Query Expansion and Classification of Retrieved Documents

C. de Loupy[1,2], P. Bellot[1], M. El-Bèze[1] and P.-F. Marteau[2]

(1) Laboratoire d'Informatique d'Avignon (LIA)
339 ch. des Meinajaries, BP 1228
F-84911 Avignon Cedex 9 (France)
{patrice.bellot,marc.elbeze}@lia.univ-avignon.fr

(2) Bertin & Cie
Z.I. des Gatines - B.P. 3
F-78373 Plaisir cedex
{deloupy,marteau}@boston.bertin.fr

Abstract: *This paper presents different methods tested by the University of Avignon and Bertin at the TREC-7 evaluation. A first section describes several methodologies used for query expansion: synonymy and stemming. Relevance feedback is applied both to the TIPSTER corpora and Internet documents. In a second section, we describe a classification algorithm based on hierarchical and clustering methods. This algorithm improves results given by any Information Retrieval system (that retrieves a list of documents from a query) and helps the users by automatically providing a structured document map from the set of retrieved documents. Lastly, we present the first results obtained with TREC-6 and TREC-7 corpora and queries by using this algorithm.*

keywords: ad-hoc information retrieval, automatic relevance feedback, synonymy, automatic classification, cluster-based and hierarchical methods.

1. Introduction

Our first goal in TREC-7 was to measure the performances of an Information Retrieval (I.R.) system, and the improvements brought by different methodologies. The basic tool of this system uses a Part Of Speech (POS) tagger and a lemmatizer[1]. Several modules have been tested: query enrichment techniques using stems or synonyms and two automatic relevance feedback methods: one using the TIPSTER corpus and the other using the World Wide Web.

The second goal was to evaluate a classification algorithm based on hierarchical and clustering methods. It is applied to the set of documents retrieved — and not to the collection as a whole — by using statistical techniques. Its purpose is to improve results given by any Information Retrieval system (that retrieves a list of documents from a query) and to help the users by automatically providing a structured document map from the set of retrieved documents.

This is our first participation in TREC evaluation. A lot of work had to be done and most of our efforts have been devoted to tune our system, so that not enough time was left for thorough testing. Nevertheless, among several achievements, it is possible to point out that some experiments have been very conclusive, as it will be reported in sections 2.5 and in 3.2. We have chosen to participate in the *ad hoc* task, using title or short queries.

2. Query expansion

It is well established that search procedures based only on words contained in a query cannot achieve high scores in Document Retrieval (DR) tasks. Indeed, one must deal with polysemy, synonymy. Furthermore, it is very important to identify the most relevant terms of the domain the query is referring to.

2.1. The basic methodology

The different words (obtained from the query or enrichment or relevance feedback procedures) are combined using fuzzy operators. The importance (information quantity) of a lemma λ depends on its frequency of occurrence [Maarek, 1991]:

$$I(\lambda) = -\log_2\left(P(\lambda)\right) \text{ with } P(\lambda) = \frac{K(\lambda)}{\sum_{\lambda'} K(\lambda')} \text{ where } K(\lambda) \text{ is}$$

the number of occurrences of λ in the corpus.

The quantity of information associated with a document D is then defined as:

$$I(D) = \sum_{\lambda \in D} \alpha(D, \lambda).I(\lambda)$$

where $\alpha(D, \lambda)$ is in fact a coefficient dependent on the size of the document and the number of occurrences of the lemma in this document.

[1] We use ECSTA, the Part of Speech tagger developed at the LIA[Spriet & El-Bèze, 1997]. Lemmatization is provided by lexical access.

The similarity between a document D and a query Q is:

$$S(D,Q) = \frac{I(D,Q)}{I(Q)} = \frac{\sum_{\lambda \in D \cap Q} \alpha(D,\lambda).I(\lambda)}{\sum_{\lambda \in Q} \alpha(Q,\lambda).I(\lambda)} \leq 1$$

The denominator is a constant and is used only for the normalization, it can be eliminated.

2.2. Enrichment

A first method to expand a query is to consider each of its lemmas independently, and to search for associated words: synonyms or words having the same stem. These expansions have been used to enrich *title* and *short* queries.

2.2.1. When and how to use associated words?

Query expansion with synonyms or stems must be handled carefully. Expanding very frequent lemmas could be dangerous. There is always a risk of a bad equivalence when expanding a lemma with its associated lemmas (that is to say synonyms or lemmas having the same stem). And the more frequent is the lemma, the more important are the consequences of an error in such equivalence. Therefore, words associated with a lemma are not taken into account if the latter appears in more than 5 % of the documents in the collection.

Moreover, polysemy is one of the most difficult problems in IR. Synonyms are related to the sense of the word and not to the word itself. If the sense of the word is not known, one will consider all the synonyms corresponding to its different senses as equivalent. Some experiments not reported here have shown that the precision would be very low in such a situation. Unfortunately, it is very difficult to determine the sense of a word in context. We are experimenting different methods to affect sense to words in context [Loupy et al., 1998b – Loupy et al., 1998c]. But since the different methods are not yet validated through reliable assessment, we prefer not to use them in an IR task. This validation is in progress within the SENSEVAL project [Kilgarriff, 1998]. Since a Word Sense Disambiguation (WSD) tool is not yet included in the IR tool, the system verifies the number of possible senses for one term, before deciding to expand it. If the word has no more than two senses, it is expanded with its synonyms.

At last, even if two lemmas can be highly related by the way of synonymy, a document containing the words of the query should be considered more relevant than documents containing their synonyms. In order to give more importance to the lemmas of the query, a coefficient δ ($\delta < 1$) is applied to the information quantity of the associated lemmas.

2.2.2. Stemming

Stemming can be very useful to expand queries, particularly to alleviate lacks in the lexicon. But stemming is not based on a high-level linguistic knowledge and, in many cases, the confusion involved by the use of stems leads to spurious effects. A guesser could solve some of the problems of Out Of Vocabulary (OOV) words. But there was not enough time to extend to English the one developed at the University of Avignon [Spriet et al., 1996] for French. Moreover, stemming can find out very close (semantically) words. In the experiments described in this paper, the Porter's stemmer[2] has been used [Porter, 1980].

For instance, consider the request 317 (TREC 6): "*Unsolicited Faxes"*. The tagger finds that *unsolicited* is an adjective and *faxes* is an inflected form of the noun *fax*. The stemmer links the noun *fax* with the verb *to fax* and the OOV word *megafax* (!).

2.2.3. Synonymy

Many experiments were done with WordNet [Miller et al., 1993] to cope with the synonymy phenomenon (see for instance [Voorhees, 1993]). We have chosen to use this thesaurus. We have also done several experiments taking advantage from the hyponymy relation given in WordNet. But if this led to a slight improvement in the average precision/recall curve for all the queries in TREC-6, it was a real disaster for some requests. The reason is that the depth of the semantic inheritance tree can be too important. Therefore, only synonyms have been used.

The use of synonymy enrichment, for the request 317, linked *unsolicited* with the adjective *undesired* and *fax* with the noun *facsimile.*

2.3. Lexical affinities

A lexical affinity (LA) [Maarek, 1991] represents the affinity between *n* lemmas in a given corpus, that is to say, if they often appear in a near context. The system considers a context of 5 lemmas (content words) before and after a given lemma. According to [Martin et al., 1983 - cited in Maarek, 1991], 98 % of the lexical relations relate words contained in a 11 lemmas window. The LAs considered by the system relate 2 or 3 lemmas. The goal is to take into account the adjacency between the words of the request within the document.

First, the query is analyzed in order to extract the lexical affinities it contains. Let A a LA of the query *Q*. We can define a quantity of information for A and a similarity

[2] An implementation in C of the Porter's algorithm is available at http://www.cs.jhu.edu/~weiss/ir.html

between a document and a query using only LAs as done in 2.1 for lemmas. The similarity using single lemmas (S_{lem}) and the one using LAs (S_{LA}) are combined:

$S(D,Q) = \beta.S_{lem}(D,Q) + (1-\beta).S_{LA}(D,Q)$ where β is a coefficient ($\beta \leq 1$). We have used the empirical value $\beta = 0.7$.

In fact, when the query is expanded with synonyms or stems, an associated word is considered as the lemma itself (equivalence) for the construction of the LAs.

2.4. Automatic relevance feedback

Relevance feedback is a very classical way to automatically expand queries. Several methods are possible. The implemented methods search for documents containing all the terms of the query. Consequently, it is not possible to apply them on the so-called 'short' queries because they are too long. And, since we did not have time to develop a specific method, relevance feedback was used only with 'title'.

2.4.1. Relevance feedback using TIPSTER corpus

To get relevant terms, the texts containing all the lemmas of the query are analyzed in order to get the words appearing in the near context (a 10-word window) of these lemmas within the texts of the TIPSTER corpus. The list of all these words is arranged according to the number of times they were seen. Finally, in order to keep the ones that do not occur a lot of times in the corpus, but often in the context of the query terms, empirical thresholds have been used.

For example the enrichment processing from the documents containing *fax* and *unsolicited* returns the following lemmas: *advertisement*, *mail*, *machine*, *junk*, *ban*, *firm*, *ad*, which do not appear in the query.

2.4.2. Relevance feedback using the World Wide Web

As mentioned in the previous paragraph, it is useful to search for relevant terms in the context of query terms. We only tried such a method on the TIPSTER corpus, which is the textual database to search in for relevant texts. But the size of this corpus (2 Gb) is not very large compared to the amount of texts available on the Internet. Then, the second method used to enrich the query consists in searching for relevant lemmas in texts found in the World Wide Web. But, if the advantage of the WWW is the great amount of available information, this wealth also constitutes a serious drawback. We cannot use the same method to retrieve texts on the Web than the one applied on the TIPSTER corpus.

We have chosen to consider the query not as a set of words or lemmas, but as an indivisible entity. Pattern-matching retrieval has been used. Therefore the texts retrieved from the web with a TREC query contained 'exactly' this query[3]. The number of retrieved texts is not too large and the probability of their relevance is relatively high. For instance, considering the request 301, the retrieved texts contain exactly the **string**: "international organized crime". Of course, in this way, a lot (and even most) of pertinent documents are left aside, but what we need is to avoid unrelevant ones.

The method used to get the lemmas for the enrichment from the document is the same than in 2.4.1. But in this case, the validation process is not based on the number of occurrences. It is important to verify that these new lemmas, retrieved from every kind of sources, have real affinities with the words of the query. Hence, the context of these terms in the texts retrieved from the WWW is analyzed and a word is kept if one of the lemmas occurring in the query appears frequently enough in its context.

If we consider request "*unsolicited fax*", the following lemmas are taken into account to enrich the query: *sender*, *calling*, *bell*, *gt*, *e-mail*, *illegal*, *voice*, *spam*, *phone*, *machine*, *facsimile*, *email*, *check*, *anyone*.

2.5. Combining methods

In order to improve results, the different methods are combined. Table 1 hereafter shows the performances of the different modules. The first column indicates the method, the second one the number of relevant document retrieved (for all the queries), the third is the first point of the curve recall/precision, the fourth is the average precision (A-prec) and the last the R-precision (R-Prec).

	Rel.	**at 0.10**	**A-Prec**	**R-Prec**
Basic	1884	0.4029	0.1739	0.2256
Stems	2033	0.3925	0.1951	0.2381
Synonyms	1876	0.4060	0.1742	0.2269
Syn+Stems	2034	0.4140	0.2115	0.2535
L.As.	2006	0.4208	0.2165	0.2585
Rel. Feed. WWW	1920	0.3883	0.1834	0.2309
Rel. Feed. TIPSTER	1933	0.4105	0.1845	0.2308
Rel. Feed. (2 modules)	1937	0.4034	0.1880	0.2358
all modules (except LAs)	2078	0.4220	0.2188	0.2593

Table 1: Scores of the different modules

These figures show that, if each module can, more or less, improves some scores, the combination of several methods

[3] In fact, some characters are not submitted (like parenthesis) and others are replaced by 'and' or 'or' (like '/').

is the best way to increase both recall and precision. For example, on the one hand, the use of synonyms slightly improves precision (0.4060 at 0.10), but does not lead to a gain in recall (1876 relevant documents retrieved). On the other hand, stemming decreases precision (0.3925 at 0.10) but highly improves recall (2033 relevant documents retrieved). The combination of stems and synonyms clearly improves both precision (0.4140 at 0.10) and recall (2034 documents retrieved).

The following figure gives the recall/precision curves for the basic method (B.), the enrichment by associated words (L.) corresponding with Syn+Stem in Table 1, the relevance feedback method with both modules (R.F.) and all the modules together.

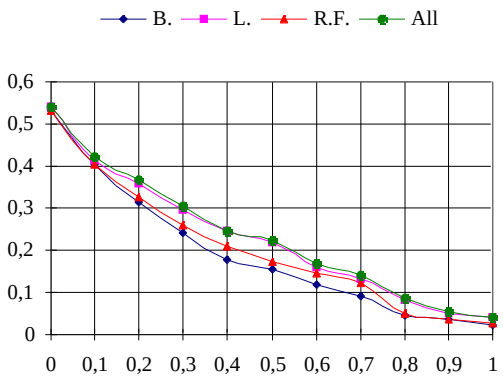

Figure 1: Recall/Precision curves

One can see that the most important improvement comes from synonymy and stemming enrichment procedures. But, although relevance feedback does not lead to a great improvement, the methodology used to get the relevant lemmas could be improved: we only take into account lemmas in the context of the query. Many other techniques could be used.

3. Clustering of retrieved documents

Classification algorithms have been used already in IR to improve the efficiency and effectiveness of retrieval by classifying the documents of a target corpus [VanRijsbergen,1979- Salton,1989- Rasmussen,1992]. A classical information retrieval system retrieves and ranks documents extracted from a corpus according to the computation of distances between the texts and a user query. The answer list is often so long that users cannot examine all the documents retrieved whereas some relevant items are badly ranked and thus never retrieved. In order to solve this problem, we have chosen to automatically cluster retrieved documents according to their topics. Indeed, one assumes that relevant documents are close just like in a relevance feedback scheme one thinks that a relevant document will help to retrieve the other ones (the "Cluster Hypothesis" [Van Rijsbergen,1979]). We present an algorithm combining hierarchical classification and cluster-based (K-means like) methods. They are applied to the set of documents retrieved — and not to the collection as a whole — by using statistical techniques. Hence, the classification is sensitive to the content of the queries. One can summarize this process of information retrieval as shown in Figure 2.

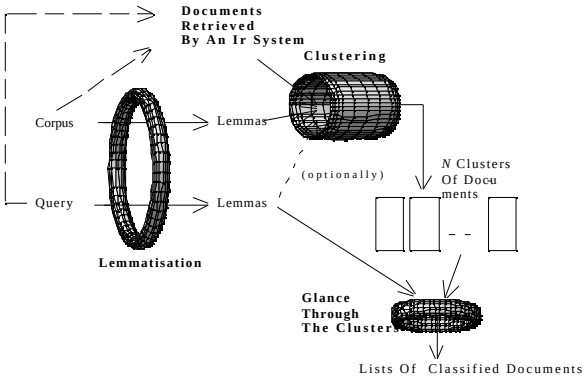

Figure 2 - Classification of documents retrieved

3.1. A Clustering algorithm with hierarchical and cluster-based aspects

An important criteria of an IR system is the time a user has to wait for an answer. Thus, we have chosen to use a K-means like method [Diday, 1982] to cluster the retrieved documents, in particular because its time and storage requirements are much lower than those required by hierarchical algorithms.

This algorithm aims to cluster items as follows:

- Find an initial partition (see 3.1.1)
- Do:
  1. Compute representatives of each existing cluster[4];
  2. assign each document to the most similar cluster.

  while a clustering quality criterion increases or until there is little or no change in cluster membership.

Since the cluster's representatives are computed only at the beginning of an iteration, the cluster's memberships are order independent.

Optionally, a document could be placed in a cluster only if the distance between the document and the cluster representatives does not exceed a given threshold[5].

[4] The number of clusters must be initially chosen.

Since this process may be seen as post-processing, it can be used with any IR-system which returns a list of documents to a user query. This method may only use the set of retrieved documents. Optionally, the knowledge of cumulative frequencies of each lemma in the corpus (and not only in the retrieved documents) improves the quality of the classification. Lastly, we can use queries in order to rank clusters.

3.1.1. Class representatives

It is required to compute centroids of each cluster so that the distances between a cluster and a document or between a cluster and a query can be calculated to allocate documents to their most similar cluster and to be able to rank clusters.

We choose to represent a cluster by the k documents[6] which are the closest to the geometric centre: we compute, for each document, the sum of distances from it to other texts belonging to the same cluster and choose the k documents corresponding to the k smallest distances as representatives.

3.1.2. Initial partition

Since the result of this cluster-based method depends on the initial set, the first problem one faces is to decide how to obtain a valid initial partition. A randomly attribution of documents in clusters is a simple idea but not a good one because clusters are consequently close, their representations similar and the number of iterations of the algorithm before convergence is too large.

We have made several experiments by using different methods [Bellot & El-Bèze,1998b] which will not be reported here. During our first tests on some TREC-5 corpora and queries [Bellot & El-Bèze, 1998a] we have used, among others, a partial hierarchical method to obtain the initial partition and thus strongly improved the quality of the final classification.

In order to obtain the initial classification, we do:

(a) for each couple of documents i and j such that $d(i,j) < threshold$[7]:
   - if i and j are not yet in a class, then create a new one;
   - if i and/or j are already assigned, merge all the documents of the class containing i (resp. j) with those containing j (resp. i);

(b) after this first step, the number of classes created may be greater than the preset number of clusters. So, while the number of classes is greater than the predefined one:
   - compute class representatives;
   - compute distance between every pair of classes (triangular matrix);
   - merge the two closest classes.

3.1.3. Distances

In order to be able to measure the quality of the partition and to assure convergence of the classification process, we must have a *real* distance (satisfying the triangular inequality). That is the case of the so-called *MinMax* distance described hereafter.

Let R and D be two documents, u a lemma and its syntactical tag, $N(u)$ the number of documents containing u in the corpus as a whole and not only in retrieved documents.

The *information quantity* of a lemma is based on its occurrences within the corpus:

$$I(u) = -\log_2 \frac{N(u)}{\sum_{u' \in \text{Corpus}} N(u')}$$

The information quantity of a document is the sum of information quantities of its lemmas:

$$I(D) = \sum_{u \in D} I(u)$$

We assume that the greater the information quantity of the intersection of lemmas of two documents is, the closer they are.

Let the distance between two documents R and D be:

$$d(R,D) = 1 - \frac{I(R \cap D)}{\text{Max}\left(I(R), I(D)\right)}$$

Let k be the number of representatives of a cluster C. Let $D_i \ (1 \le i \le k)$ be a representative[8] of C.

Let the distance between a document and a cluster be:

$$d(R,C) = \min_{1 \le i \le k} \left(d(R, D_i) \right)$$

[5] Tried values range from the average 'document to document' distance to 1 (to exclude documents which have not a common word with the cluster representatives).

[6] So far we have empirically selected $k = 3$.

[7] We choose the threshold value so as the quantity of documents assigned at the end of step '(a).' is greater than half the total number of documents.

[8] Let us recall that a representative of a cluster is a subset of documents.

In order to provide to the user a ranked list of documents from the partition or an arranged view of the clusters, we have to be able to compute distances between a cluster and a query ("what is the cluster which is the closest to the query ?"). This is accomplished using the above distance (*R* a query).

We have also to estimate what are the documents which are the closest to the query in each cluster so that one can rank them. This can be achieved using the similarity or distance values given by the IR system.

3.2. Experiments with TREC corpora and queries

We have assumed that a good classification allows to cluster documents according to their themes. If a query has only one theme, we should consider the best ranked cluster which should contain the relevant documents. But what to do if a query covers several topics ? We could look at the best ranked documents of each cluster *i.e.* at the documents for each theme which are the closest to the query or, merely at each cluster according to its rank. Lastly, we can present each cluster to the user so as he/she chooses those containing the largest number of relevant documents. In order to evaluate the classification process without taking into account the ranking process of clusters, we use the list of relevant documents (the *qrels* file) supplied by the TREC organization and select the best clusters for each query according to the number of relevant documents they contain (see [Hearst & Pedersen, 1996] and [Silverstein & Pedersen, 1997]).

We have chosen to assign a document to a cluster only if the distance between them is lower than a empirical threshold and to group together all remaining documents in a new cluster at the end of the process. Moreover, the documents in each cluster are ranked according to the similarity values between them and the queries.

3.2.1. TREC-6

By using the TREC-6 corpora and queries (from 301 to 350) and by classifying the 1000 best ranked documents retrieved by the "SynStem + LAs" method (cf. 2.5) for each query, we have been able to obtain some better results with classification rather than without it. We should be able to get better ones by choosing different parameters and by modifying the similarities used.

The graph and the table printed below show the recall-precision curves and results obtained:

(a) without classification ("SynStem + LAs" method alone);

(b) with 2 clusters ranked for each query according to the similarity *d* defined here;

(c) with 2 clusters ranked for each query according to the number of relevant documents they contain;

(d) with 8 clusters ranked for each query according to the number of relevant documents they contain.

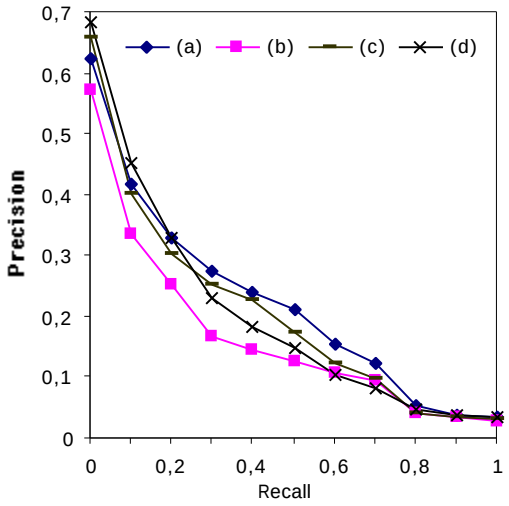

Figure 3 - Results with TREC-6 queries

	Precision at recall 0.00	Precision at recall 0.10	Precision at 5 docs	Precision at 10 docs
(a)	0,624	0,412	0,412	0,328
(b)	0,574	0,336	0,348	0,296
(c)	0,659	0,402	0,42	0,344
(d)	0,684	0,452	0,432	0,368

Table 2: Results with TREC-6 queries and corpora

By making a choice of 8 clusters and by ranking them according to the number of relevant documents they contain, the relative increase of precision for 0.00 and 0.10 recall values is 9.6% and 9.7% —curve (d)—. Precision is 4.8% better at 5 documents and 12.2% at 10 documents. By choosing 2 clusters and by ranking them according to the number of relevant documents they contain —curve (c)—, at 0.00 recall value, the relative increase is equal to 5.6% and at 0.1 recall value, precision is lower than those obtained without classification (-2.4%). However, precision is better: +2% at 5 documents and +4.9% at 10 documents.

Lastly, by choosing 2 clusters and by ranking them according to the similarity *d* defined above —curve (b)—, precision is always lower than without classification[9].

3.2.2. TREC-7

By using the TREC-7 corpora and queries (from 351 to 400) and by classifying the 1000 best ranked documents retrieved for each query, we have obtained better results with classification rather than without it.

Table 3 shows the results obtained:

(a) without classification;

(b) with 2 clusters ranked for each query according to the similarity *d* defined here;

(c) with 2 clusters ranked for each query according to the number of relevant documents they contain.

By choosing 2 clusters and by ranking them according to the number of relevant documents they contain —curve (c)—, the relative increase of precision at 5 documents is equal to 10.5% and at 10 documents, is equal to 6%. Lastly, by choosing 2 clusters and by ranking them according to the similarity *d* defined above —curve (b)—, precision is better with classification rather than without it only at 10 documents (+3%).

	Precision at recall 0.00	Precision at recall 0.10	Precision at 5 docs	Precision at 10 docs
(a)	0.57	0.37	0.38	0.34
(b)	0.58	0.38	0.38	0.35
(c)	0.63	0.35	0.42	0.36

Table 3: Results with TREC-7 queries and corpora

Moreover, for each query, the average ratio of retrieved relevant documents which are in the best class equals 76% (these classes contain 61% of the retrieved documents). That confirms the classification helps to regroup relevant documents.

4. Conclusion

It is clear that we have not reached the roots of all the methodologies described in this paper. Nevertheless, some interesting results have emerged from the different experiments. WordNet and stems can be used effectively together. Using Internet for relevance feedback seems full of promise, since the method we use to get relevant documents and lemmas is very simple. Concerning the classification process, we have obtained some improvements. Choosing different parameters and modifying the similarities used should lead to better results. We have shown that this classification method helps to regroup relevant documents. It increases the effectiveness of retrieval by providing to users a structure of texts and by allowing them a faster examination through the list of retrieved documents. Our participation to the AUPELF-UREF *Amaryllis-2* information retrieval project for the French language will permit to present some new results obtained with our tools.

[9] For recall values from 0.4 to 0.8, one can see that, whatever the classification evaluated, the best results are those obtained without classification. In order to resolve this problem we will try to provide a ranked list of documents which is not the entire succession of clusters contents but the first ones of the first cluster followed by the first ones of the second cluster and so on…

5. References

[Bellot & El-Bèze, 1998a] P. Bellot, M. El-Bèze, "*Classification Automatique et Recherche d'Informa-tion*", Technical Report, IR-06-1998, Laboratoire d'Informatique d'Avignon, 1998.

[Bellot & El-Bèze, 1998b] P. Bellot, M. El-Bèze, "*A Clustering Method for Information Retrieval and Text Segmentation*", to be submitted.

[Diday, 1982] E. Diday, J. Lemaire, J. Pouget, F. Testu, "*Eléments d'Analyse des Données*", Dunod Informatique,1982.

[Frakes, 1992] William B. Frakes, Ricardo Baeza-Yates (Editors), "*Information Retrieval, Data Structures & Algorithms*", Prentice-Hall Inc., 1992, ISBN-0-13-463837-9.

[**Hearst & Pedersen, 1996]** Marti A. Hearst, Jan O. Pedersen, "*Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results*", Proceedings of ACM-SIGIR 96, pp.76-84, 1996.

[Kilgarriff, 1998]: A. Kilgarriff; "*SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs*"; in Proceedings of the First International Conference on Language Resources & Evaluation; pp. 1255-1258; Granada, Spain; 28-30 May 1998.

[Loupy *et al*, 1998a] C. de Loupy, P.-F. Marteau & M. El-Bèze; "*Navigating in Unstructured Textual Knowledge Bases*"; in Proceedings of Nîmes'98 - La Lettre de l'IA; pp. 82-85; May 1998.

[Loupy *et al*, 1998b] C. de Loupy, M. El-Bèze & P.-F. Marteau; "*Word Sense Disambiguation using HMM Tagger*"; in Proceedings of the First International Conference on Language Resources & Evaluation; pp. 1255-1258; Granada, Spain; 28-30 May 1998.

[Loupy et al., 1998c] C. de Loupy, M. El-Bèze & P.-F. Marteau; "*WSD based on three short context methods"*; in Proceedings of SENSEVAL Workshop; Herstmonceux Castle, England; 2-4 September 1998.

[Maarek, 1991] Y. S. Maarek; "*Software Library Construction From an IR Perspective"*; SIGIR forum, Fall 1991, 25:2; pp. pp. 8-18; 1991.

[Marteau *et al*,1998] P.-F. Marteau, C. de Loupy, P. Bellot & M. El-Bèze, "Le Traitement Automatique du Langage Naturel Appliqué à l'Intelligence Economique: vers une Architecture "Push-Pull" d'Accès à l'Information", Système & Sécurité, submitted.

[Martin *et al*, 1983]: W.J.R. Martin, B.P.F. Al & P.G.G van Sterkenburg; "*On the processing of a text corpus: from textual data to lexicographic information"*; in R.R.K. Hartmann, editor, *Lexicography: Principles and Practice*; Applied Language Studies Series, Academic Press; London, 1993.

[Miller *et al*, 1993] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross & K. Miller; *Introduction to WordNet: An on-line lexical database*; http://www.cosgi.princeton.edu/~wn; August 1993.

[Porter, 1980]: M. F. Porter; *An algorithm for Suffix Stripping*; Program 14 (3); pp. 130-137; July 1980.

[Rasmussen,1992] Edie Rasmussen, "*Clustering Algorithms*", in [Frakes,1992], pp.419-442, 1992.

[Rissanen *et al*.,1992] Jorma Rissanen, Eric Sven Ristad, "*Unsupervised Classification with Stochastic Complexity*", Proceedings of the US/Japan Conf. on the Frontieres of Statistical Modeling, 1992.

[Salton,1989] G.Salton, "*Automatic Text Processing*", Reading, Mass. Addison-Wesley, 1989.

[Salton *et al*,1994] Gerard Salton, James Allan, Chris Buckley, "*Automatic Structuring and Retrieval of Large Text Files"*, Communications of the ACM, Vol.37, No.2, 1994.

[Silverstein & Pedersen, 1997] C. Silverstein, Jan O. Pedersen, "*Almost-Constant-Time Clustering of Arbitrary Corpus Subsets*", Proceedings of ACM-SIGIR 97, p.60-66, 1997.

[Spriet *et al*., 1996] T. Spriet, F. Béchet, M. El-Bèze, C. de Loupy & Liliane Khouri, *Traitement automatique des mots inconnus*; in Proceedings of TALN'96; 1996; Marseille; pp 170-179.

[Spriet & El-Bèze, 1997]: T. Spriet & M. El-Bèze; *Introduction of Rules into a Stochastic Approach for Language Modelling*; in Computational Models for Speech Pattern Processing, NATO ASI Series F, editor K.M. Ponting; 1997.

[VanRijsbergen,1979] C. J. Van Rijsbergen, "*Information Retrieval*", Buttherwords, London, 1979.

[Voorhees, 1993]: Ellen M. Voorhees; *Using WordNet to Disambiguate Word Sense for Text Retrieval*; ACM-SIGIR'93; pp. pp. 171-180; Pittsburg, PA, USA; June 1993.