Minimum Redundancy Cut in Ontologies for Semantic Indexing*

Florian Seydoux and Jean-Cédric Chappelier

School of Computer and Communication Sciences,
École Polytechnique Fédérale de Lausanne (EPFL),
CH-1015 Lausanne, Switzerland
{florian.seydoux, jean-cedric.chappelier}@epfl.ch

Abstract. This paper presents a new method that aims at improving semantic indexing while reducing the number of indexing terms. Indexing terms are determined using a minimum redundancy cut in a hierarchy of conceptual hypernyms provided by an ontology (e.g. *WordNet*, *EDR*). The results of some information retrieval experiments carried out on several standard document collections using the *EDR* ontology are presented, illustrating the benefit of the method.

1 Introduction

The main idea of semantic indexing is to use word senses rather than, or in addition to the words[1] for indexing documents, in order to improve both recall (by handling synonymy) and precision (by handling homonymy and polysemy). However, the related experiments reported in the Information Retrieval (IR) literature lead to contradicting results: some claim that this substitution (or addition), carried out in an automatic way, degrades the performances [1–4]; for others conversely, the gain seems significant [5–9].

Although it seems desirable for document indexing to take a maximum of semantic information into account, the resulting expansion of the data processed could happen to be counter-productive. Indeed, the growth of the number of indexing terms not only increases the processing time, but could also reduce the precision: discriminating documents by using a very large number of indexing terms is a hard task ("*curse of dimensionality*" effect). This problem is not new, and various techniques aiming at reducing the size of the indexing set already exist: filtering by stoplist, part of speech tags, frequencies, or through statistical techniques such as LSI [10] or PLSI [11]. However, most of these techniques are not adapted to the case where an explicit semantic information is available in the form of an ontology, i.e. with some underlying formal – not statistical – structure.

The focus of the work presented here is to use ontologies to create semantic indexing sets of "sensible" sizes. This relates, but from a different point of

* This work was partially supported by the Swiss National Fund for Scientific Research (SNFSR) under grant #200020–103529.

[1] Usually stems or lemmas.

C. Bento, A. Cardoso, and G. Dias (Eds.): EPIA 2005, LNAI 3808, pp. 658–668, 2005.

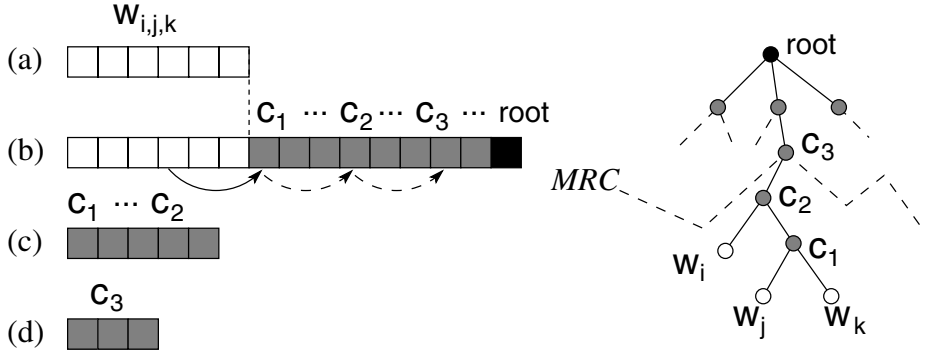


Fig. 1. Different indexing scheme: (a) usual indexing with words (stems or lemmas); (b) using a semantic ontology (displayed on the right), each indexing term is extended with all the concepts that dominate it; this leads to an explosion of the number of indexes for documents; (c) synset (or hypernyms synsets) indexing: each indexing term is replaced with its (hypernyms) synset; this, in principle, can reduce the size of the indexing set since all the indexing terms that are covered by the same hypernym are regrouped in one single indexing feature; (d) Minimum Redundancy Cut (*MRC*) indexing: each indexing term is replaced with its dominating concept defined by *MRC*. This reduces further the size of the indexing set since all the indexing terms that are subsumed by the same concept in the *MRC* are regrouped in one single indexing feature.

view, with experiments reported in [8], [12] or [9], which uses the synsets (or hypernyms synsets [9]) of *WordNet* as indexing terms. Our work goes one step ahead, selecting the indexing set using an information theory based criterion, the *Minimum Redundancy Cut* (*MRC*, see Fig. 1). This criterion is applied to the inclusive "is-a" relation (hypernyms) provided by the *EDR* taxonomy [13].

2 Ontology-Cut Model

2.1 Goals

The choice of the appropriate hypernym (a concept in the ontology) to be used for representing a word is not easy: be it too general, the performances of the system will degrade (lack of precision); be it too specific, the indexing set will not reduce enough, preserving some distinction between words with close senses (lack of recall).

To select the appropriate level of conceptual indexing, we consider cuts in the ontology. A cut is a minimal set[2] of nodes in the ontology defining a coverage of all the leaf nodes (i.e. words). Each node in the cut is used to represent every leaf node it dominates.

The problem is to find a computable strategy to select an optimal cut. For a related task, Li and Abe [14] use the Minimum Description Length principle

[2] By "minimal set" we mean that no node can be removed from the set without decreasing it's coverage (i.e. the number of leaves it dominates).

(MDL). Although easy to compute, this criterion has the drawback in practice (with *EDR*) of often selecting as a cut the root of the ontology, which is not really useful for document indexing. We rather propose to use a new criterion, based on information theory, that selects a cut for which the redundancy is minimal, i.e. a cut where the degree of description of the indexing features in the ontology is as balanced as possible (maximum entropy).

2.2 Minimum Redundancy Cut (*MRC*)

Let $\mathcal{N} = \{n_i\}$ and $\mathcal{W}$ respectively be the set of nodes and the set of words in the ontology. A cut Γ is defined as a minimal subset[2] of $\mathcal{N}$ which covers $\mathcal{W}$. A probabilized cut $M = (\Gamma, P)$ consists of a cut Γ with a probability distribution P on it. From now on, the probabilized cut $M = (\Gamma, P_{\text{tf}})$ is considered, where P_{tf} is defined by the relative frequencies of the words in the collection:

$$P_{\text{tf}}(n_i) = \frac{f(n_i)}{|D|}, \tag{1}$$

$f(n_i)$ being the number of occurrences of the node n_i in a document collection D, and |D| the total number of word occurrences in this collection. To compute $f(n_i)$, we consider that an occurrence of n_i happens whenever some of its hyponym words occurs.

The redundancy $R(M)$ of a probabilized cut $M = (\Gamma, P)$ is defined as [15]:

$$R(M) = 1 - \frac{H(M)}{\log |M|},$$

where $H(M) = -\sum_{n \in \Gamma} P(n) \cdot \log P(n)$ and $|M|$ denotes the number of nodes in the cut Γ.

Minimizing the redundancy is equivalent to maximizing the ratio between the entropy of the cut, $H(M)$, and its maximum possible value, $\log |M|$, i.e. balancing the probabilities of the nodes in the cut so far as it could.

To illustrate *MRC* with a toy example, consider the ontology given in Fig. 2 and an hypothetic dataset for which the frequencies of the words in the ontology are indicated on the same figure. The redundancy of the example cut Γ = [Animal, Plant, Transport] is 0.272:

n	Animal	Plant	Transport
$f(n)$	20	33	2
$P_{\text{tf}}(n)$	0.3704	0.5926	0.0370
$-P_{\text{tf}}(n) \log_2 P_{\text{tf}}(n)$	0.5307	0.4473	0.1761
$R(\Gamma) = 1 - \frac{1.1541}{\log_2(3)} = 0.2718$			

In this example, it can checked by examining each of the 2036 possible cuts that the global *MRC* is the one displayed on Fig. 2. Its redundancy is 0.071. It also corresponds to the local *MRC* found (with only 117 evaluations) by the local search algorithm presented in the next section.

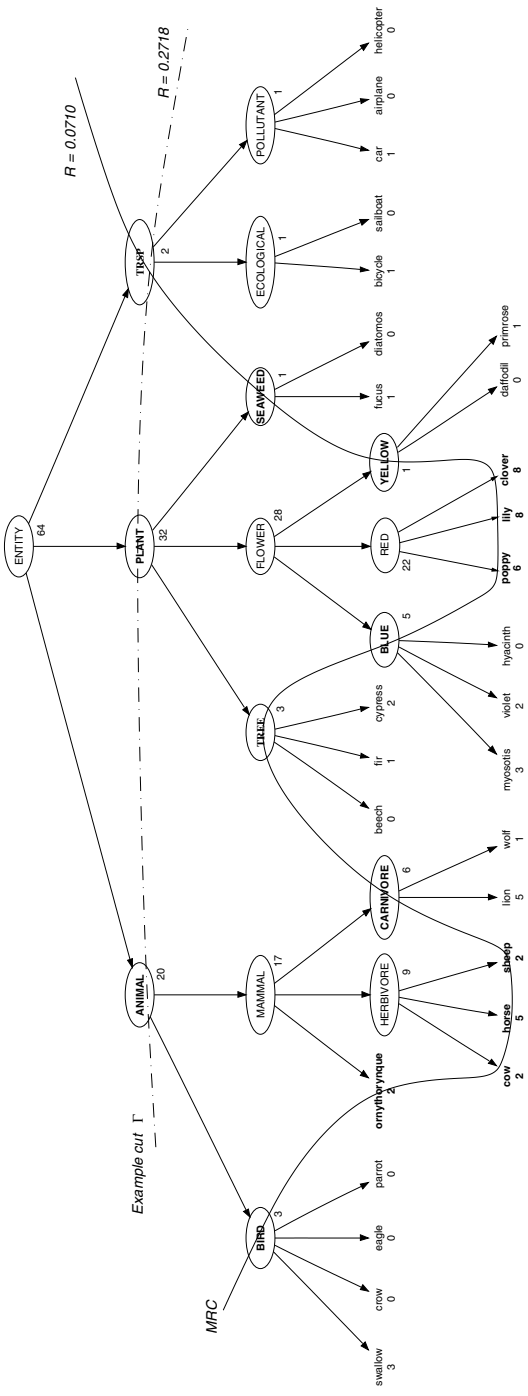


Fig. 2. Example of cuts and their redundancy value

Notice that R does not necessarily have a unique minimum on all possible cuts, but may rather have several equally minimal cuts. In practice, this can easily be overcome, considering for instance any of the minimal cuts or those having a minimal number of nodes, or the minimum average depth of the nodes, etc.

2.3 Finding a *MRC*

In order to identify global *MRC*, the whole set of possible cuts has to be considered. We thus decided to give up global optimality for the sake of tractability and focussed rather on more efficient heuristics.

The algorithm we propose for finding a *MRC* starts from the cut containing all the leaves and iteratively modifies it by systematically choosing the replacement of a node by its parent or its children[3] that minimizes the redundancy. More precisely, for each node n_i in the current cut, we consider on one hand $n_i{}^{\downarrow}$, the (set of) children of n_i, (see Fig. 3a) and on the other hand $n_i{}^{\uparrow}$, the (set of) parents of n_i (see Fig. 3b). The cut with minimal redundancy among these new considered cuts (and the current one) is kept, and the search continues as long as better cuts are found. The full algorithm[4] is given on Fig. 3.

3 Experiments

To evaluate the benefit of the *MRC* indexing method, we carried out several experiments with some of the standard document collections of the SMART [16] system[5] and ontologies generated from the *EDR Electronic Dictionary* [13].

EDR gathers information about approximatively $420,000$ "words" of different types (including compounds and idiomatic expressions); organized into $\approx 490,000$ concepts, with $\approx 500,000$ super/sub relations between them. The two different ontologies provided by *EDR* were used: a very large scale general ontology and a smallest one specialized on information science.

For the evaluation, the vector-space SMART information retrieval system and an external lemmatizer[6] (which also acts as a tokenizer) are used. A filtering based on the POS tag is also carried out (but no stoplist, nor frequency filtering). The new indexing sets are produced while preprocessing the data as follow:

1. First of all, the textual information (title and contents) is aggregated for each document and query; all other informations (authors, sources, etc.) are removed; then, documents and queries are tokenized and lemmatized by the third-party tool[6], and filtered based on their POS tag (name, adjective, verb and adverb are kept).

[3] Due to the DAG structure of the ontology, this replacement can involve more than one node in the cut.

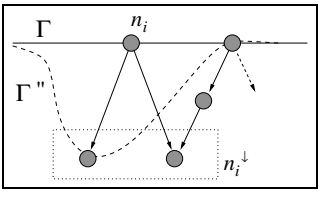
[4] In practice, several optimizations can be made, which do not conceptually change the algorithm and are thus not presented here for the sake of clarity.

[5] Available online at `ftp://ftp.cs.cornell.edu/pub/smart/`.

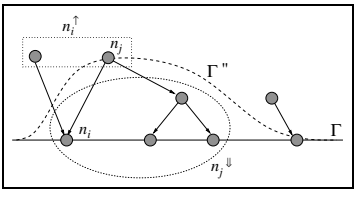
[6] Sylex 1.7, © 1993-98 DECAN INGENIA.

Requires: a hierarchy $\mathcal{N}$ (the leaves of which are $\mathcal{W}$).
Provides: a cut Γ with (local) minimal redundancy.

$\Gamma \leftarrow \mathcal{W}$ # *The current cut. We start from the leaves.*
repeat
  $\Gamma' \leftarrow \varnothing$ # *The new best cut.*
  $\Gamma'' \leftarrow \varnothing$ # *The tested candidate.*
  continue $\leftarrow$ false # *Search-loop control flag.*
  for all $n_i \in \Gamma$ **do**
    # *Evaluate the children's cut:*
    $\Gamma'' \leftarrow (\Gamma \setminus \{n_i\}) \cup \left({n_i}^{\downarrow} \setminus (\Gamma \setminus \{n_i\})^{\Downarrow}\right)$
    $\Gamma' \leftarrow \text{Argmin}\left(R(\Gamma'), R(\Gamma'')\right)$
    # *Evaluate each parent's cut:*
    for all $n_j \in {n_i}^{\uparrow}$ **do**
      $\Gamma'' \leftarrow (\Gamma \cup \{n_j\}) \setminus {n_j}^{\Downarrow}$
      $\Gamma' \leftarrow \text{Argmin}\left(R(\Gamma'), R(\Gamma'')\right)$
  if $R(\Gamma') < R(\Gamma)$ **then**
    $\Gamma \leftarrow \Gamma'$ # *Keep the best cut.*
    continue $\leftarrow$ true # *The search goes on.*
    # *Some watchdog or timer can be put here.*
until continue is false
return Γ



(a) Lower search on ${n_i}^{\downarrow}$: keep only those nodes of ${n_i}^{\downarrow}$ that are not already covered by Γ, i.e. ${n_i}^{\downarrow} \setminus (\Gamma \setminus \{n_i\})^{\Downarrow}$.



(b) Upper search on ${n_i}^{\uparrow}$: if $n_j \in {n_i}^{\uparrow}$ is kept, then remove from Γ the nodes it covers, i.e. those in $(n_j)^{\Downarrow}$.

Fig. 3. *MRC* local search algorithm, where $n^{\Downarrow}$ is the transitive closure of $n^{\downarrow}$, and $R(\varnothing) = R(\{c\}) = 1$, by convention. Sub-figures (a) and (b) illustrate the two local search steps of the algorithm.

2. Then we look for the correspondences between the tokens in a document and the entries (leaves) in the ontology, with the lexical string first and the lemmatized form then, if necessary. Tokens without correspondence in the ontology are indexed in the standard way. The coverage rate[7] of the collections by the ontology was 90% in average.

[7] The coverage rate is the number of different words in the collection that are in the ontology, divided by the total number of different words in the collection.

3. Then the hierarchy of concepts related to the tokens found in the ontology is expanded by:
   (a) in a first set of experiments, selecting all possible senses (relying upon the mutual reinforcement induced by collocations to have a sort of disambiguation);
   (b) in a second run of experiments, selecting only one sense, always the same independently of the context, e.g. the most frequent one[8].
4. An *MRC* cut is then computed with the algorithm previously presented (in practice, we limit the cut only to the nodes covering words contained in the documents, but neither the whole ontology, nor the words in the queries are considered).
5. Finally, the tokens in both documents and queries are substituted by the identifiers of the concepts which subordinate them in the cut determined at the preceding step. Tokens of the queries which do not occur in the documents and are not subordinated by a node of the cut are ignored.

Table 1. Indexing sizes and Mean Average Precision (ntn weighting) of several indexing schemes (see Fig. 1); (a): words only; (b): words + concepts; (b1): same with trivial WSD; (c): direct hypernyms; (c1): same with trivial WSD (d): hypernyms from the *MRC*; (d1): same with trivial WSD.

	(a)	(b)	(b1)	(c)	(c1)	(d)	(d1)
ADI collection (82 documents, 35 topics) [Documents from Information Science]							
$\|index\|$	1800	14664	5254	10080	2891	5516	**1740**
MAP	0.3431	0.2871	0.3840	0.3140	**0.4071**	0.2976	**0.4071**
TIME collection (423 documents, 83 topics) [General world news articles from the Time magazine (1963)]							
$\|index\|$	21815	92117	53142	69354	31583	18949	**18404**
MAP	0.5349	0.3908	0.4999	0.4284	0.5442	**0.5476**	**0.5471**
CACM collection (3204 documents, 64 topics) [Titles and abstracts from a computer science journal]							
$\|index\|$	10053	51712	25208	38524	14696	8410	**8339**
MAP	0.2814	0.1245	0.2324	0.1869	0.2774	0.2532	**0.2950**
MED collection (1033 documents, 30 topics) [Abstracts from a medical journal]							
$\|index\|$	11893	56091	30305	41742	18113	**10206**	10306
MAP	**0.4470**	0.2532	0.3984	0.2704	0.4284	0.3679	0.4226
CISI collection (1460 documents, 112 topics) [Articles from Information science (Library science)]							
$\|index\|$	10019	53453	26278	39544	14998	8404	**8114**
MAP	0.1535	0.0875	0.1413	0.0959	**0.1632**	0.1300	0.1489

[8] In *EDR*, this information is available for each word.

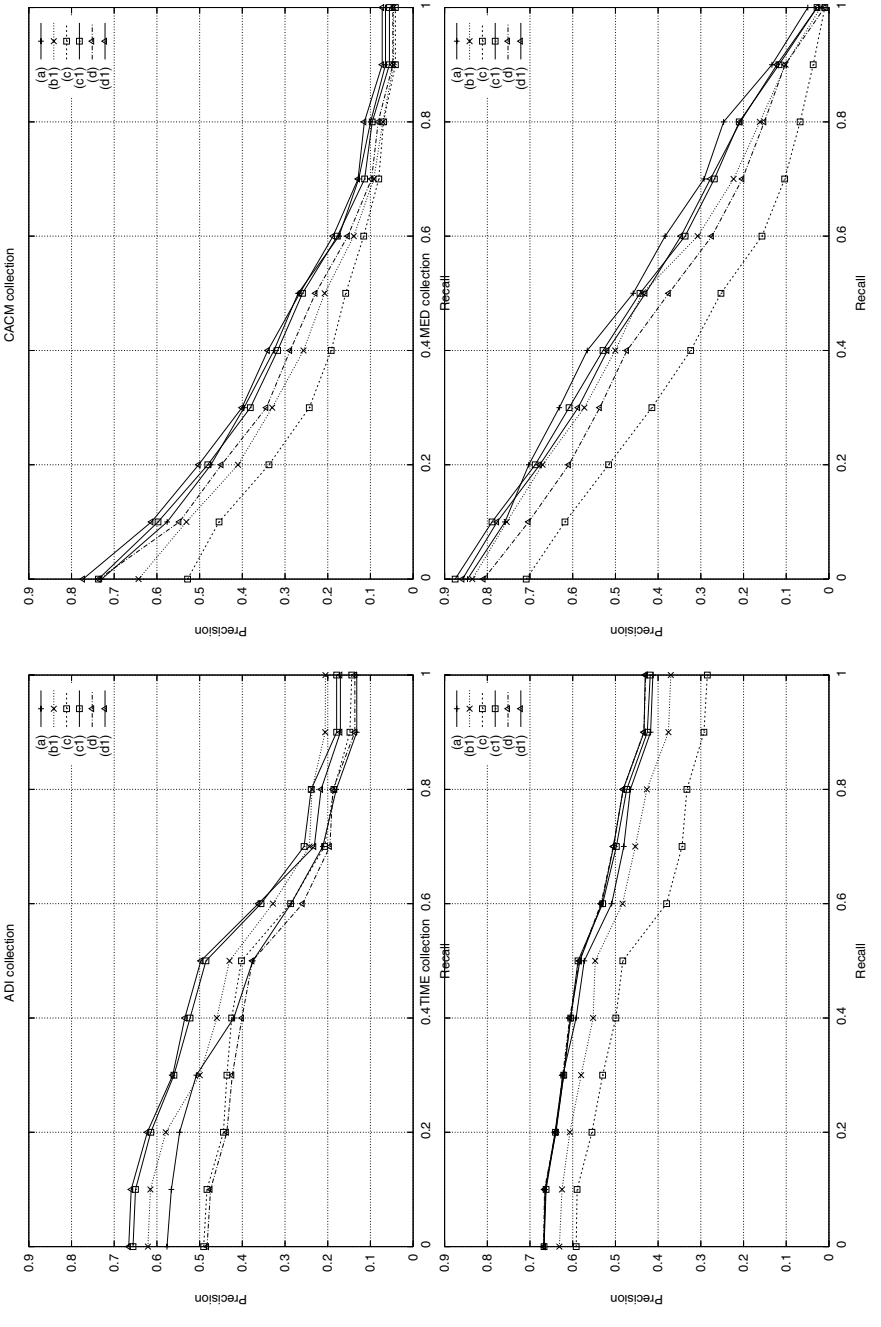


Fig. 4. Precision-Recall curves for several indexing methods (see Fig. 1 and table 1) on the ADI, TIME, CACM and MED collections

Table 1 gathers the index size and Mean Average Precision (MAP)[9] results for the IR experiments carried out. In Fig. 4, we furthermore provide the precision-recall curves for the four most interesting collections.

4 Discussion and Conclusion

Five main conclusions can be drawn out of these experiments:

1. Using *adapted* additional semantic information can enhance the indexing of documents, and thus the performances of a IR system. The results of semantic (ontology-based) indexing (columns (c), (c1), (d) and (d1)) are indeed better than the baseline system for four of the collections, but slightly worse on the MED collection.

   This can be explained by the specificity of the vocabulary of these collections and their adequacy with the semantic resource. ADI and CACM have an important technical vocabulary well covered by the *EDR* ontology, whereas the MED collection has an extremely specific vocabulary, for which the *EDR* ontology is not adapted. Moreover, although the vocabulary of TIME is very general, semantic indexing methods perform well on it. Finally, the CISI collection present documents with a significant number of dates, proper names, etc., for which the POS filtering seems to have annoying consequences; the low performances clearly indicating an initial loss of informations.

   The results presented here confirm those obtained with similar experiments using *WordNet* [17], which has a quite different structure than *EDR*.
2. The overall performance of *MRC* indexing ((d) and (d1)) is better than indexing with direct hypernyms ((c) and (c1)) on ADI, TIME and CACM, both on MAP measure and precision-recall curves. *MRC* is often ahead, with a noticeably smaller indexing set. It furthermore performs better both on specific vocabulary in adequacy with the ontology (ADI, CACM) and on general purpose large documents (TIME), which is a good omen for the future of the method.
3. Semantic disambiguation, even rudimentary, appears to be necessary. Indeed, the simple heuristics consisting in removing semantic ambiguities by choosing, for a given word, always the same of its senses[10] already allows a significant increase in the performances.

   The expected mutual reinforcement of collocations as a kind of "natural" disambiguation does actually not occur. The reason is that the number of

[9] The "Mean Average Precision" is the average precision of all relevant documents for a query, averaged over all queries as given by the `trec_eval` system (`http://trec.nist.gov/trec_eval`).

[10] I.e. doing "Word Sense Disambiguation" independently of the context. Notice that this choice is made prior to the computation of a *MRC*. Thus the *MRC* with WSD is not at all related to the one without. It can even be closer to the leaves, having then more nodes as for instance in (d) vs (d1) on the MED collection.

different senses of ambiguous words tends to give more importance to the most ambiguous terms, i.e. increases the cosine measure between queries and documents with ambiguous word, which may not be relevant.

Anyway, these results urge on the use of a proper WSD procedure for further improving the results.

4. The results obtained on the TIME collection are illustrating the benefit of the method quite well: comparing indexing size and performance increase between columns (c) and (c1) on one hand, and (d) and (d1) on the other, clearly shows that choosing one concept among many is of real importance at the level of synset indexing (c), whereas it as much less impact at the level of higher cuts in the hierarchy (d). This shows that the description level chosen by the *MRC* cut (d) has reached a level of generalization which is good enough for the targeted discrimination task, indexing the documents correctly enough so as not to fall into semantically meaningless (polysemic) details.
5. As a final evaluation, we tried different weighting schemes in `trec_eval`. As expected, schemes including idf were always giving better results. For this reason, we had the idea to evaluate the impact of idf weighting early on the computing of *MRC*, i.e. changing P_{tf} of Eq. 1 for

$$P_{\text{tf.idf}} = \frac{f(n_i)}{|D|} \cdot \log \frac{|d|}{df(n_i)}$$

Although this is no longer a probability distribution on Γ, this weighting gave better results:

		ADI	TIME	CACM	MED	CISI
(d1) P_{tf}	$\|index\|$	1740	18404	8339	10306	8114
	MAP	0.4071	0.5471	0.2950	0.4226	0.1489
(d1) $P_{\text{tf.idf}}$	$\|index\|$	1808	19067	8779	10741	8642
	MAP	**0.4155**	**0.5501**	**0.3018**	**0.4300**	**0.1540**

The conclusion is that weighting plays definitely an important role (this is not new!) and should be closely watched. It would for instance certainly be interesting to confront the results presented here with similar experiments using other weighting schemes, such as Lnu weighting for example, which is known to be more reliable for handling noisy data and dealing with long documents.

As another future work, we also plan to generalize this technique with ontologies modelling thematic relationships between terms, in addition to a hierarchical structure of hypernyms. It would be also interesting to use better WSD procedure.

References

1. Salton, G.: Automatic Information Organization and Retrieval. McGraw-Hill (1968)
2. Harman, D.: Towards interactive query expansion. In: Proc. of the 11th Annual Int. ACM-SIGIR Conference on Research and development in information retrieval. (1988) 321–331
3. Voorhees, E.M.: Using WordNet to disambiguate word senses for text retrieval. In: Proc. of 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. (1993) 171–80
4. Voorhees, E.M.: Using WordNet for text retrieval. In Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press (1998) 285–303
5. Richardson, R., Smeaton, A.F.: Using WordNet in a knowledge-based approach to information retrieval. Technical Report CA-0395, Dublin City University, Glasnevin, Dublin 9, Ireland (1995)
6. Smeaton, A.F., Quigley, I.: Experiments on using semantic distances between words in image caption retrieval. In: Proc. of 19th Int. Conf. on Research and Development in Information Retrieval. (1996) 174–180
7. Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with WordNet synsets can improve text retrieval. In: Proc. of the COLING/ACL 1998 Workshop on Usage of WordNet for Natural Language Processing. (1998) 38–44
8. Gonzalo, J., Verdejo, F., Peters, C., Calzolari, N.: Applying EuroWordNet to multilingual text retrieval. Journal of Computers and the Humanities **32** (1998) 185–207
9. Mihalcea, R., Moldovan, D.: Semantic indexing using WordNet senses. In: Proc. of ACL Workshop on IR & NLP. (2000)
10. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
11. Hofmann, T.: Probabilistic latent semantic indexing. In: proc. of the 22th International Conference on Research and Development in Information Retrieval (SIGIR). (1999) 50–57
12. Whaley, J.M.: An application of word sense disambiguation to information retrieval. Technical Report PCS-TR99-352, Dartmouth College, Computer Science, Hanover, NH (1999)
13. Miyoshi, H., amd M. Kobayashi, K.S., Ogino, T.: An overview of the EDR electronic dictionary and the current status of its utilization. In: Proc. of COLING. (1996) 1090–1093
14. Li, H.: A probabilistic approach to lexical semantic knowledge acquisition and structural disambiguation. Master's thesis, Graduate School of Science, University of Tokyo (1998)
15. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27** (1948) 379–423
16. Salton, G.: The SMART Retrieval System – Experiments in Automatic Document Processing. Prentice Hall (1971)
17. Seydoux, F., Chappelier, J.C.: Semantic indexing using Minimum Redundancy Cut in ontologies. In: Proc. of the International Conference on Recent Advances in Natural Language Processing (RANLP'05), Bulgaria. (2005)