Fuzzy Proximity Ranking with Boolean Queries

Annabelle Mercier and Michel Beigbeder

École Nationale Supérieure des Mines de Saint-Étienne (ENSM-SE)

158 cours Fauriel, 42023 Saint-Etienne Cedex 2, FRANCE

{annabelle.mercier, mbeig}@emse.fr

Abstract

Based on the idea that the closer the query terms are in a document, the more relevant this document is, we experiment an IR method based on a fuzzy proximity degree of the query term occurences in a document to compute its relevance to the query. Our model is able to deal with Boolean queries, but contrary to the traditional extensions of the basic Boolean IR model, it does not explicitly use a proximity operator. The fuzzy term proximity is controlled with an influence function. Given a query term and a document, the influence function associates to each position in the text a value dependant on the distance of the nearest occurence of this query term. To model proximity, this function is decreasing with distance. Different forms of function can be used: triangular, gaussian etc. For practical reasons only functions with finite support were used. The support of the function is limited by a constant called k. The fuzzy term proximity functions are associated to every leaves of the query tree. Then fuzzy proximities are computed for every nodes with a post-order tree traversal. Given the fuzzy proximities of the sons of a node, its fuzzy proximity is computed, like in the fuzzy IR models, with a mimimum (resp. maximum) combination for conjunctives (resp. disjunctives) nodes. Finally, a fuzzy query proximity value is obtained for each position in this document at the root of the query tree. The score of this document is the integration of the function obtained at the tree root. For the experiments, we modified Lucy (version 0.5.2) to implement our IR model. Three query sets are used for our eight runs. One set is manually built with the title words and some description words. Each of these words is OR'ed with its derivatives like plurals for instance. Then the OR nodes obtained are AND'ed at the tree root. The two automatic query sets are built with an AND of automatically extracted terms from either the title field or the description field. These three query sets are submitted to our system with two values of k: 50 and 200. As our method is aimed at high precision, it sometimes give less than one thousand answers. In such cases, the documents retrieved by the BM-25 method implemented in Lucy was concatenated after our result list.

1 Introduction

In the information retrieval domain, the systems are based on three basic models: The Boolean model, the vector model and the probabilistic model. These models were derived within many variations (extended Boolean models, models based on fuzzy sets theory, generalized vector space model,...) [1]. Though, all of them are based on weak representations of documents: either sets of terms or bags of terms. In the first case, what the information retrieval system knows about a document is if it contains or not a given term. In the second case, the system knows the number of occurences – the *term frequency*, *tf* – of a given term in each document. So whatever is the order of the terms in the documents, they share the same index representation if they use the same terms. The worthy of note exceptions are most of the Boolean model implementations which propose a NEAR operator [10]. This operator is a kind of AND but with the constraint that the different terms are within a window of size n, where n is an integral value. The set of retrieved documents can be restricted with this operator, for instance, it is possible to discriminate documents about "data structures" and those about "data about concrete structures". Using this operator results in an increase in precision of the system [5]. But the Boolean systems that implement a NEAR operator share the same limitation as any basic Boolean system: These systems are not able to rank the retrieved documents because with this model a document *is* or *is not* relevant to a query. In fact, different extensions were proposed to the basic Boolean systems to circumvent this limitation. These extensions represents the documents with some kind of term weights most of the time computed on a *tf* basis. Then they apply some combining formulas to compute the document score given the term weigths and the query tree. But these extensions are not compatible with the NEAR operator. So some works defined models that attempt to directly score the documents by taking into account the proximity of the query terms within them.

2 Uses of Proximity

Three methods were proposed to score the documents by taking into account some sets of intervals containing the query terms. These methods differ in the set of intervals that are selected in a first step, and then in the formulas used to compute a score for a given interval. The method of Clarke and al. [2] selects the shortest intervals that contains all the query terms (this constraint is relaxed if there are not enough retrieved documents), so the intervals cannot be nested. In the method of Hawking and al. [4], for each query term occurence, the shortest interval containing all the query terms is selected, thus the selected intervals can nest. Rasolofo and al. [8] chose to select intervals only containing *two* terms of the query, but with the additionnal constraint that the interval is shorter than five words.

Moreover, the passage retrieval methods use indirectly the notion of proximity. In fact, in several methods, document ranking is done by selecting doc-

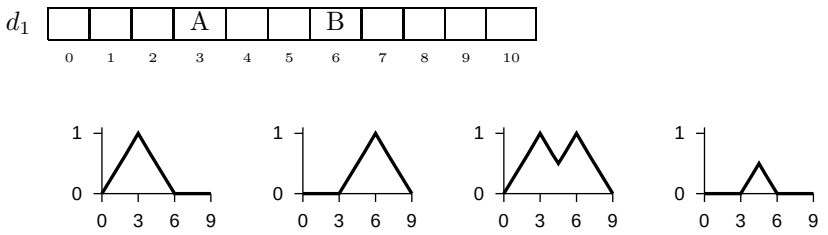

Figure 1: Document 1 – In order, w_A^{d1}, w_B^{d1}, $w_{A \text{ OR } B}^{d1}$ and $w_{A \text{ AND } B}^{d1}$ are displayed.

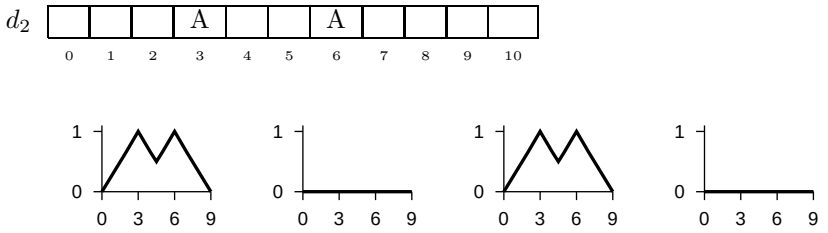

Figure 2: Document 2 – In order, w_A^{d2}, w_B^{d2}, $w_{A \text{ OR } B}^{d2}$ and $w_{A \text{ AND } B}^{d2}$ are displayed.

uments which have passages with high density of query terms, that is to say documents where the query terms are closed [12, 3, 6]. The next section presents our method based on term proximity to score the documents.

3 Fuzzy Proximity Matching

To address the problem of scoring the documents by taking into account the relative order of the words in the document, we have defined a new method based on a *fuzzy proximity* between each position in the document text and a query. This fuzzy proximity function is summed up over $\mathbb{Z}$ to score the document.

We modelize the fuzzy proximity to an occurence of a term with an influence function f that reaches its maximum (value 1) at the value 0 and decreases on each side down to 0. Different types of functions (Hamming, rectangular, gaussian, etc.) can be used. In the sequel the examples and the experiments will be based on a triangular function $x \mapsto \max(\frac{k-|x|}{k}, 0)$. The constant k controls the support of the function and this support represents the influence extent of each term occurence. A similar parameter can be found for other shapes.

So, for a query term t, the fuzzy proximity function to the occurence at position i of the term t is $x \mapsto f(x - i)$. Now, we define the term proximity function w_t^d which modelizes the fuzzy proximity at the position x in the text to the term t by combining the fuzzy proximity functions of the different occurences

of the term t:

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

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

Figure 1 and Fig. 2 show the fuzzy proximity functions $w_A^{d_1}$, $w_B^{d_1}$, $w_A^{d_2}$, and $w_B^{d_2}$ to the terms A and B in the documents d_1 and d_2.

The query model is that of the classical Boolean model: A tree with terms on the leaves and OR or AND operators on the internal nodes. At an internal node, the proximity functions of the sons of this node are combined in the query tree with the usual formulas pertaining to the fuzzy set theory. So the fuzzy proximity is computed by

$$w_{q\,\mathrm{OR}\,q'}^d = \max(w_q^d, w_{q'}^d)$$

for a disjunctive node and by

$$w_{q\,\mathrm{AND}\,q'}^d = \min(w_q^d, w_{q'}^d)$$

for a conjunctive node. With a post-order tree traversal a fuzzy proximity function to the query can be computed at the root of the query tree as the fuzzy proximity functions are defined on the leaves.

So we obtain a function w_q^d from $\mathbb{Z}$ to the interval $[0,1]$. The result of the summation of this function is used as the score of the document:

$$s(q,d) = \sum_{x=-\infty}^{+\infty} w_q^d(x) \ .$$

So, the computed score $s(q,d)$ depends on the fuzzy proximity functions and it allows to rank the documents according to the query term proximity in the documents.

4 Experiments in the Robust Track

We carried out experiments in the TREC 2005 Robust Track evaluation campaign[1]. We use the retrieval tool LUCY which is based on the Okapi BM-25 information retrieval model [9] to index this collection. Our method was easily integrated into this tool because it keeps in the index the occurence positions of the terms in the documents.

For this track, we use the AQUAINT test collection, which is composed of newspapers articles in XML format. Figure 3 shows the origin and the number of documents in this corpus.

For each document (`<DOC>` tag), the field `<DOCNO>` with the tag and the document number, the textual contents of the tags `<TEXT>`, `<P>`, `<HEADLINE>`, `<DOCTYPE>` are passed to LUCY.

[1] http://trec.nist.gov/

articles from	1996	1997	1998	1999	2000
APW			107 882	77 876	53 818
NYT			85 817	104 698	90 829
XIN	93 458	95 563	103 470	104 698	82 244

Figure 3: Number of documents indexed by newpapers/year.

4.1 Building the queries

Each topic has three parts: `<title>`, `<desc>`, `<narr>`. We built three sets of queries for our experiments. They were either manually or automatically built from the textual contents of the title and description fields.

Automatically built queries (two sets). For the first set, a query is composed of the terms from the title field where the stop words are removed.

Let us look at an example. Here is the original topic #375:

```
<top>
<num> Number:  375
<title> hydrogen energy
<desc> Description:
What is the status of research on hydrogen as a feasible energy source?
<narr> Narrative:
A relevant document will describe progress in research on controlled
hydrogen fusion or the use of hydrogen as fuel to power engines.
</top>
```

The topic number and the title fields are extracted and concatenated:

```
375 hydrogen energy
```

From this form, the queries are automatically built by simple derivations:

Lucy: `375 hydrogen energy`
conjunctive fuzzy proximity: `375 hydrogen & energy`

For the second set of automatically built queries, terms are extracted from the text of the description field by a natural langage processing method [11]. With the topic #375, the queries built by this method are:

Lucy: `375 energy feasible hydrogen source status`
conjunctive fuzzy proximity: `375 energy & feasible & hydrogen & source & status`

For the automatic runs, we only used conjunctive queries.

Manually built queries (one set). They are built with all the terms from the title field and some terms from the description field. The general idea was to build conjunctions (which are the basis of our method) of disjunctions. The disjunctions are composed of the plural form of the terms and some derivations to compensate the lack of a stemming tool in LUCY. Sometimes some terms from the same semantic field were grouped together in the disjunctions.

The queries for the native method implemented in the LUCY tool are the flat queries composed of the different derivations of the terms. Here is an example with the topic #375:

```
fuzzy proximity: 375 (hydrogen & (energy | energies | fusion))
                     & ((power | powers) & (engine | engines))
Lucy:            375 hydrogen energy energies fusion power powers
                     engine engines
```

4.2 Building the Result Lists

The Okapi model and our fuzzy method with different values of k were compared. It is known that the Okapi method is one of the best performing one. On another hand a previous study showed that the proximity based methods improve retrieval [7]. If one of our experiments with our proximity based method does not retrieve enough documents (one thousand for the TREC experiments), then its results list is supplemented by the documents from the Okapi result list that have not yet been retrieved by the proximity based method.

4.3 The Runs

In the official runs, the queries used were:

1. the conjunction of the terms automatically extracted from the title field with $k = 50$ (run `RIMam05t050`) and with $k = 200$ (run `RIMam05t200`);
2. the conjunction of the terms automatically extracted from the description field (by a NLP method) with $k = 200$ (run `RIMam05d200`);
3. manually built queries with terms from the title and description fields with $k = 50$ (run `RIMam05l050`) and with $k = 200$ (run `RIMam05l200`).

For the runs `LucyTitle`, `LucyDesc` and `LucyLemme`, the queries are flat (bag of terms). These runs provide the baselines for the comparison with our method. The queries used were:

1. the automatically extracted terms from the title field (run `LucyTitle`);
2. the automatically extracted terms from the description field by a NLP method (run `LucyDesc`);
3. the manually extracted terms from the title and description fields (run `LucyLemme`).

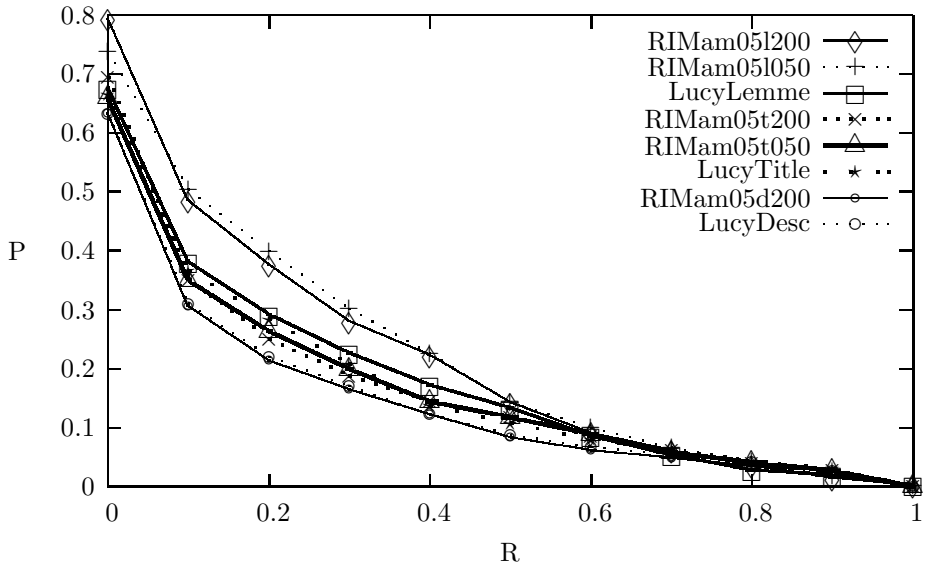

Figure 4: Precision-recall curves for the eight official runs.

The recall precision results are provided in Fig. 4 for all the runs showing that the best performing runs (up to 50% recall) are obtained with our our proximity based method with manually built queries.

The least performing runs are those obtained with automatically built queries with a NLP method on the description field. In fact, these two runs are not distinguishable on the precision-recall curves, even if they are plotted alone as in Fig. 5.

On Fig. 6 it can be seen that the Lucy method performs better than our method with $k = 50$ but our method is better at the first level of recall with $k = 200$, that is to say with a largest area of influence for the term occurences.

Figure 7 displays the precision-recall curves obtained with the manually built queries. The best results in our comparison were obtained with these queries. With these queries, our method performs the best again with $k = 200$ at the lowest levels of recall. But even with $k = 50$ our method performs better than the Lucy one. With the manually built queries, our method retrieves more documents by itself and the Lucy results are not used to supplement our result list up to one thousand documents. So in this case the proximity between query terms is the main factor to select and rank documents. At every recall levels, our method is better than Lucy, but the curves of our method with $k = 50$ and $k = 200$ cross several times.

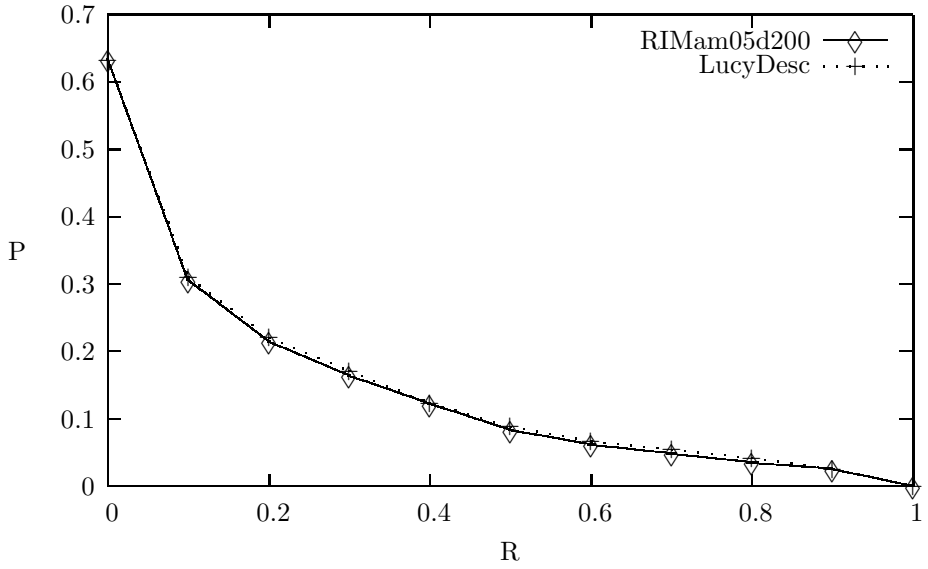

Figure 5: Precision-recall curves for the two runs with the queries built automatically from the description field.

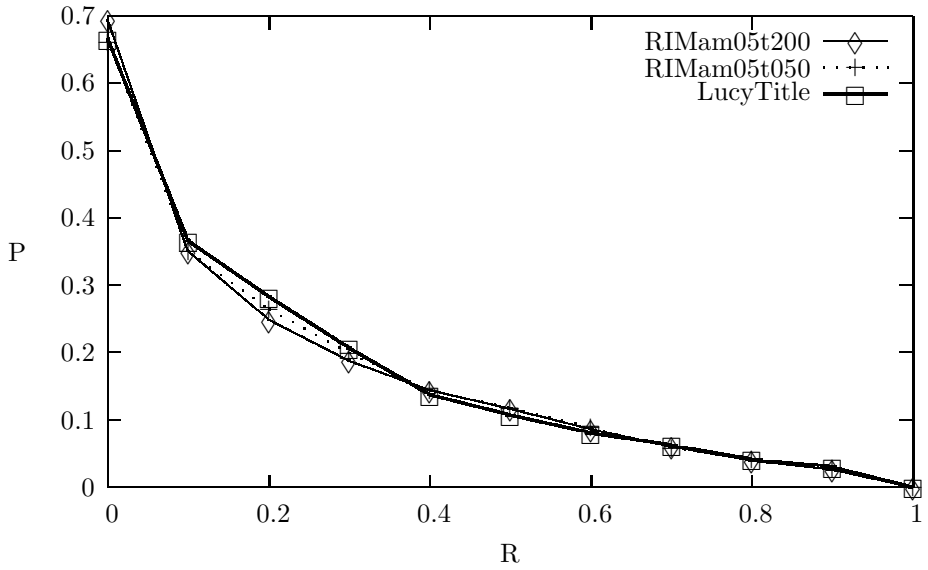

Figure 6: Precision-recall curves for the three runs with the queries built automatically from the title field.

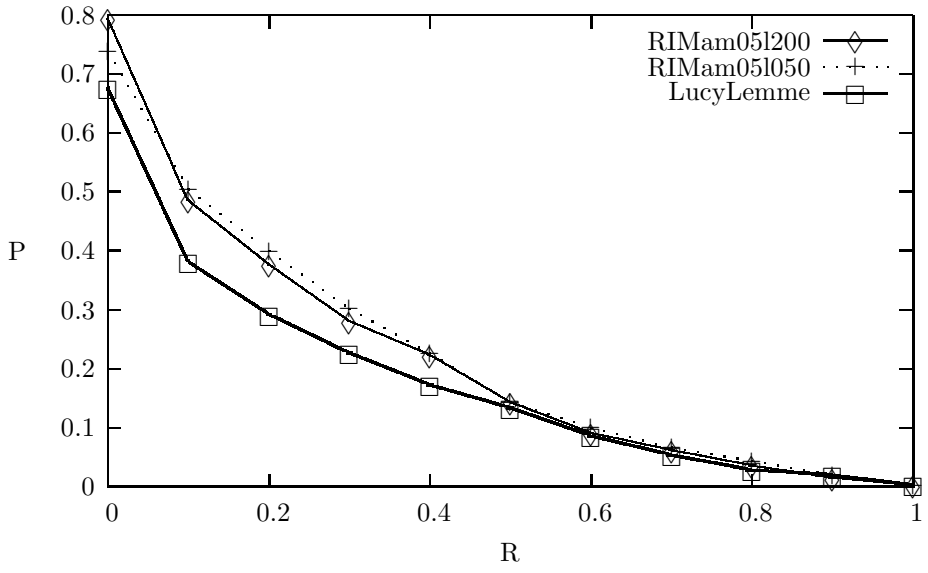

Figure 7: Precision-recall curves for the three runs with the manually built queries.

5 Conclusion

We have presented our information retrieval model which takes into account the position of the term occurences in the documents to compute the relevance scores. We experimented this method on the TREC 2005 Robust Track test collection. We notice that the largest the area of influence of the terms is, the better the results are. In further experiments, we are going to use another influence function more flexible which will allow to dynamically adapt the value of the k constant to the wanted number of retrieved documents. We think also that the results could be improved by using an automatic stemming and evenly a thesaurus in order to retrieve more documents with our method.

References

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[2] Charles L. A. Clarke, Gordon V. Cormack, and Elizabeth A. Tudhope. Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2):291–311, 2000.

[3] Owen de Kretser and Alistair Moffat. Effective document presentation with a locality-based similarity heuristic. In *SIGIR '99: Proceedings of*

the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 113–120. ACM Press, 1999.

[4] D. Hawking and P. Thistlewaite. Proximity operators - so near and yet so far. In D. K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 131–143. Department of Commerce, National Institute of Standards and Technology, 1995.

[5] E. M. Keen. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, 18:89–98, 1992.

[6] Koichi Kise, Markus Junker, Andreas Dengel, and Keinosuke Matsumoto. Passage retrieval based on density distributions of terms and its applications to document retrieval and question answering. In Andreas Dengel, Markus Junker, and Anette Weisbecker, editors, *Reading and Learning: Adaptive Content Recognition*, volume 2956 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2004.

[7] A. Mercier. Etude comparative de trois approches utilisant la proximité entre les termes de la requête pour le calcul des scores des documents. In *INFORSID 2004*, pages 95–106, mai 2004.

[8] Yves Rasolofo and Jacques Savoy. Term proximity scoring for keyword-based retrieval systems. In *25th European Conference on Information Retrieval Research*, number 2633 in LNCS, pages 207–218. Springer, 2003.

[9] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Department of Commerce, National Institute of Standards and Technology, 1994.

[10] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

[11] Xavier Tannier, Jean-Jacques Girardot, and Mihaela Mathieu. Analysing Natural Language Queries at INEX 2004. In Norbert Fuhr, Mounia Lalmas, Saadia Malik, and Zoltàn Szlàvik, editors, *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004*, volume 3493 of *Lecture Notes in Computer Science*, pages 395–409. Springer-Verlag, 2005.

[12] Ross Wilkinson. Effective retrieval of structured documents. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–317. Springer-Verlag New York, 1994.