ENSM-SE at CLEF 2006 : AdHocUses of fuzzy proximity matching function

Annabelle Mercier and Michel Beigbeder
Ecole Nationale Superieure des Mines de Saint Etienne (ENSM-SE)
158 cours Fauriel 42023 Saint Etienne Cedex 2 FRANCE
{annabelle.mercier,michel.beigbeder}@emse.fr

August 20, 2006

Abstract

Starting from the idea that the closer the query terms in a document are to each other the more relevant the document, we propose an information retrieval method that uses the degree of fuzzy proximity of key terms in a document to compute the relevance of the document to the query. Our model handles Boolean queries but, contrary to the traditional extensions of the basic Boolean information retrieval model, does not use a proximity operator explicitly. A single parameter makes it possible to control the proximity degree required. To improve our system we use a stemming algorithm before indexing, we take a specific influence function and we merge fuzzy proximity result list built with different spread of influence function. We explain how we construct the queries and report the results of our experiments in the ad-hoc monolingual French task of the CLEF 2006 evaluation campaign.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Experimentation

Keywords

term proximity, fuzzy information retrieval

1 Introduction

In the information retrieval domain, systems are based on three basic models: the Boolean model, the vector model and the probabilistic model. These models have many variations (extended Boolean models, models based on fuzzy sets theory, generalized vector space model,...) [1]. However, they are all 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 whether it contains a given term or not. In the second case, the system knows the number of occurrences – the *term frequency*, *tf* – of a given term in each document. So whatever the order of the terms in the documents, they share the same index representation if they use the same terms. Noteworthy exceptions to this rule 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 between 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. Different extensions have been proposed to the basic Boolean systems to circumvent this limitation. These extensions represent the documents with some kind of term weights. Most of the time these weights are computed on a *tf* basis. Some combining formulas are then applied to compute the document score given the term weights and the query tree. But these extensions are not compatible with the NEAR operator. Some researchers have thus proposed 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 have been proposed to score documents taking into account different 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 the score for a given interval. The method of Clarke et al. [2] selects the shortest intervals that contain 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 et al. [4], for each query term occurrence, the shortest interval containing all the query terms is selected, thus the selected intervals can nest. Rasolofo et al. [8] chose to select intervals only containing *two* terms of the query, but with the additional constraint that the interval is shorter than five words.

Moreover, passage retrieval methods indirectly use the notion of proximity. In fact, in several methods, documents are ranked by selecting documents which have passages with a high density of query terms, that is to say documents where the query terms are near to each other [11, 3, 6]. The next section presents our method which scores documents on the basis of term proximity.

3 Fuzzy Proximity Matching

To address the problem of scoring the documents 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 model the fuzzy proximity to an occurrence 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. We used an adhoc and a triangular one displayed in figure 1. In the following, 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 extent of influence of each term occurrence. A similar parameter can be found for other shapes.

So, for a query term t, the fuzzy proximity function to the occurrence at position i of the term t is $x \mapsto f(x-i)$. Now, we define the term proximity function w_t^d which models the fuzzy proximity at the position x in the text to the term t by combining the fuzzy proximity functions of the different occurrences 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 positions of the term t in the document d and f is the influence function.

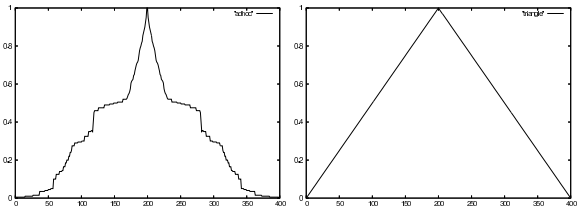

Figure 1: The adhoc and triangular influence functions used in experiments

The query model is 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 fuzzy set theory formulas. So the fuzzy proximity is computed by

$$w^d_{q\,\mathrm{OR}\,q'} = \max(w^d_q, w^d_{q'})$$

for a disjunctive node and by

$$w^d_{q\,\mathrm{AND}\,q'} = \min(w^d_q, w^d_{q'})$$

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

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

Thus, the computed score $s(q, d)$ depends on the fuzzy proximity functions and enables document ranking according to the query term proximity in the documents.

4 Experiments and Evaluation

We carried out experiments within the context of the CLEF 2006 evaluation campaign in the adhoc monolingual French task[1]. We used the retrieval search engine LUCY[2] which is based on the Okapi information retrieval model [9] to index this collection. It was easy to adapt this tool to our method because it keeps the positions of the terms occurring in the documents in the index. Thus, we extended this tool to compute the relevance score values for our fuzzy proximity matching function.

Documents in the CLEF 2006 test collection are newspapers articles in XML format from *SDA* and *Le Monde* of the years 1994 and 1995. For each document (tag `<DOC>`), we keep the fields `<DOCNO>` with the tag and the document number, the textual contents of the tags `<TX>`, `<LD>`, `<TI>`, `<ST>` for *SDA French* and `<TEXT>`, `<LEAD1>`, `<TITLE>` for *Le Monde* 1995. We used the topics and the relevance judgements to evaluate the different methods by the `trec_eval` program.

4.1 Building the Queries

Each topic is composed of three tags: `<FR-title>`, `<FR-desc>`, `<FR-narr>`. Two sets of queries were built for our experiments.

[1] http://clef.isti.cnr.it/

[2] http://www.seg.rmit.edu.au/lucy/

Automatically built queries. For this set, a query is built with the terms from the title field where the stop words[3] are removed. Here is an example with the topic #278. The original topic is expressed by:

```
<top>
<num> 278 </num>
<FR-title> Les moyens de transport pour handicapés</FR-title>
<FR-desc> A quels problèmes doivent faire face les personnes
handicapées physiques lorsquelles empruntent les transports
publics et quelles solutions sont proposées ou adoptées?
</FR-desc>
<FR-narr> Les documents pertinents devront décrire les
difficultés auxquelles doivent faire face les personnes
diminuées physiquement lorsquelles utilisent les transports
publics et/ou traiter des progrès accomplis pour résoudre ces
problèmes.
</FR-narr>
</top>
```

First, the topic number and the title field are extracted and concatenated:

```
278 moyens transport handicapés
```

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

Lucy: `278 moyens transport handicapés`
conjunctive fuzzy proximity: `278 moyens & transport & handicapés`
disjunctive fuzzy proximity: `278 moyens | transport | handicapés`

Manually built queries. 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.

Queries for the method implemented in the Lucy tool are flat queries composed of different inflectional and/or derivational forms of the terms. Here is an example for topic #278:

fuzzy proximity:
```
278 (moyen | moyens) & (transport | transports)
    & (handicap | handicapé | handicapés)
```
Lucy:
```
278 moyen moyens transport transports
    handicap handicapé handicapés
```

4.2 Building the Result Lists

The Okapi model and our fuzzy method were compared.

In first time, it is well known that the Okapi method gives one of the best performances. However, a previous study showed that 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 clef experiments), then its results list is supplemented by documents from the Okapi result list that have not yet been retrieved by the proximity based method.

In second time, we note in past experiments that the higher the area ($k = 200$ was used) of influence of a term the better the results are. Moreover, we retrieved more documents with fuzzy

[3]Removed stop words: à, aux, au, chez, et, dans, des, de, du, en, la, les, le, par, sur, uns, unes, une, un, d', l'.

proximity with a large spread of influence function. So, we merge for each type of queries (title, description and manual), the results obtained with several k values (k equal to 200, 100, 80, 50, 20, 5).

4.3 Submitted Runs

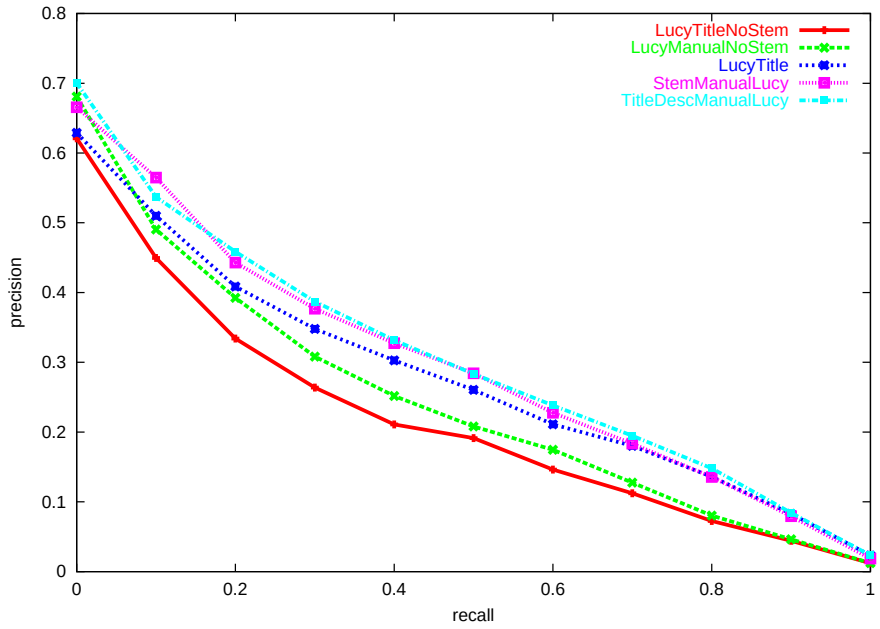

Figure 2: Okapi Lucy - Result with stemming at indexing (LucyTitle, StemManualLucy, TitleDescManualLucy) and no stemming (LucyTitleNoStem, LucyManualNoStem).

In the official runs, the queries used with fuzzy proximity method and adhoc influence function were:

1. the conjunction of the terms automatically extracted from the title field in run `RIMAM06TL`;
2. the conjunction of the terms automatically extracted from the description field in run `RIMAM06TDNL` and;
3. manually built queries with terms from the three fields run `RIMAM06TDML`.

The run `RIMAM06TDMLRef` use the triangular influence function.

Here we present with CLEF 2005 queries the differences obtained between runs with stemming (or not) at indexing step or at query formulation step.

For the runs where the Okapi method was used, the queries are flat (bag of terms). These runs were produced by using the native Lucy search engine and they provide the baselines for the comparison with our method. The recall precision results are provided in figures 2.

We can see that the runs with no stemming step before indexing have less precision than the others. The figure 3 shows also the stemming step provide better results. But we can see that the run TitleDescManual is above the ConjTitle100 which means that the words added for "stemming" queries increase the precision results. The last figure 4 the differences between Okapi Lucy method and fuzzy proximity method. We can see that our method is better than Lucy one (ManualNoStem200 vs. LucyTitleNoStem; StemManual200 vs. ConjTitle200). We can note the run with stemming at indexing and at query time is the better one.

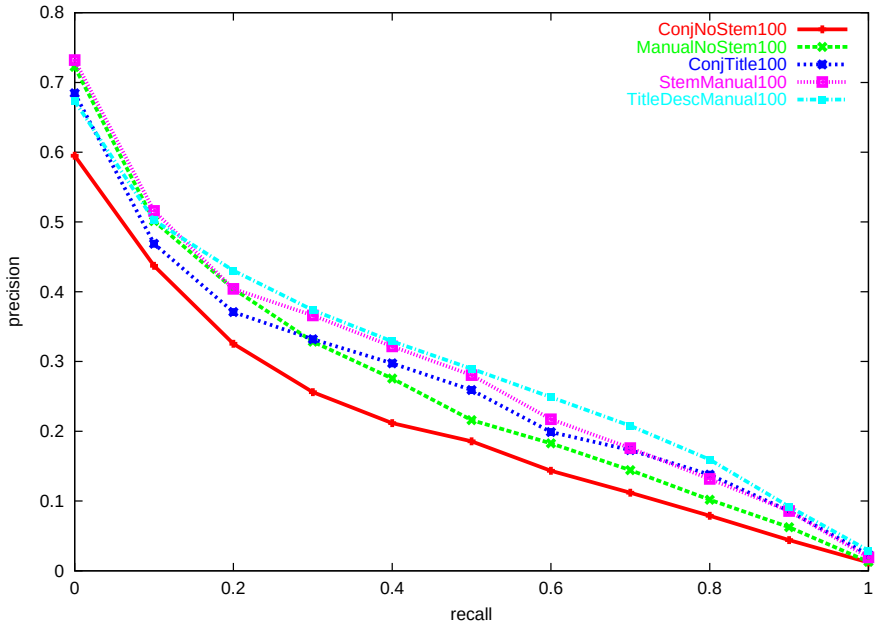

Figure 3: Fuzzy Proximity with $k = 100$ - Result with stemming at indexing (ConjTitle100, StemManual100, TitleDescManual100) and no stemming (ConjNoStem100, ManualNoStem100).

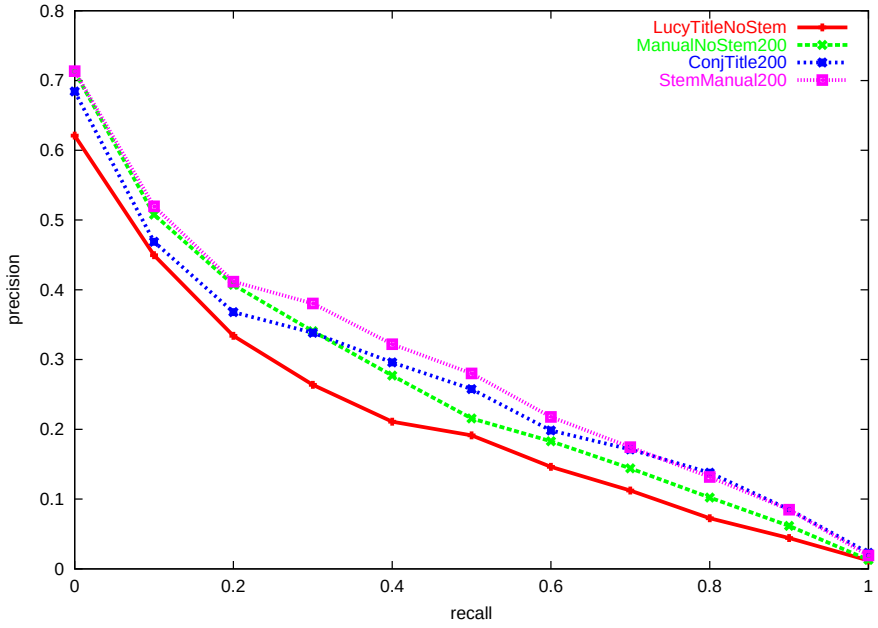

Figure 4: Fuzzy Proximity with $k = 100$ - Result with stemming at indexing (ConjTitle200, StemManual200) and no stemming (LucyTitleNoStem, ManualNoStem200).

5 Conclusion

We have presented our information retrieval model which takes into account the position of the query terms in the documents to compute the relevance scores. We experimented this method on the CLEF 2006 Ad-Hoc French test collection.

In these experiment, we submit runs which use different k values in order to retrieve more documents with our method. We think also that the results could be improved by using an automatic stemming procedure. Eventually we think that the user can add query word taken within a thesaurus in order to retrieve more documents with our method. In further experiments, we will use other function at AND and OR nodes and we want to specifies "phrases" in the queries.

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 ACM SIGIR Annual International Conference on Research and Development in Information Retrieval*, pages 113–120. ACM, 9 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. Étude 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] 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, 7 1994.