Building domain specific lexical hierarchies from corpora

Olivier Ferret*, Christian Fluhr*, Françoise Rousseau-Hans† and Jean-Luc Simoni

* CEA - LIST
BP 6
18, route du Panorama
92265 Fontenay-aux-Roses Cedex
{olivier.ferret, christian.fluhr}@cea.fr
† CEA – DTI
Saclay
91191 Gif-sur-Yvette Cedex
{françoise.rousseau}@cea.fr

Abstract

In this article, we present a new algorithm for building domain specific lexical hierarchies from texts. The basic elements of such a hierarchy are the normalized terms – mono and multi-word terms – extracted from a large corpus by a terminological extractor. The algorithm relies on collocations for representing the meaning of these terms, finding hierarchical relations between them and finally, organizing them into a hierarchy. Moreover, it takes into account the polysemy of terms while it builds the hierarchy. We also present the results of its application on a part of the corpus designed for the ARC A3 of the Francil network and we go through its possible applications.

1. Introduction

During last years, lexical resources such as WordNet (Miller et al., 1989) or EuroWordNet (Vossen, 1998) proved to be very useful for supporting a large set of tasks related to language engineering. However, building such resources is still a big work as a large part of it is achieved manually. This investment is conceivable for the core vocabulary of a language but cannot be done for each specific domain or even for most of the multi-words of a language.

In this paper, we address one aspect of the problem of automatically building such resources. More precisely, we aim at building lexical hierarchies from corpora in specific domains. The basic elements of such a hierarchy are the normalized terms – mono and multi-word terms – extracted from a corpus by a terminological extractor. The system we present here organizes them by finding semantic relations between them. It focuses more particularly on hierarchical relations comparable to hyperonymy.

The problem of extracting hyperonymy relations between terms was already tackled by several researchers. Two major approaches can be distinguished in this field (Bourigault and Jacquemin, 2000): the first one is based on the identification of linguistic patterns and the second one on statistical criteria.

In the linguistic approach, a set of linguistic patterns that specifically detect the occurrences of the relation to extract, hyperonymy in our case, is built manually from a reference corpus and is then applied for identifying the occurrences of the relation and its arguments in another corpus. For instance, in the sentence "le chat est un félin" ("cats are felines")[1], the hyperonymy relation between is "chat" ("cats") and "félin" ("felines") is found by applying the pattern "NP1 est un NP2" ("NP1 are NP2"), where "NP" states for "noun phrase". This approach is very precise but its productivity is generally low as the linguistic cues on which are based the extraction patterns are not very frequent. For hyperonymy, it is mainly represented by the work of Hearst (1992), Jouis (1993), Morin (1999) and Séguéla and Aussenac (1999).

In the statistical approach, each term is characterized by the set of its occurrences and their contexts, and the relations between terms rely on the proximity of their contexts. Depending on what kind of preprocessing is applied to texts, the building of such contexts is based on simple collocations between terms or on syntactic relations[2]. The characteristics of this approach and those of the first one are complementary: the type of the relations it extracts is more unspecified but its productivity is much higher, provided that the corpus to process is large enough. This second approach is represented by systems such as SEXTANT (Grefenstette, 1994), LEXICLASS (Assadi, 1998) and ZELLIG (Habert et al., 1996).

The work that we present in this article extends (Grefenstette, 1994) and therefore, takes place in the statistical approach. More precisely, it relies on collocations for finding hierarchical relations between terms and organizing them into a hierarchy.

2. Principles

The method for building lexical hierarchies that we describe in this paper is based on a distributionalist viewpoint about the meaning of terms: the meaning of a term *T* in a corpus is characterized by the set of contexts associated to each occurrence of *T* in the corpus. In our case, such a context is made up of the terms that collocate with *T* in a paragraph. We chose paragraph for delimiting the context of a term occurrence because most of the time, this textual unit is topically homogeneous. Hence, its terms are strongly linked from the viewpoint of their meaning. Moreover, this unit is explicitly marked in texts.

[1] As our work was done for French, we give our examples in French and their translation in English in brackets.

[2] Syntactic analysis is a means for selecting collocations whose words are part of the same noun phrase for instance or for selecting collocations between a verb and its subject or its object.

When paragraphs are not tagged or they are not reliable in relation to their topical homogeneity, it is possible to delimit topical segments by using a tool that achieves linear text segmentation, such as *TextTiling* (Hearst, 1997) or *C99* (Choi, 2000) for instance.

More precisely, we represent the meaning of a term in a corpus by the result of the aggregation of all its contexts in this corpus. This representation is called *Semantic Context* (*SC*). The building of a hierarchy of terms is based on the relations between their *SCs* according to the following principle: if a term *T1* is the parent of a term *T2* in the hierarchy of terms, the *SC* of *T2* is included in the *SC* of *T1*.

This principle, that was initially developed in (Barbiéri, 1992) and then in (Simoni, 2000), is implemented by three modules: the first one extracts and selects the most significant terms of a corpus; the second one builds the semantic context of each of the selected terms in order to characterize their meaning in relation to the corpus; finally, the third one organizes the selected terms into a hierarchy. This last module takes into account the polysemy of terms while it builds their hierarchy by splitting the semantic contexts of polysemous terms.

3. Term extraction and selection

The first step of the method we propose consists in extracting a large set of terms, normalizing them and then, selecting the most significant ones. For mono-terms, the extraction and the normalization are achieved by the morpho-syntactic tagger and the lemmatizer associated to the SPIRIT information retrieval system (Fluhr, 1994). For multi-terms, they are performed by a term extractor that classically relies on a chunker and a set of lexico-syntactic patterns (Debili, 1982). The chunker splits sentences into large groups of words that are called chains. More precisely, it distinguishes two kinds of chains: nominal chains and verbal chains. Lexico-syntactic patterns are then used for extracting multi-terms from chains. The term extractor takes as input the mono-terms given by the SPIRIT's tools.

The selection of the most significant terms relies on both their type – only nominal terms are kept – and statistical criteria: a term is selected if both its frequency in the considered corpus and the number of documents in which it is present are high enough. These two criteria are implemented by thresholds whose value was set experimentally. They are parameters of our method.

4. Building semantic contexts of terms

In accordance with the definition of *Semantic Context* (*SC*) we have given in Section 2, the first stage of their building consists in storing for each selected term *T* the set of selected terms that collocate with it in a paragraph. This is done for all paragraphs of the considered corpus in which *T* is present. The *SC* of *T* is the union of all these sets. Each term in a *SC* is associated to its number of occurrences, *i.e.* the number of paragraphs in which it collocates with the reference term of the *SC* (*T* in the present case).

SCs are then filtered to remove those of their terms that are not strongly linked to their reference term. As the meaning of a term is characterized by the context of its occurrences, the evaluation of the strength of the link between a term and a term of its *SC* is based on the number of times that the first term collocates with the second one. More precisely, we use the inclusion coefficient (Michelet, 1988):

$$I(Ti \rightarrow Tj) = \frac{coocc(Ti, Tj)}{occ(Ti)}$$

where $I(Ti \rightarrow Tj)$ is the inclusion coefficient of the term *Ti* in the term *Tj*, *i.e.* the proportion of the occurrences of *Ti* that collocate with an occurrence of *Tj*.

Finally, a term T_{SC} of the SC of the term T_{ref} is selected only if it fulfills the two following conditions: $I(T_{SC} \rightarrow T_{ref}) > S1$ and $I(T_{ref} \rightarrow T_{SC}) > S2$, where *S1* and *S2* are thresholds that were set experimentally. These two conditions discard terms that are too general, *i.e.* spread over a large number of *SCs*, or too specific. In this way, they ensure that the link between a reference term and a term of its *SC* is really significant from the viewpoint of their meaning.

5. Organization of terms into a hierarchy

5.1. Overall algorithm

The building of a hierarchy of terms[3] is achieved by an iterative process that starts from the most general terms: at each iteration, a set of root terms is first defined from the terms that are not linked to the hierarchy yet (unclassified terms). A parent term in the hierarchy is then selected for each of these root terms and a hierarchical semantic relation is set between them. The root terms are terms for which a parent term cannot be found in the set of unclassified terms. The starting root terms, *i.e.* the roots of the hierarchy of terms, are given *a priori* (directly by an user or by applying another criterion) or automatically determined by the algorithm that finds the root terms at each iteration. The global process goes on until the set of unclassified terms is empty. This set shrinks after each iteration when the terms that have been newly linked to the hierarchy are removed from it.

Each iteration of the algorithm we have presented is divided into four stages:

1. selection among unclassified terms of root terms that will be linked to the hierarchy under construction at the end of the iteration;
2. search for the possible parents of each root term, *i.e.* the terms of the hierarchy to which this root term can be linked with a hierarchical semantic relation;
3. selection of the parent of each root term in the hierarchy (see Section 5.2);
4. integration of root terms into the hierarchy under construction (see Section 5.3).

The second stage aims at improving the efficiency of the algorithm: the cost of the criteria used for selecting the parent of a root term at stage 3 is too high from an algorithmic viewpoint for applying them to each term of the hierarchy under construction. The selection of a restricted set of possible parents is a way to quickly dismiss terms of the hierarchy that have clearly no relation

[3] In this article, we use the expression "hierarchy of terms" for referring to the result of our algorithm. More formally, the structure that it builds is a forest of trees as it can have multiple roots.

with a root term. More precisely, a term of the hierarchy is selected as a possible parent for a root term if it fulfills the two following conditions:

– the *SC* of the term from the hierarchy must have a larger size than the *SC* of the root term;

– the term from the hierarchy must belong to the *SC* of the root term.

The selection of a set of possible parents for an unclassified term is used not only as a preliminary filter for determining the parent of a root term but also for the selection of root terms at stage 1 since in practice, a root term is defined as a term that has none possible parent among unclassified terms.

5.2. Selection of the parent of a root term

As stated by the principles presented in Section 2, the search for the term *Tp* as a possible parent in the hierarchy under construction of a root term *Tr* is based on criteria related to the intersection of their *SCs*: the *SC* of *Tp* must cover a large part of the one of *Tr* but the intersection between the two *SCs* must not to be too small in relation to the *SC* of *Tp*. When these two criteria are not fulfilled, we can suppose that the difference of generality between *Tp* and *Tr* is too important and that they cannot be linked directly.

More formally, these criteria rely on a coefficient that evaluates the covering of a *SC* by another. The coefficient of covering of a term *Ti* by a term *Tj* is given the coefficient of covering of their *SCs*:

$$C(Ti \rightarrow Tj) = \frac{card(SC(Ti) \cap SC(Tj))}{card(SC(Ti))}$$

The first criterion for finding the parent *Tp* of a root term *Tr* finds expression in the selection of terms such as $C(Tr \rightarrow Tp) >$ S3. The second criterion selects terms such as $C(Tp \rightarrow Tr) >$ S4. S3 and S4 are parameters whose value was set experimentally as for previous parameters.

If the set of the possible parents for *Tr* contains more than one term after this filtering, the parent of *Tr* is the term *Tp* whose the coefficient of mutual covering with *Tr* has the highest value. For two terms *Ti* and *Tj*, this coefficient is given by the following product:

$$C(Ti, Tj) = C(Ti \rightarrow Tj) \times C(Tj \rightarrow Ti)$$

At the conclusion of this third stage, a parent term in the hierarchy under construction is assigned to each root term of the set of unclassified terms.

5.3. Integration of terms into a hierarchy

The integration of a term *T* into the hierarchy of terms goes together with an adaptation of its *SC*. This adaptation characterizes the fact that *T* is now in the context of its parent term. It is made necessary by the heterogeneousness of the content of *SCs*, which results from the way they are built. Two factors mainly contribute to this heterogeneousness:

– the collocation of two terms in a semantic unit such as a paragraph indicates only partially the type of the relation between them: this relation may be a semantic one, as the relations we try to find, but it also may be a syntactic one or results from a contingency, even if the selection of significant terms and the filtering of their semantic contexts tend to reduce this last case;

– the polysemy of terms[4]. This phenomenon exists in general language as it can be seen in dictionaries but it is more acute when the meaning of a term is defined in relation to a corpus, as it is done in our case. In such a situation, terms generally have more senses as a new sense tends to be defined for each specific context in which this term is used. But the collocations collected for building the semantic contexts of terms are collocations between terms and not between senses of terms. Even if a term has one main sense in a corpus, this sense is rarely unique. As a consequence, in the collocations that contains this term, this one refers most of the time to this main sense but it sometimes refers to minority senses as well, which can be a significant source of noise. The problem is of course more acute when the different senses of a term in a corpus are more balanced. It will be tackled more specifically in Section 6.

The organization of terms into a hierarchy contributes to reduce the heterogeneousness of *SCs*. When a term is integrated into the hierarchy under construction, its *SC* is updated in relation to the *SC* of its parent term. The resulting *SC* is called *Restricted Semantic Context* (*RSC*) of the term.

More precisely, the *RSC* of a term is the result of the intersection of its initial *SC* and the *RSC* of its parent term. This operation counters to some extent the two sources of heterogeneousness for *SCs* we have mentioned before:

– by definition, the words that are part of the initial *SC* of a term *T* only by chance are not likely to be part of the *RSC* of its parent term. Hence, they are discarded from the *SC* of *T* by taking the intersection of the two semantic contexts;

– the words that are part of the initial *SC* of a term *T* because they come from collocations that implicate minority senses of *T* in the corpus also have few chances to be in the *RSC* of the parent term of *T* if these senses are different from the one corresponding to the parent term of *T*.

After a term was integrated into the hierarchy under construction, the semantic context taken into account for this term by our algorithm is not its initial *SC* any more but its *RSC*. As a consequence, *RSCs* are more and more precise as the level in the hierarchy of terms increases. At the top of the hierarchy, the *RSCs* of the root terms are by definition identical to their *SCs*.

6. Polysemy of terms

Even in a homogeneous corpus, a term may have different meanings. The differentiation of these meanings is necessary for reducing the noise in the semantic contexts of terms (see Section 5.3) but it is also interesting for characterizing in a more accurate way the concepts that underlie the corpus. The detection and the processing of this phenomenon are taken into account by a little extension of the algorithm we have presented for building

[4] For mono-terms, the problem is even larger as two terms may be homonyms.

a hierarchy of terms. After a term *T* was linked to its parent term, a test is performed to determine if the remaining part of its initial *SC* after the difference between the initial *SC* of *T* and its *RSC* can be linked to another term of the hierarchy under construction. If such a parent term is found, a second test is performed to determine if the *RSC* of this parent term is similar to the part of the initial *SC* of *T* that is already covered by the parent terms found for the previous senses of *T*. A too high similarity would mean that the new sense is very close to the senses that were already distinguished and therefore, that such a differentiation is not justified. In practice, this condition is implemented by comparing to a threshold, S5, the coefficient of mutual covering between the *RSC* of the parent term found for the new sense of *T* and the part of its *SC* already covered (see Section 5.2). If this coefficient is smaller than S5, a new sense is created whose *RSC* corresponds to the intersection between the part of the *SC* of *T* already covered and the *RSC* of the parent term found for this new sense. The creation of new senses for *T* goes on until the conditions for linking the remaining part of its initial *SC* to a term of the hierarchy under construction are not fulfilled any more.

The threshold S5 directly controls the degree of polysemy that is accepted for terms and as a consequence, the granularity of the senses that are distinguished in the hierarchy. If this granularity is high, the resulting hierarchy is a hierarchy of terms while if it is low, the resulting hierarchy is rather a hierarchy of senses.

7. Results and discussion

7.1. Results

The algorithm we have described was implemented by Jean-Luc Simoni during his thesis. The resulting system was tested on the SPIRALE corpus, one of the two corpora built by the Strategic Research Action (ARC) A3 (El Hadi et al. 2001) of the research network Francil (Francophone Network on Language Engineering) for evaluating term and semantic relation extractors in French. The SPIRALE corpus is made up of 19 issues of the SPIRALE journal in the field of education and pedagogy. It gathers around 400 texts in French and has a total size of 16 Mbytes.

Figures 1 and 2 give two extracts of the hierarchy of terms built from this corpus. The values of the parameters of our algorithm that were used for getting these results are the followings:

Filtering of *SCs*: S1 = 0.01 and S2 = 0.01 (see Section 4)

Selection of a parent term: S3 = 0.01 and S4 = 0.01 (see Section 5.2)

Polysemy: S5 = 0.05 (see Section 6)

The hierarchies built by the algorithm we have presented must not be considered as general ontologies such as CYC (Lenat et al., 1990) but rather as the representation of the content of a particular corpus. In the hierarchy of Figure 1 for instance, the fact that the term "arithmétique" (arithmetic) has the term "entier" (integer) as an indirect parent term is specific to the SPIRALE corpus that was the starting point of this hierarchy. The relation could be reversed in another corpus, which would be closer to a general ontology. Some relations are even more specific, as the relation "élève" (student) → "texte" (text) for instance. This characteristic is a major drawback for building general ontologies but it is interesting for building domain specific resources provided that the corpora used for such a task are representative of the considered domains.

élève (student)
  texte (text)
    mot (word)
      lettre (letter)
      passage (passage)
      phrase (sentence)
      mémoire (memory)
      retour (return)
      son (sound)
    langue (language)
      unité (unit)
      grammaire (grammar)
    expression (expression)
    récit (story)
      personnage (character)
      roman (novel)
      suite (continuation)
    proposition (clause)
      liste (list)
    document (document)

Figure 1. Extract from the hierarchy of terms built from the top root term "élève" (student)[5]

élève → problème → mathématique
(student → problem → mathematics → statement)
→ énoncé

calcul (calculation)
  entier (integer)
    fraction (fraction)
      arithmétique (arithmetic)
      nombre rationnel
      (rational number)
    exercice application
    (application exercise)
      activité préparatoire
      (preliminary activity)
        fonction didactique
        (didactic function)
    nombre décimal
    (decimal number)

Figure 2. Extract of a deep sub-hierarchy[6]

[5] We give terms of the hierarchy in their lemmatized French form. Their translation appears in brackets.

Figures 1 and 2 also show that the relations between the terms of a hierarchy resulting from our algorithm are not only hierarchical relations comparable to hyperonymy. Relations such as "texte" (text) → "mot" (word) or "récit" (story) → "personnage" (character) are metonymical ones and more generally, specific relations such as "élève" (student) → "texte" (text) can be considered from a topical point of view: "élève" defines the context in which "texte" is used in this corpus.

7.2. Discussion

7.2.1. Evaluation

As stated by (El Hadi et al., 2001), the evaluation of semantic relation extractors is very difficult both because of the lack of maturity of the field and the difficulty to build gold standards. Moreover, the fact that the hierarchies of terms built by our algorithm are closely tied to the corpora they come from make their comparison with thesaurus that were built manually very difficult.

We have chosen for the moment to study the effect of the parameters of our algorithm on the hierarchies it builds and focus more particularly on metrics that enable us to compare classifications, which is an important point in relation to the evaluation of the extractors of hierarchical semantic relations. More precisely, we work on similarity measures between trees and statistical tests that are used for comparing distance matrix, such as the Mantel test or a tuned version of the Kappa test (Ferret et al., 2001).

We also plan to evaluate the interest of the hierarchies of terms built by our method in an indirect way by using them in the information retrieval field for achieving automatic query expansion (see Section 7.2.2).

7.2.2. Applications

Several kinds of applications are concerned by domain specific hierarchies of terms.

- **Navigation into document bases**

For searching information into a base of documents, a hierarchy of terms that is representative of the content of this base can be used as a conceptual map for navigating in it and quickly determining the topics of its documents.

- **Query expansion**

Two terms that are directly linked in a hierarchy of terms are semantically close. Hence, such a hierarchy can be used as a source of knowledge for query expansion, *i.e.* for adding to queries terms that are linked to their initial terms in order to improve their recall. This can be done in an automatic or a manual way. (Voorhees, 1998) showed that automatic query expansion with a general resource such as WordNet does not get good results if a semantic disambiguation of the query's terms is not done, which is still a difficult task. These results would be certainly better with a hierarchy of terms built from the documents that are queried. This is a point we want to test (see Section 7.2.1). Such an evaluation could start by expanding queries with terms that are direct child terms in the hierarchy of the queries' terms. A manual expansion could also be tested for exploiting more distant relations between terms.

- **Thesaurus building**

The hierarchies of terms resulting from our work are structurally close to thesaurus. They contain of course too much noise, *i.e.* relations that are too specific, for being directly used as thesaurus but they can be a starting point for the representation of a new domain. This requires having a base of documents that is representative of the considered domain.

- **Competitive intelligence**

As a hierarchy of terms gives a global view of the content of a set of documents, it can also show weak signals, *i.e.* information that is not prominent yet but that will be perhaps important in the near future.

[6] The path between the head term of this extract ("calcul") and the top root term of the hierarchy ("élève") is given at the beginning of the extract.

8. Conclusion

In this article, we have presented a method for organizing a set of terms extracted from a corpus into a hierarchy based on a semantic relation between terms comparable to hyperonymy. This method, which is a statistical one, relies on a distributionalist hypothesis: the meaning of a term in a corpus can be characterized by the set of the contexts associated to its occurrences in this corpus. Moreover, it takes into account the polysemy of terms while the hierarchy is built. Hence, the resulting hierarchy is a hierarchy of senses and not only a hierarchy of terms.

This method was implemented and applied on several corpora (scientific journal, patents and administrative documents). The resulting system is now redesigned and reimplemented both to test more easily a large set of hypotheses and to use it for the applications we have presented in Section 7.2.2.

9. References

Assadi, H., 1998. *Construction d'ontologies à partir de textes techniques. Application aux systèmes documentaires*. Thèse de l'Université Paris 6, Paris.

Barbiéri, B., 1992. *Vers une construction automatique de concepts*. Thèse de l'Ecole Centrale de Paris, Paris.

Bourigault, D. and C. Jacquemin, 2000. Construction de ressources terminologiques. In J-M. Pierrel (eds.), *Ingénierie des langues*, Paris: Hermès.

Choi, F., 2000. Advances in domain independent linear text segmentation. *NAACL'00*, Seattle, Washington, USA.

Debili, F., 1982. *Analyse syntaxico-sémantique fondée sur une acquisition automatique de relations lexicales-sémantiques*. Thèse de doctorat d'état de l'Université Paris XI, Orsay.

El Hadi, W.M., I. Timimi, A. Beguin and M. de Brito, 2001. The ARC A3 Project: Terminology Acquisition Tools: Evaluation Method and Task. *ACL-2001 Workshop on Evaluation Methodologies for Language and Dialog Systems*, 42-51, Toulouse, France.

Ferret, O., B. Grau and M. Jardino, 2001. A cross-comparison of two clustering methods. *ACL-2001 Workshop on Evaluation for Language and Dialogue Systems*, Toulouse, France.

Fluhr, C., 1994. SPIRIT : un système d'exploration de données textuelles. *Le Traitement Informatique des Corpus Textuels*, INALF.

Grefenstette, G., 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publisher, Boston, MA.

Habert, B., E.Naulleau and A.Nazarenko, 1996. Symbolic word clustering for medium-size corpora. *16th International Conference on Computational Linguistics (COLING'96)*, Copenhague, Danemark.

Hamon, T., A. Nazarenko and C. Gros, 1998. A step towards the detection of semantic variants of terms in technical documents. *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, 498-504, Montréal, Canada.

Hearst, M., 1997. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1): 33-64.

Hearst, M., 1992. Automatic acquisition of hyponyms from large text corpora. *14th International Conference on Computational Linguistics (COLING'92)*, Nantes, France.

Jouis, C., 1993. *Contribution à la conceptualisation et à la modélisation des connaissances à partir d'une analyse de textes. Réalisation d'un prototype : le système SEEK*. Thèse en informatique de l'Ecole des Hautes Etudes en Sciences Sociales, Paris.

Lenat, D., R. Guha, K. Pittman, D. Pratt and M. Sheperd, 1990. Cyc: towards programs with common sense. *Communications of the ACM*, 33(8): 30-49.

Michelet, B., 1988. *L'analyse des associations*. Thèse de doctorat de l'Université Paris VII, Paris.

Miller, A.G., C. Fellbaum and D. Gross, 1989. WordNet: a Lexical Database Organised on Psycholinguistic Principles. *IJCAI First International Lexical Acquisition Workshop*, Detroit, USA.

Morin, E., 1999. Using Lexico-Syntactic Patterns to Extract Semantic Relations between terms from Technical. *TKE'99*, 268-278, Innsbruck, Austria.

Séguéla, P. and N. Aussenac, 1999. Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine. In R. Teulier (eds.), *IC'99*, 79-88, Palaiseau, France.

Simoni, J-L, 2000. *Accès à l'information à l'aide d'un graphe de termes construit automatiquement*. Thèse en Information Scientifique et Technique de l'Université Paris VII, Paris.

Voorhees, E., 1998. Using WordNet for Text Retrieval. In C. Fellbaum (eds.), *WordNet: An Electronic Lexical Database*, 285-303, Cambridge: MIT Press, MA, USA.

Vossen, P., 1998. Introduction to EuroWordNet. *Computers and the Humanities*, 32(2-3): 73-89.