given by each occurrence of t, at each text position we add the local relevance values (signals) computed for all the term occurrences and we express the local relevance at the text position x with

$$r^d_t(x) = \sum_{i \in Occ(t,d)} f\,(x - i).$$

For the query evaluation, we have to combine the signals according to the AND and OR operators. Firstly, we consider the case of disjunctive queries. As in the proximity model, we want that the contribution of either two occurrences of the same term B or one of the term B and one of the term C are equivalent.

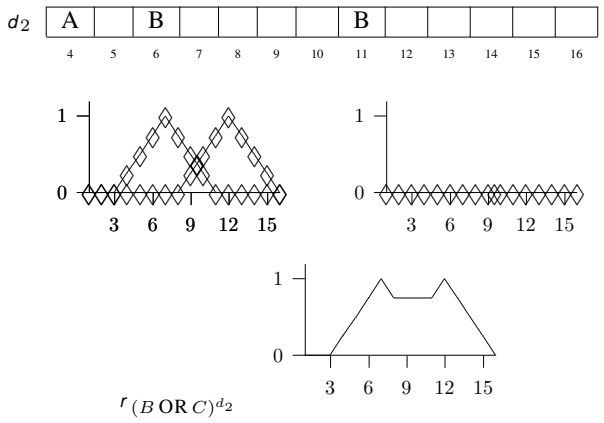

Fig. 5. Document 2 – local relevance (signal). The first curve represents the influence functions of the two occurrences of the term B. The second function (influence function of the term C) is flat because the term C does not occur in the document d_2. The last curve represents the local relevance to the query (B OR C).

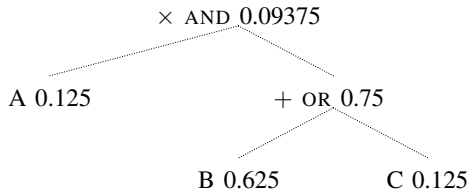

Fig. 6. The query tree for $A \text{ AND } (B \text{ OR } C)$. The local relevance values at position 8.5 are at the leaves, the functions used at the nodes are before the boolean operators and the evaluations of these functions are displayed at each nodes.

So, we set for the OR operator

$$r^d_{q \text{ OR } q'} = r^d_q + r^d_{q'}.$$

We must define a functional operator to apply on AND nodes. If we try to use the min function as in the fuzzy proximity model, the query A AND (B OR C) leads to an incoherence. In fact, we should have $r^d_{((A \text{ AND } (B \text{ OR } C))} = r^d_{(A \text{ AND } B) \text{ OR } (A \text{ AND } C))}$ at each position of the document d, but this is not the case. For example, at the position $x = 8.5$ in the first document where A, B and C occur, the local relevance values at the position x are $r^d_A = 0.125$, $r^d_B = 0.625$ and $r^d_C = 0.125$, we obtain :

$$r^{d_1}_{(A \text{ AND } (B \text{ OR } C))}(8.5) = \min(0.12, 0.12 + 0.62) = 0.12$$

and

$$r^{d_1}_{((A \text{ AND } B) \text{ OR } (A \text{ AND } C))}(8.5)$$
$$= \min(0.12, 0.62) + \min(0.12, 0.12) = 0.24$$

so, the Morgan's laws are not verified and we cannot use this function at the operator AND nodes of the tree. These laws can simply be verified by setting for the AND operator

$$r^d_{q \text{ AND } q'} = r^d_q \cdot r^d_{q'}.$$

The figures 4 and 5 show the local relevance value functions for the query (B OR C) of the two example documents, in this second model we also have the same local relevance for the query if the

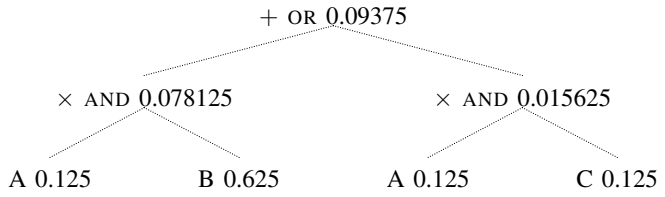

Fig. 7. The query tree for $(A \text{ OR } B) \text{ AND } (A \text{ OR } C)$. The local relevance values at position 8.5 are at the leaves, the functions used at the nodes are before the boolean operators and the evaluations of these functions are displayed at each nodes.

document contains an occurrence of B or C at the position $x = 7$ and $x = 12$. In the figure 6, the query (A AND (B OR C)) is represented with the values of the local relevance function to the document d_1 at the position $x = 8.5$. The values are computed bottom up using addition for OR nodes and multiplication for AND nodes. The value at the root is used to compute the score. The same evaluation is shown for the query ((A AND B) OR (A AND C)) in figure 7 which is another boolean expression to formulate the information need.

As in the fuzzy proximity case, the score of the document d for a given query q is defined with

$$s(q, d) = \int_{-\infty}^{+\infty} r^d_q(x) \, \mathrm{d}x \, .$$

The score, in $\mathbb{R}^+$, is totally determined by the local relevance values so it allows to take into account the relative position between terms which corresponds to the second Luhn's idea (cf. section 1).

IV. Special cases

Varying the parameter k allows to control the influence sphere width. With such variations the usual formulations of the classical models like the coordination level, the vector and the Boolean ones can be derived.

A. Coordination level and vector model

In the information retrieval domain, one of the first similarity measures between a document and a query was the coordination level. In this case, the query is a set of keywords. Computing the relevance score by the coordination level method consists in counting the query terms occurrences number in a document. We can reproduce this behaviour in our model by :

1) choosing a rectangular influence function of width 1 and height 1 (cf. fig. 8) ; so the sphere of influence of any term occurrence is limited to this occurrence as the spheres of influence do not overlap,
2) using a disjunctive query.

So, by only taking into account the raw term frequency, our computing method is equivalent to the coordination level one.

The behaviour of the vector model can be reproduced if we assign at the positions where the query terms occur a local relevance value (resp. fuzzy proximity value) which depends on the document frequency[1]. so, we get documents scores which

[1]We can also use various *idf* functions with normalised value for that.

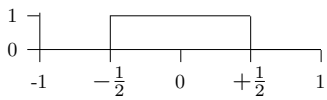

Fig. 8. Influence function which limits the influence area, the zero value match with the normalized *idf* value for the given term.

depend on document frequency and term frequency which well corresponds to the vector model.

B. Boolean Model

In our fuzzy proximity model, if the sphere of influence is extended to the whole document, what will be done by taking limits when the parameter k tends towards the infinity, our computing method behaves like the Boolean one. Here is the demonstration.

First of all, an influence function which is a rectangular function of width $2k$ and height $\frac{1}{2k}$

$$x \mapsto \frac{1}{2k} \cdot \mathbb{1}_{[-k,l+k]}(x)$$

will be used[2]. This function is illustrated in figure 9.

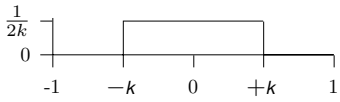

Fig. 9. Rectangular influence function.

Given a term t and a document d of length $l+1$, we maximize the function p_t^d at any position x

$$p_t^d(x) = \max_{i \in Occ(d,t)} f(x-i) \leq \max_{i \in [0,l]} f(x-i) \leq \frac{1}{2k} \cdot \mathbb{1}_{[-k,l+k]}(x).$$

Given a query q, this maximisation is true for every leaf, so trivially it is true for every nodes in the tree. Using this maximization at the tree root, we have

$$s_k(q,d) = \int_{-\infty}^{+\infty} p_q^d(x)\ \mathrm{d}x \leq \int_{-\infty}^{+\infty} \frac{1}{2k} \cdot \mathbb{1}_{[-k,l+k]}\ \mathrm{d}x = \frac{l+2k}{2k}$$

and

$$\lim_{k \to +\infty} s_k(q,d) \leq \lim_{k \to +\infty} \frac{l+2k}{2k} = 1.$$

A query q in our model is composed of leaves with terms and internal nodes with AND and OR operators. When such a query is developped by distributing the AND operators over the OR operators, we obtain a disjunctive normal form $q = q_1\ OR\ q_2\ OR\ \ldots\ OR\ q_n$ where all the conjunctive terms[3] $(q_i)_{1 \leq i \leq n}$ are composed of elements of T. Such a document scores to 1 in the Boolean model and we will now proove that $\lim_{k \to +\infty} s_k(q,d)$ is equal to 1.

[2] $\mathbb{1}_E$ denotes the characteristic function of the set E.

[3] Here, "term" is used in its algebraic meaning.

Consider now a document which matches the boolean query. Such a document matches at least one of the $(q_i)_{1 \leq i \leq n}$, say q_{i_0}. And we have

$$p_q^d = \max_{1 \leq i \leq n} p_{q_i}^d \geq p_{q_{i_0}}^d.$$

Remembering that q_{i_0} is a conjunctive query, it can be written $t_1\ AND\ t_2\ AND\ \ldots\ AND\ t_k$ for some $(t_j)_{1 \leq j \leq k} \subset T$. As d matches (q_{i_0}), each term t_j, for $1 \leq j \leq k$, appears in the document d. The function $r^d q_{i_0}$ is the "intersection" of every influence functions and thus is the "intersection" of the two farthest one. Let us denotes u (resp. v) the first (resp. the last) position where an occurrence of one of the $(t_j)_{1 \leq j \leq k}$ does appear, formally

$$u = \min \bigcup_{1 \leq j \leq k} Occ(t_j, d)$$

and

$$v = \max \bigcup_{1 \leq j \leq k} Occ(t_j, d).$$

With these notations, we can derive

$$p_{q_{i_0}}^d = p_{t_1\ AND\ \ldots\ AND\ t_k}^d = \min_{1 \leq j \leq k} p_{t_j}^d$$

and this is equal to $\min(p_U^d, p_V^d)$ for the term U that appears at position u in d and for the term V that appears at position v in d (cf. fig 10). As $p_U^d = \frac{1}{2k} \cdot \mathbb{1}_{[u-k,u+k]}$ and $p_V^d = \frac{1}{2k} \cdot \mathbb{1}_{[v-k,v+k]}$ we have $\min(p_u^d, p_v^d) = \mathbb{1}_{[v-k,u+k]}$ (as shown by 10). So we have

$$p_q^d(x) \geq p_{q_{i_0}}^d(x) = \frac{1}{2k} \cdot \mathbb{1}_{[v-k,u+k]}(x)$$

and then

$$s_k(q,d) = \int_{-\infty}^{+\infty} p_q^d(x)\mathrm{d}x \geq \int_{-\infty}^{+\infty} \frac{1}{2k} \cdot \mathbb{1}_{[v-k,u+k]}(x)\mathrm{d}x.$$

The last sum can be computed:

$$\int_{-\infty}^{+\infty} \frac{1}{2k} \cdot \mathbb{1}_{[v-k,v+k]}(x)\mathrm{d}x = \frac{1}{2k} \cdot (u+k) - (v-k) = \frac{2k + (u-v)}{2k}$$

so

$$\lim_{k \to +\infty} s_k(q,d) \geq \lim_{k \to +\infty} \frac{2k-u+v}{2k} = 1.$$

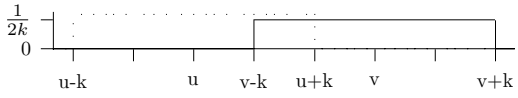

Fig. 10. The intersection of the rectangles represents the local document relevance. We have two term occurrences one at position u and the other one at position v.

As we previously proved that this limit was lower than 1, it is then equal to 1.

Reciprocally, consider now a document d which does not match the boolean query. So d does not match any of the $(q_i)_{1 \leq i \leq n}$. Given some i, $1 \leq i \leq n$, q_i is a conjunctive query :

$$t_1\ AND\ t_2\ AND\ \ldots\ AND\ t_k$$

and at least one of the (t_j), $1 \leq j \leq k$, say t_{j_0} does not appear in the document d. So

$$(\forall x) \qquad p^d_{t_{j_0}}(x) = 0,$$

and then

$$(\forall x) \qquad p^d_{q_i}(x) = 0,$$

and finally,

$$p^d_q = \max_{1 \leq i \leq n} p^d_{q_i} = 0.$$

Thus, the integral is zero whatever the value of k and its limits is zero too.

Therefore, on one hand if a document d matches a boolean query q we showed that $\lim_{k \to +\infty} s_k(q, d) = 1$ and on the other hand if the document d doesn't match the query q we have $s_k(q, d) = 0$ for any value of k, so a function $\lim_{k \to +\infty} s_k(q, d) = 0$. So we proved that the classic Boolean scores can be retrieved by our model when we consider the limit as k tends towards the infinity.

V. Conclusion

In this paper, firstly, we reminded the classical models and made the link with the use of term occurrences proximity in information retrieval. Then, from our assumption that documents having nearby query terms occurrences should be highly ranked, we detailed our "sphere of influence" model which uses boolean queries. We also have seen that classical information retrieval models are special cases either with $k = \frac{1}{2}$ for the vector model or with $k \to \infty$ for the Boolean model. Actually, this parameter controls the spread of the terms occurrences influence. A value about 5 specifies a phrase level proximity. A value from 15 to 30 is at the sentence level and a value near 100 at the paragraph level. Consequently, our model reaches our first aim that is to say scoring a document according to the query terms localisation but also can be set to reproduce the behaviour of classical information retrieval models.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. [Online]. Available: http://www.sims.berkeley.edu/~hearst/irbook/

[2] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, pp. 159–168, 1958.

[3] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill International, 1983.

[4] S. Miyamoto, *Fuzzy Sets in information retrieval and cluster analysis*. Kluwer Academic Publishers, 1990.

[5] G. Salton, E. A. Fox, and H. Wu, "Extended Boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.

[6] P. C. Mitchell, "A note about the proximity operators in information retrieval," in *meeting on Programming languages and information retrieval*. ACM Press, 1973, pp. 177–180.

[7] E. M. Keen, "The use of term position devices in ranked output experiments," *The Journal of Documentation*, vol. 47, no. 1, pp. 1–22, 1991.

[8] ——, "Some aspects of proximity searching in text retrieval systems," *Journal of Information Science*, vol. 18, pp. 89–98, 1992.

[9] C. L. A. Clarke, G. V. Cormack, and E. A. Tudhope, "Relevance ranking for one to three term queries," *Information Processing and Management*, vol. 36, pp. 291–311, 2000.

[10] D. Hawking and P. Thistlewaite, "Proximity operators - so near and yet so far," in *TREC-4 proceedings*, D. K. Harman, Ed., 1995. [Online]. Available: http://trec.nist.gov/pubs/trec4/papers/anu.ps.gz

[11] Y. Rasolofo and J. Savoy, "Term proximity scoring for keyword-based retrieval systems," in *ECIR 2003 proceedings*, 2003, pp. 207–218.

[12] C. L. A. Clarke, G. V. Cormack, and F. J. Burkowski, "Shortest substring ranking," in *The Fourth Text REtrieval Conference (TREC-4)*, 1995. [Online]. Available: http://trec.nist.gov/pubs/trec4/papers/uwaterloo.ps.gz

[13] C. Clarke and G. Cormak, "Interactive substring retrieval: Multitext experiments for trec5," in *The Fifth Text REtrieval Conference*, 1996. [Online]. Available: http://trec.nist.gov/pubs/trec5/papers/waterloo.ps.gz

[14] A. Mercier, "Etude comparative de trois approches utilisant la proximite entre les termes de la requete pour le calcul des scores des documents," in *INFORSID 2004 proceedings*, May 2004, pp. 95–106.

[15] B. Bouchon-Meunier, *La logique floue et ses applications*. Addison-Wesley, 1995.

[16] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *International Journal Fuzzy Sets Systems*, vol. 1, pp. 3–28, 1978.

[17] M. Beigbeder and A. Mercier, "Evaluating boolean queries taking into account the fuzzy proximity of term occurrences," in *20th Annual ACM Symposium on Applied Computing, SAC 2005*, March 2005.