Structuring terminology using analogy-based machine learning

Vincent CLAVEAU and 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) study 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 one wishes to change domains or languages of study. Others add little information and make the most of existing data (Zweigenbaum & Grabar, 2000; 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 yet original machine learning technique together with a terminological extraction system can be used to find whether a term is related to another and also to precisely predict 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 whole technique is evaluated in the domain of computer science and applied on a French corpus.

We first present the framework of this research. 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.

Various semantic links are encoded in the dictionary. 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
Anti(*installer*) = *désinstaller* (Eng. *to install – to uninstall*); antonymy
Able$_2$(*programme*) = *programmable* (Eng. *program – programmable*); the key word can be verb-ed
De_nouveau(*compiler*) = *recompiler* (Eng. *to compile - to recompile*); once again
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
Caus$_1$Func$_0$(*imprimé*) = *imprimer* (Eng. *printout - to print*); the agent creates the key word

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, where such analogies can be drawn from our morphologically related pairs. For example we have analogies like: *connecteur* : *connecter* :: *éditeur* : *éditer* (Eng. *connector* : *to connect* :: *editor* : *to edit*); knowing that Sinstr(*connecter*) = *connecteur*, we can guess that there is the same link (*i.e.* the same LF) between *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); 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.

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. In the experiments reported below, about 900 examples were gathered this way.

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,

$$\mathrm{Sim}(w_1\text{-}w_2,\, w_3\text{-}w_4) = 1 \text{ if } \begin{cases} w_1 = \textsc{lcss}(w_1, w_2) +_{pre} \mathrm{Pre}_1 +_{suf} \mathrm{Suf}_1, \text{ and} \\ w_2 = \textsc{lcss}(w_1, w_2) +_{pre} \mathrm{Pre}_2 +_{suf} \mathrm{Suf}_2, \text{ and} \\ w_3 = \textsc{lcss}(w_3, w_4) +_{pre} \mathrm{Pre}_1 +_{suf} \mathrm{Suf}_1, \text{ and} \\ w_4 = \textsc{lcss}(w_3, w_4) +_{pre} \mathrm{Pre}_2 +_{suf} \mathrm{Suf}_2 \end{cases}$$

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 V0(*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 Able$_2$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), which 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. 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 comprises about 1,000,000 words dealing with different topics in computing. This corpus is thus compared to the French general corpus *Le Monde*, composed of newspaper articles (Lemay *et al.,* forthcoming).

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 *application-appliquer* (Eng. *software-apply*) (in which *appliquer* is morphologically related to *application* from a diachronic point of view, but not semantically related in the computer science domain), since *appliquer* 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.

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.

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.

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 use the standard recall/precision approach. The global quality of the system is measured with the help of a single rate, the f-measure, 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%.

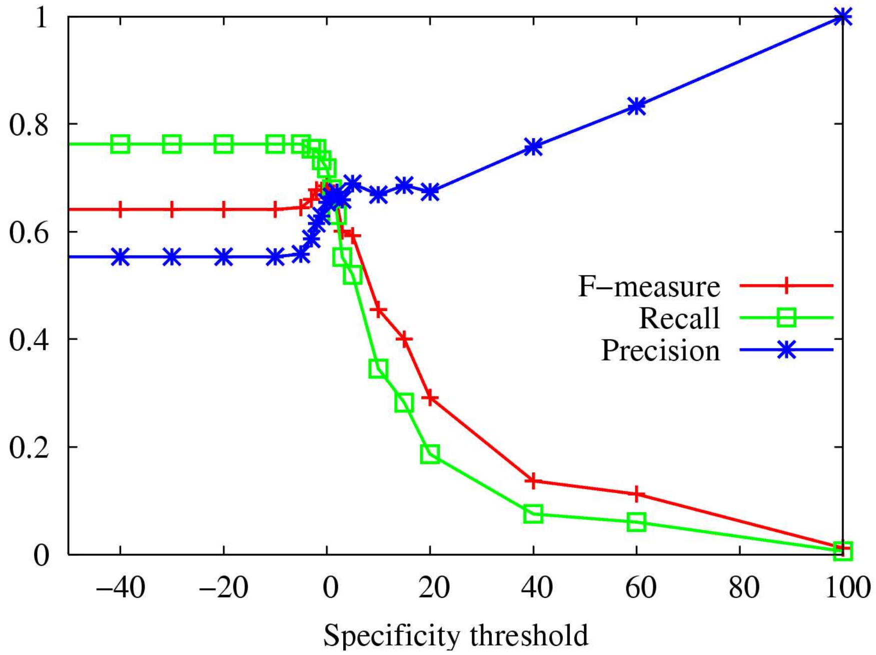

Figure 1: Variation of the Recall and Precision rates and f-measure according to the specificity threshold

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.

There are two kinds of errors that are made by our method. In the first, it wrongly detects a semantic relation in a pair. This is mainly due to the detection of valid pairs but with a wrong LF (indeed, many errors are due to nouns ending in *-eur* that can be instruments, like *éditeur*, or agents, like *programmeur*, of the related verb). The second kind of error is the failure to detect valid pairs. This is mainly due to the absence of one of the terms in the list of specificities or to rare morphological configurations that do not appear in our examples (*e.g.* S_0Inter(*connecter*) = *interconnexion*, (Eng. *to connect-interconnection*)).

Finally, results can be presented to the terminographer as graphs such as Figure 2. Note that there are two wrongly Sres LFs detected between *compiler-compilation* and *recompiler-recompilation* in this figure.

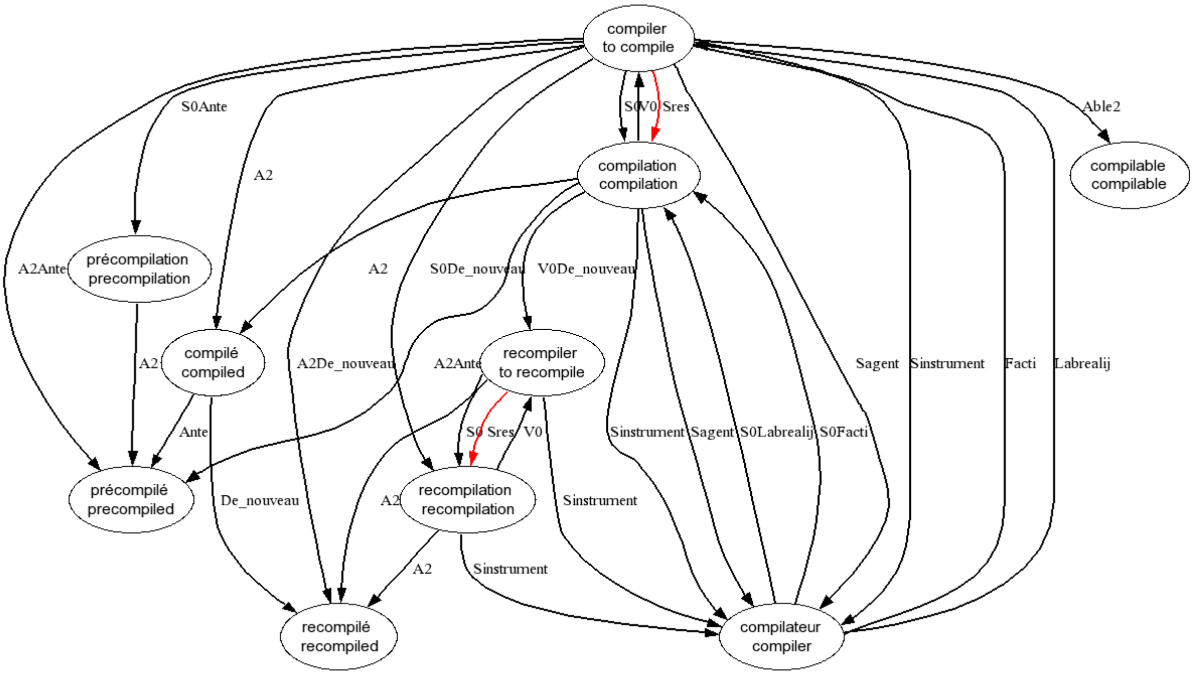

Figure 2: Resulting graph for the "compilation" morphological family

V. Conclusion

This paper presents a simple yet original 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. It does not rely on predefined classes of relations or LFs, nor on external knowledge or language. Moreover, results obtained are very good, both in terms of completeness and precision of the semantic relations found.

Through these experiments, we have also confirmed the first hypothesis underlying this work: morphological proximity generally indicates semantic proximity. 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, 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 (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. References

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

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

Daille B., Kageura K., Nakagawa H. and Chien L.-F. (Eds). 2004. *Terminology. Special issue on Recent Trends in Computational Terminology.* 10(1).

Drouin P. 2003. Term-extraction using non-technical corpora as point of leverage. *Terminology,* 9(1).

Kolodner J. (Ed) 1993. *Machine Learning, special issue on Case-Based Reasoning,* 10(3).

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

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

Lepage Y. 2004. Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus. In *20th International Conference on Computational Linguistics, COLING'04*. Geneva, Switzerland.

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

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

Namer F., Zweigenbaum P. 2004. Acquiring meaning for French medical terminology: contribution of morpho-semantics. In *Conference Medinfo 2004*. San-Francisco, USA.

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

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

Zweigenbaum P. and Grabar N. 2003. Learning medical words from medical corpora. In *Conference on Artificial Intelligence in Medecine, AIME'03,* Protaras, Cyprus.