Structuring Terminology using Analogy-Based Machine learning

Vincent CLAVEAU & Marie-Claude L'HOMME
OLST, University of Montreal
C.P. 6128 succ Centre-Ville
Montréal, QC, H3C 3J7, Canada
{vincent.claveau,mc.lhomme}@umontreal.ca

I – Introduction

In the field of computational terminology, in addition to work on term extraction, more and more research highlights the importance of structuring terminology, that is, finding and labeling the links between terminological units. Retrieving such relations between terms is usually undertaken using either "external" or "internal" methods (see *Daille et al.* (2004) for an overview). External methods rely on the (automatic) analysis of corpora to see what kind of words can be associated with a term in context (*e.g.* Claveau & L'Homme, 2004). Internal methods rely only on the form of the terms to make such associations. Some of this research relies heavily on the use of external knowledge resources (Namer & Zweigenbaum, 2004; Daille, 2003), which implies a lot of human intervention if the technique is defined for another domain or language. Others add little information and make the most of existing data, such as thesauri (Zweigenbaum & Grabar, 2000) or corpora (Zweigenbaum & Grabar, 2003) but aim to identify morphological families without distinguishing the semantic roles of the individual members.

This paper explores the way a simple machine learning technique together with a terminological extraction system can be used to find whether a term is related to another. Our work bears a number of similarities with that developed by Zweigenbaum & Grabar (2003), but it also aims at precisely predicting the semantic link between the two terms. This work relies on two main hypotheses:

1. specialized corpora contain regular morphological relationships coupled with a regular semantic relation;
2. such morphological links may be "exclusive" to the studied domain.

The machine learning technique based on analogy we propose allows us to take into account the particularities of this classification task and does not need any external morphology knowledge in order to comply with our second

hypothesis. The whole technique is evaluated in the domain of computer science and applied on a French corpus.

We first present the framework of this research, and in particular the way we describe and encode the semantic links between morphologically related terms. Then we present the supervised, analogy-based machine-learning technique developed for this task, as well as the terminological extraction system it relies on. Last, we describe the methodology used for the evaluation of our technique and the results obtained.

II – Framework

The work is undertaken in order to assist terminologists in the enrichment of a French specialized dictionary of computing. The dictionary is compiled using a lexico-semantic approach to the analysis of terminology (L'Homme, 2004) and relies heavily on lexical functions, hereafter *LFs* (Mel'čuk et al., 1984-1999) to represent semantic relations between terms. (Entries can be accessed at: http://olst.ling.umontreal.ca/dicoinfo.)

Lexical functions are viewed as a means to capture terminological relationships by an increasing number of researchers. Their completeness and systematicity make them relevant and suitable for several terminological tasks: encoding relationships in dictionaries (Jousse & Bouveret, 2003), structuring terms (Daille, 2003), and classifying specialized collocations (Wanner, 2005, forthcoming).

Various semantic links are encoded in our dictionary of computing. First, users will find syntagmatic links, *i.e.* those expressed by collocates; *e.g. enregistrer* (Eng. *to save*), *défragmenter* (Eng. *to defragment*) and *externe* (Eng. *external*) for *disque dur* (Eng. *hard disk*). Secondly, entries also cover paradigmatic relations, such as hyperonymy, synonymy, antonymy, and actantial relationships. LFs are used to explain in a uniform and systematic manner the meanings of collocates or the relationships between a given key term and another semantically related term.

The work reported in this article is concerned with a subset of semantic relationships. They can be syntagmatic or paradigmatic but they all involve pairs of terms that are morphologically related. Examples of such links are listed below with their corresponding LFs.

S_0(*formater*) = *formatage* (Eng. *to format – formatting*); noun which has the same sense as key word

Sagent(*programme*) = *programmeur* (Eng. *program – programmer*); typical agent of the key word

Sinstr(*éditer*) = *éditeur* (Eng. *to edit – editor*); typical instrument of the key word

Sres(*programmer*) = *programme* (Eng. *to program – program*); typical result of the key word

Anti(*installer*) = *désinstaller* (Eng. *to install – to uninstall*); antonymy

Able$_1$(*interagir*) = *interactif* (Eng. *to interact – interactive*); the agent can + sense of the key word

Able$_2$(*programmer*) = *programmable* (Eng. *program – programmable*); the key word can be verb-ed

A$_1$(*résider*) = *résident* (Eng. *to reside – resident*); the agent has or is + the sense of the key word

A$_2$(*infecter*) = *infecté* (Eng. *to infect – infected*); the patient is + the sense of the key word

De_nouveau(*compiler*) = *recompiler* (Eng. *to compile - to recompile*); once again

Fact$_1$(*pirate*) = *pirater* (Eng. *hacker – to hack*); the key word performs an action on the patient

Labreal$_{12}$(*navigateur*) = *naviguer* (Eng. *browser – to browse*); the agent uses the key word to act on the patient

Caus$_1$Func$_0$(*imprimé*) = *imprimer* (Eng. *printout - to print*); the agent creates the key word

Caus$_1$Oper$_2$(*partition*) = *partitionner* (Eng. *partition – to partition*); the agent causes that the patient has a key word

CausPred(*valide*) = *valider* (Eng. *valid – to validate*); something or somebody renders + key word

It is important to point out that LFs are designed to represent semantic relationships regardless of formal similarity (morphological resemblance is considered as accidental in this framework). However, in this work, according to our first hypothesis, it is assumed that formal resemblance is likely to be indicative of a strong semantic link.

Other work has shown that morphological proximity – even if it does not reveal the entire terminological structure of a domain – can shed light on important terminological relations in many domains:

- Medicine (Zweigenbaum & Grabar, 2000) : *acide, acido, acidité, acidurie, acidémie, acidophile, acidocitose*;
- Agri-food industry (Daille, 2003): *solubilisation micellaire => insolubilisation micellaire, plume de canard => plumard de canard, filetage de saumon => filet de saumon*;
- Business (Binon *et al.*, 2000): *promotion, promo, promoteur, promotrice, promouvoir, promotionner*.

III – Machine Learning Technique

1 – Learning by Analogy

The learning method underlying our approach is based on analogy. Analogy can be formally represented as A : B :: C : D which means *"A is to B what C is to D"* (Lepage, 2003). Learning by analogy has already been used in some NLP applications (Lepage, 2004).

It is particularly suited for our task, in which such analogies can be drawn from our morphologically related pairs. For example we have analogies like the following one: *connecteur* : *connecter* :: *éditeur* : *éditer* (Eng. *connector* : *to connect* :: *editor* : *to edit*); knowing that Sinstr(*connecter*) = *connecteur*, we can guess that the same link (*i.e.* the same LF) is valid for describing *éditeur* and *éditer*, that is Sinstr(*éditer*) = *éditeur*.

From a machine learning point of view, this approach using learning by analogy has several interesting particularities. First, it is "inherently" a supervised method, being a special case of case-based learning (Kolodner, 1993) in which an instance is a pair of word; thus, we do need examples of related pairs along with their LF. Secondly, the number of classes considered, that is the different LFs describing our derivational links, is quite large and dependent on the set of examples. Last, a given pair of morphologically related words can be (correctly) tagged by several LFs. These properties make it impossible to use many other existing machine learning techniques in which multiple classes cannot be assigned to a given instance.

2 – Preparing the Training Data

In order to identify morphological analogies, we need examples of morphologically related terms along with their LF. To gather them, we use the existing entries in the dictionary we are planning to enrich. They are automatically extracted from it by searching, within all the encoded links between terms, for the ones such that the two linked terms are "close" in terms of edit distance or longest common substring.

Thus, even if our learning method is supervised, our technique finally does not require any human intervention; the whole process is actually semi-supervised. In the experiments reported below, about 900 examples are gathered this way and then used to draw the analogies with the test set pairs.

3 – Analogy between Morphologically Related Pairs

The most important feature in learning by analogy is of course the notion of similarity which is used to determine that two pairs of propositions – in our case, two pairs of lemmas – are analogous. The similarity notion we use, hereafter *Sim*, is quite simple but well adapted to French (as well as many other languages), in which derivation in mainly obtained by prefixation and suffixation.

Let us note LCSS(X,Y) the longest common substring shared by two strings X and Y, $X +_{suf} Y$ being the concatenation of the suffix Y to X, $X -_{suf} Y$ being the subtraction of the suffix Y of X, $X +_{pre} Y$ being the concatenation of the prefix Y to X, and $X -_{pre} Y$ being the subtraction of the prefix Y of X. The similarity notion Sim works as follows (an example is given below): if we have two pairs of words W_1-W_2, W_3-W_4,

$$\mathrm{Sim}(W_1\text{-}W_2, W_3\text{-}W_4) = 1 \text{ if } \begin{cases} W_1 = \mathrm{LCSS}(W_1, W_2) +_{pre} Pre_1 +_{suf} Suf_1, \text{ and} \\ W_2 = \mathrm{LCSS}(W_1, W_2) +_{pre} Pre_2 +_{suf} Suf_2, \text{ and} \\ W_3 = \mathrm{LCSS}(W_3, W_4) +_{pre} Pre_1 +_{suf} Suf_1, \text{ and} \\ W_4 = \mathrm{LCSS}(W_3, W_4) +_{pre} Pre_2 +_{suf} Suf_2 \end{cases}$$

otherwise

$$\mathrm{Sim}(W_1\text{-}W_2, W_3\text{-}W_4) = 0.$$

Pre_i and Suf_i are any character strings. Intuitively, Sim checks that the same "path" of deprefixation, prefixation, desuffixation and suffixation is needed to go from W_1 to W_2 as to go from W_3 to W_4. If $\mathrm{Sim}(W_1\text{-}W_2, W_3\text{-}W_4) = 1$, the analogy $W_1 : W_2 :: W_3 : W_4$ stands and, if the LF between W_1 and W_2 is known, the same one certainly holds between W_3 and W_4.

Our morphological tagging process involves checking if an unknown pair is in analogy with one or several of our examples. If so, the unknown pair is tagged with the same LF (or possibly several LFs) as the examples. Practically, we learn from our examples the way Sim is computed, that is, the path of operations needed to go from a word to another in terms of Pre_i and Suf_i, and assigns the LF to this path. For instance, if V_0(*programmation*) = *programmer* (Eng. *programming, to program*) is an example, the following path is learned:

$$V_0(W_1) = W_2 \quad \text{if} \quad W_1 -_{suf} \text{“ation”} +_{suf} \text{“er”} = W_2$$

Any new pair following this path will be annotated with the V_0 LF. Conversely, since we also know that S_0(*programmer*) = *programmation*, we also have a rule:

$$S_0(W_1) = W_2 \quad \text{if} \quad W_1 -_{suf} \text{“er”} +_{suf} \text{“ation”} = W_2$$

Similarly, from the example $\mathrm{Able}_2\mathrm{Anti}$(*activer*) = *désactivable* (Eng. *activate – deactivatable*), the following rule is built:

$$\mathrm{AntiAble}_2(W_1) = W_2 \quad \text{if} \quad W_1 -_{suf} \text{“er”} +_{suf} \text{“able”} +_{pre} \text{“dés”} = W_2$$

In all, 402 morphological rules are obtained from our examples, allowing us to identify 67 different LFs. Any pair of words that complies with one of these rules is therefore in analogy with one of our 900 example pairs and can be annotated by the same LF as in this example.

4 – Use of the Term Extraction System TermoStat

In addition to the learning process described above, we use a corpus-based term-extraction system called TermoStat (Drouin, 2003). This system, contrary to many other term-extraction techniques, is able to retrieve single-word terms. To perform this extraction, TermoStat computes the "specificities" of words occurring in a specialized corpus by comparing their frequency in the corpus and in a general-language corpus. Basically, the higher the specificity of a word, the more likely it is to be a term of the domain. Conversely, a word with a negative specificity coefficient certainly belongs to the general language.

The French domain-specific corpus used in our experiments is composed of several articles from books or web sites specialized in computer science; all of them were published between 1996 and 2004. It covers different computer science sub-domains (networking, managing Unix computers, webcams...) and comprises about 1,000,000 words. This corpus is thus compared to the French general corpus *Le Monde*, composed of newspaper articles (Lemay *et al.*, 2005).

In our experiments, TermoStat, by providing us with words likely to be domain-specific terms, is used to filter out non-related pairs within the domain framework. Indeed, we can avoid wrong associations like *architecture-architectural* (Eng. *architecture (of a system or network)-architectural*) (in which *architectural* is morphologically related to *architecture* from a diachronic point of view, but not semantically related in the computer science domain), since *architectural* does not have a high specificity coefficient. Thus, to retrieve domain-relevant morphologically related terms and annotate them with their LFs, the 402 learned rules are applied to each possible pair of words having a specificity coefficient higher than a certain threshold.

It is interesting to note that the use of TermoStat allows us not only to focus on terms of the domain, but also to reduce the time complexity of our algorithm. Indeed, detecting analogies implies to test every possible pair of lemmas supplied by the corpus with our rules; thus, the complexity of our approach is $O(n^2)$ with n the number of lemmas in the corpus. Focusing on lemmas with specificities higher than a certain threshold keeps n at a lower level and reduces the computing cost of the analogy search.

IV - Evaluation

This section is devoted to describing the evaluation of the technique presented above. We first present the test set used, and then we describe the measures chosen to precisely evaluate our system and the results obtained.

1 – Building the Test Set

In order to evaluate the completeness and the precision of the results obtained by our technique, we built a test set containing morphologically related terms along with their LFs. The first step of this process involves randomly selecting more than 220 words from the lemma list of the computer science corpus. Then, for each of these 220 test words, we constitute pairs by manually retrieving in the corpus all the morphologically related lemmas, but only if the two words composing the pair are terms sharing an actual semantic link in the computer science domain. This means that pairs like *découvrir – découverte* (Eng. *to discover-discovery*) are not considered as relevant since neither of the words are terms and that the pair *référentiel – référencer* (Eng. *referential – to reference*) is not considered as relevant since there is no semantic link in the computer-science domain. Finally, each pair of related words is given all its possible LFs. In the case of polysemous words, a pair can receive several LFs describing all the relations between the two terms); conversely, some of the words do not have any morphologically related word in the corpus.

Table 1 gives some statistics on this test set. Note that to prevent any bias in the results, none of these terms were used as examples during the learning step; they were removed from the example set.

Total number of different test words	222
Total number of pairs	469
Number of different links (LFs)	50

Table 1 Statistics on the Test Set

2 – Results

In order to evaluate our results, we are interested in two questions: do we find all the existing links between two units? do we find only valid links? To answer these two questions, we use the standard recall/precision approach. The

global quality of the system is measured with the help of a single rate, the f-measure (harmonic mean of R and P), defined as: f = 2PR/(P+R).

The evaluation process is the following: we apply the learned rules to each possible pair of words in the corpus having a specificity coefficient higher than a certain threshold and containing one of the 220 test words. A pair matched by one of the rules is in analogy with one of the example and thus receives the same LF. The list of annotated pairs obtained is compared to the one built manually in order to compute R, P and f. This evaluation process is repeated for different specificity thresholds in order to evaluate the influence of this parameter. Figure 1 presents the variation of R, P and f with respect to the specificity threshold. The threshold value that maximizes the f-measure is 0; with this value, we have: f = 0.6848 with R = 71.77% and P = 65.48%.

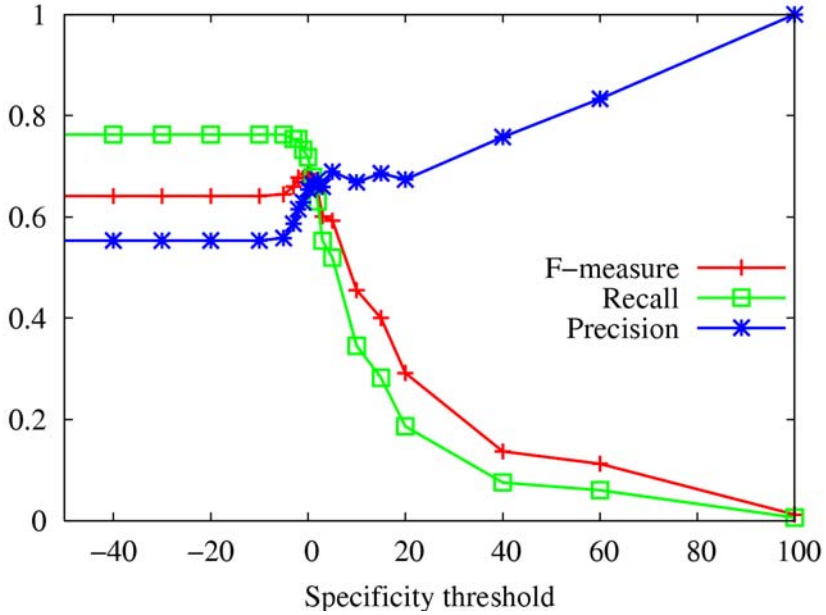

Figure 1 Variation of the Recall and Precision rates and f-measure according to the specificity threshold

Given the simplicity of our approach, these results are surprisingly good in terms of both recall and precision. As expected, focusing on the positive specificities ensures that we obtain more precise results, leading to a better recall/precision compromise than if the method had been applied on the whole list of words in the corpus. Moreover, the optimal threshold is 0, meaning that our results are coherent with the way TermoStat retrieves term candidates; and no good relations are found in words with a specificity coefficient lower than -5.

Basically, errors produced by our method can be classified into two different groups. First, it can detect an erroneous semantic relation in a pair (this type of error is called a false positive). False positives are mainly generated by:

- The detection of pairs in which at least one word is not a term of the domain: *e.g. démasquer-masquer* (Eng. *reveal-mask*).
- The detection of pairs in which terms do not share a relevant relationship in the field: e.g., *table-tablette* (Engl. *table-shelf*).
- Detection of valid pairs but with a wrong LF:
  - Many errors in this category are due to morphemes that convey different meanings. Nouns ending in *-eur* can be instruments, like *éditeur*, or agents, like *programmeur*, of the related verb). We can also mention nominalizations of verbs (nouns ending in *–ation, - age, -ment*, etc.) most of which can convey two different meanings, that of result and that of activity. However, in some cases, the noun only conveys one of those meanings: *e.g.*, in *balayage-balayer* (Engl. *analysis, to analyze*), *balayage* only conveys a meaning of activity.
  - Some morphological configurations are frequently associated with a given relationship but can be confused, in a few rare cases, with an invalid relationship: e.g. *re-* used almost exclusively in terms that mean "once again" as in *configurer-reconfigurer* (Eng. *configure-reconfigure*). Our system wrongly labeled a pair that shares a different relationship: e.g. *chercher-rechercher* (Eng. *search*). Incidently, *rechercher* cannot be decomposed into a base meaning "chercher" and a morpheme meaning "once again" and the terms in this case are synonyms.

The second kind of error is due the failure of our method to detect valid pairs (called false negatives). These are mainly due to the following errors:

- The absence of one of the terms in the list of specificities: *e.g.*, in *aide-aider* (Engl. *help-to help*), the noun was part of the specificities, but not the verb.
- Rare morphological configurations that do not appear in our examples: *e.g.* S_0Inter(*connecter*) = *interconnexion* (Eng. *to connect-interconnection*).
- Morphological configurations for which we do have some examples, but no example with the valid semantic link: *e.g. brancher-branchement* (Eng. *to connect-connection*) was identified but the semantic relationship was wrong.

Finally, results can be presented to the terminographer in the form of graphs such as the one shown in Figure 2. Note that in this case two wrong LFs were detected: Sres between *compiler-compilation* and *recompiler-recompilation*.

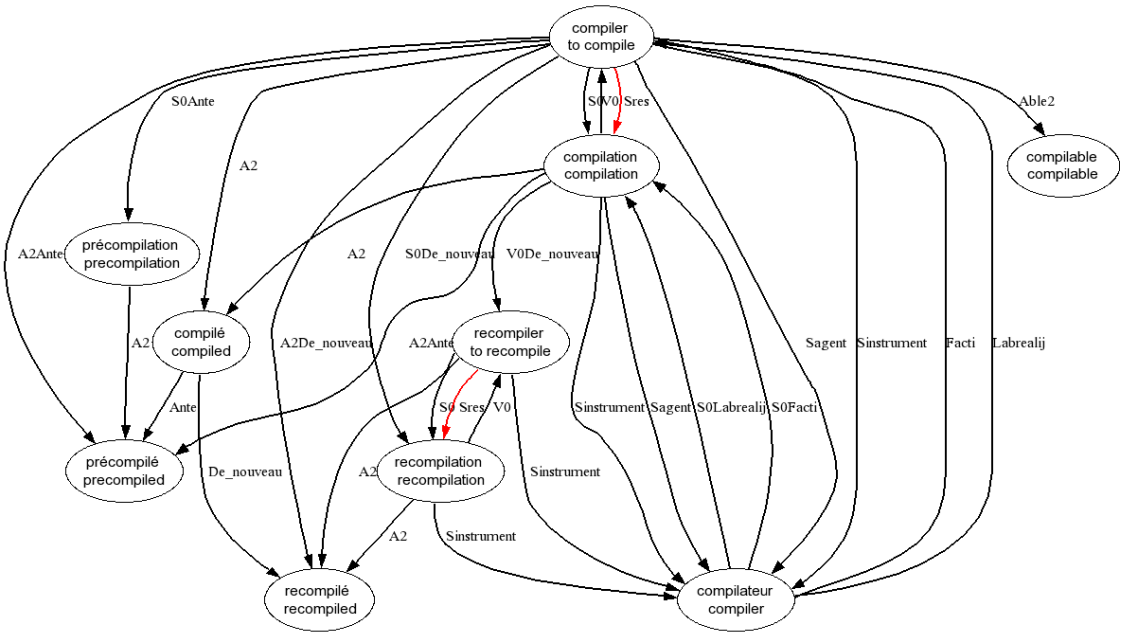

Figure 2 Resulting graph for the "compilation" morphological family

V – Conclusion

This paper presents a simple method for automatically retrieving and identifying a semantic relation, expressed with the help of Lexical Functions, between morphologically related terms of a domain. This technique uses a special kind of machine learning approach based on analogies and the results of a term-extraction system. The relative simplicity of the technique is actually one of its most important advantages. Indeed, it does not rely on predefined classes of relations or LFs, nor on external knowledge or language. Moreover, results obtained, measured through an evaluation in the field of computer science, are very good, both in terms of completeness and precision of the semantic relations found.

With these experiments, we have also confirmed the first hypothesis underlying this work: morphological proximity generally indicates semantic proximity, which can be encoded by LFs. To verify our second hypothesis, that is, that these morphological links have to be learned for each domain, it is necessary to conduct experiments on other domains. However, similar experiments

drawing analogies from general language examples (Claveau & L'Homme, 2005) and close experiments in the biomedical domain (Zweigenbaum & Grabar, 2000) tend to confirm it.

Future work is planned to solve some frequent errors, such as the ones reported in Section IV.2, by using other approaches that incorporate an analysis of syntagmatic relationships (Claveau & L'Homme, 2004). From an application point of view, we are planning to use the same technique on a computer-science corpus in English.

VI – Acknowledgements

The authors would like to thank Léonie Demers-Dion for her help in the test set construction and Elizabeth Marshman for her comments on a previous version of this article.

VII – References

Binon J., S. Verlinde, J. Van Dyck & A. Bertels, *Dictionnaire d'apprentissage du français des affaires*. Paris: Didier, 2000.

Claveau V. & M.-C. L'Homme; Discovering Specific Semantic Relationships between Nouns and Verb in a Specialized French Corpus, *Proceedings of the 3rd International Workshop on Computational Terminology (CompuTerm'04)*, Geneva, Switzerland, 2004.

Claveau V. & M.-C. L'Homme; Apprentissage par analogie pour la structuration de terminologie - Utilisation comparée de ressources endogènes et exogènes, *Actes de la conférence Terminologie et Intelligence Artificielle (TIA'05)*, Rouen, France, 2005.

Daille B.; Conceptual Structuring through Term Variation, *Workshop on Multiword Expressions. Analysis, Acquisition and Treatment. Proceedings of ACL 2003*, Sapporo, Japan, 2003.

Daille B., K. Kageura, H. Nakagawa & L.-F. Chien (eds); *Terminology*. *Special Issue on Recent Trends in Computational Terminology*, 10(1), 2004.

Drouin P.; Term-extraction using Non-technical Corpora as Point of Leverage, *Terminology*, 9(1), 2003.

Kolodner J. (ed.); *Machine Learning, Special Issue on Case-Based Reasoning,* 10(3), 1993.

Jousse, A.-L. & M. Bouveret; Lexical Functions to Represent Derivational Relations in Specialized Dictionaries, *Terminology*, 9(1), p. 71-98, 2003.

Lemay C., M.-C. L'Homme & P. Drouin; Two Methods for Extracting Specific Single-word Terms from Specialized Corpora: Experimentation and Evaluation, *International Journal of Corpus Linguistics,* 10(2), 2005.

Lepage Y.; *De l'analogie; rendant compte de la communication en linguistique*. Grenoble, France, 2003.

Lepage Y.; Lower and Higher Estimates of the Number of "True Analogies" between Sentences Contained in a Large Multilingual Corpus, *Proceedings of the 20th International Conference on Computational Linguistics, COLING'04*. Geneva, Switzerland, 2004.

L'Homme M.-C.; A Lexico-Semantic Approach to the Structuring of Terminology, *Proceedings of the 3rd Workshop on Computational Terminology, CompuTerm'04*. Geneva, Switzerland, 2004.

Mel'čuk I. *et al.*; *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques I-IV*, Montréal: Les Presses de l'Université de Montréal, 1984-1999.

Namer F. & P. Zweigenbaum; Acquiring Meaning for French Medical Terminology: Contribution of Morpho-semantics, *Conference Medinfo 2004*. San-Francisco, USA, 2004.

Wanner, L. *et al.*; The First Step towards the Automatic Compilation of Specialized Dictionaries, *Terminology*, 11(1), forthcoming, 2005.

Zweigenbaum P. & N. Grabar; A Contribution of Medical Terminology to Medical Language Processing Resources: Experiments in Morphological Knowledge Acquisition from Thesauri, *Conference on Natural Language Processing and Medical Concept Representation*, Phoenix, USA, 1999.

Zweigenbaum P. & N. Grabar; Liens morphologiques et structuration de terminologie, *Ingénierie des connaissances, IC 2000,* p. 325-334, 2000.

Zweigenbaum P. & N. Grabar; Learning Medical Words from Medical Corpora, *Conference on Artificial Intelligence in Medecine, AIME'03,* Protaras, Cyprus, 2003.