BDL-CA

Bases de données lexicales : construction et applications

Lundi 23 avril 2007
Département de linguistique et de traduction
Pavillon Lionel-Groulx
Université de Montréal

Colloque organisé par l'Observatoire de linguistique Sens-Texte (OLST)
sous la responsablilité de
Jeesun Nam (DICORA, Hankuk University of Foreign Studies)
et
Alain Polguère (OLST, Université de Montréal)

http://www.olst.umontreal.ca

Remerciements L'organisation du colloque a bénéficié d'un support financier du FQRSC (Québec) ainsi que du Département de linguistique et de traduction de l'Université de Montréal.

Tables des matières

BDL-CA, Montréal, 23 avril 2007

A Local Grammar of French determiners for deep syntactic parsing

Éric Laporte

University of Marne-la-Vallée - IGM
5, Bd Descartes - 77454 Marne-la-Vallée CEDEX 2 - France
eric.laporte@univ-mlv.fr

Abstract

Existing syntactic grammars of natural languages, even with a far from complete coverage, are complex objects. Assessments of the quality of parts of such grammars are useful for the validation of their construction. We extended a grammar of French determiners that takes the form of a recursive transition network. The result of the application of this local grammar gives deeper syntactic information than chunking or information available in treebanks. We evaluated its quality by comparison with a corpus independently annotated with information on determiners. We obtained 86% precision and 92% recall on text not tagged for parts of speech.

Keywords

Determiner, language resource, local grammar, recursive transition network, determinative noun, evaluation.

1 Introduction

The coverage of existing syntactic-semantic grammars of natural languages is far from complete, but even so, such grammars are complex objects and their construction takes many years. Therefore, it is desirable to assess the quality of parts of a grammar and to control their evolution before it is complete.

In this paper, we report the extension and evaluation of a partial syntactic-semantic grammar of French: a grammar of determiners, including complex determiners and combinations of determiners. This grammar neglects dependencies between the determiner (*Det*) and the noun (*N*). It takes the form of a recursive transition network (RTN). As compared to chunking, the syntactic information obtained by the application of the grammar is deeper, since the grammar describes complex determiners which may contain several chunks. The output of the parser was compared to a corpus independently annotated with information on determiners.

This article is organised as follows. The next section surveys related work. In section 3, we describe the grammar of determiners. Section 4 reports how the grammar was evaluated. We present the results in section 5. The article ends with concluding remarks.

2 Related work

In recent campaigns of evaluation of syntactic grammars (Paroubek et al., 2006), each grammar was assessed globally. Evaluation consisted in comparing the output of the parser to a treebank, and no evaluation of separate parts of grammars was organised. However, parts of a manually constructed grammar have not necessarily the same author or the same quality, and are not necessarily built at the same time. Therefore, it is also desirable to assess the quality of parts of a grammar and to control their evolution during their construction.

Partial grammars are mainly grammars for NE recognition, for chunkers, or for both. The grammatical formalisms used for these tasks are usually regular expressions (RE), transducers manually contructed in the form of RE-like formulae, specific formalisms designed for particular linguistic phenomena, or RTNs (Nam & Choi, 1997), (Senellart et al., 2001), (Saetre, 2004). The symbols recognising words in these grammars are either lexical words or variables which are equivalent to feature structures and refer to various features provided by lexical analysis. Evaluation is performed by comparing the output of the parser to a corpus which has been independently annotated for NEs or chunked.

Partial grammars expressed in the form of RTNs are usually called 'local grammars'. The RTN formalism is adapted to NE recognition (Nam & Choi, 1997) and chunking (Poibeau, 2006) but also to deep syntactic parsing (Fairon et al., 2005), (Blanc & Constant, 2005).

In the recent years, several projects have been devoted to the design and construction of local grammars as components of deep syntactic grammars. The delimitation of the scope of such local grammars requires that dependency on other parts can be neglected. (Typical syntactic grammars are very modular, but contain many dependencies: descriptions of sentences invoke descriptions of noun phrases, etc.). Examples of such local grammars deal with:

- determiners in French, including complex determiners and combinations of determiners (Gross, 2001), (Silberztein, 2003),
- sequences of verbs in French (Gross, 1998-1999) and Portuguese (Ranchhod et al., 2004),
- compound conjunctions in Bulgarian (Venkova, 2000),
- coordinated noun phrases in Serbo-Croatian (Nenadic, 2000),
- a general-purpose set of local grammars for constituents such as noun phrases and other clause elements in English, used to recognise syntactic constructions of verbs (Mason, 2004),

- discrimination between expletive and anaphoric occurrences of the French pronoun *il* "it" (Danlos, 2005),

- noun phrases with predicative head in French (Laporte et al., 2006).

Such projects being instances of a bottom-up approach to the construction of deep syntactic grammars, quantitative evaluation of the resulting resources is particularly useful for the validation of this approach. It is also an indication about the usability of these resources in other projects. However, only three of the contributions listed above report corpus-based evaluation. Danlos (2005) claims 97% accuracy on a corpus of about 240,000 words. Silberztein (2003) reports 100% recall on a sample of about 4200 words, but as regards precision, mentions only that it is 'very low'. Gross (1998-1999) claims 99.8% precision, but does not give the size of the evaluation corpus, nor an assessment of recall. Therefore, very few quantitative data about the coverage of local grammars are presently available. We provide such data referring to a grammar of French determiners.

3 The grammar

The grammar is a description of French determiners, including complex determiners and combinations of determiners. We developed it manually from three existing RTNs: two grammars of French determiners (Gross, 2001), (Silberztein, 2003) and a grammar of numerical expressions (Constant, 2000). The grammar is freely available on the GraalWeb page (Constant, this volume; http://igm.univ-mlv.fr/~mconstan/library/index_graalweb.html). In this section, we delimit the scope of the grammar and report how it was constructed.

3.1 Scope

In language engineering and traditional grammar, determiners are usually viewed as a part of speech, i.e. a morpho-syntactic category of words, rather than as a syntactic notion. This view is only a simplification. Determiners behave according to a complex syntax. Some of them are employed with prepositions, e.g. in *beaucoup de facteurs* 'plenty of factors'. Some combine together, as in *les sept pays* 'the seven countries'. In French, the interaction between the frequent preposition *de* 'of' and determiners involves complex rules. Some noun phrases behave as determiners of other nouns, as in *une partie des prêts* 'part of the loans'. Since most of such noun phrases comprise a determiner in turn, sequences that behave as determiners are embedded in others. We do consider such sequences as (generalised) determiners. We refer to nouns such as *partie* 'part' by the term 'determinative nouns' (*Ndet*).

The scope of the grammar is to describe generalised determiners, defined by Silberztein (2003) as follows: if each noun phrase is assigned a head noun on syntactic and semantic grounds, the (generalised) determiner of the noun is the sequence from the beginning of the noun phrase to the head noun, excluding the head noun itself and possible adjectives directly attached to the head noun. Thus, in *restituer une partie des prêts* 'give back part of the loans', selectional restrictions point to *prêts* 'loans', rather than to *partie* 'part', as the object of *restituer* 'give back'; therefore, the

determiner of the noun phrase *une partie des prêts* is the sequence *une partie des* 'part of the'. The scope of our grammar also includes the prepositions *à* and *de* when they introduce the noun group. The sequences described in the grammar are surface forms such as *au*, and not normalized forms such as *à le*. Predeterminers are considered as parts of the corresponding determiners, as *même* 'even' in *même les grandes avenues* 'even the large avenues', except if they are separated from the determiner by a preposition, as in *même dans les grandes avenues* 'even in the large avenues'.

However, the grammar does not specify morpho-syntactic agreement in gender and number, either between the determiner and the noun, or between the determiner and other elements of the sentence (e.g. the subject-verb agreement). This exclusion is motivated by the fact that the parser that we used, the Unitex parser (Paumier, 2006), does not support unification in its present version. We plan to introduce agreement constraints when the parser is compatible with unification. Determiners occurring without a head noun are also outside the scope of the grammar. For instance, *plusieurs* 'several' can be a syntactic variant of *plusieurs objets* 'several objects'. In that case, the deletion of the head noun is not accompanied by formal modifications of the determiner, but it is in other case, e.g. in *beaucoup* 'many' for *beaucoup d'objets* 'many objects'.

3.2 Method of construction of the grammar

The grammar has been developed manually from three existing RTNs (Gross, 2001), (Silberztein, 2003), (Constant, 2000). We removed from Silberztein's grammar two elements:

- the constraints involving the countable vs. uncountable feature of nouns, since this feature is absent from available lexicons of French;
- gender and number agreement; in Silberztein's grammar, agreement is represented by the existence of 4 versions of the grammar for the 4 combinations of the two genders and the two numbers; this redundancy makes the grammar difficult to maintain.

We introduced into the grammar various elements of Gross' and Constant's grammars. From Gross' grammar, we extracted lists of modifying adverbs, of negative adverbial determiners (e.g. *jamais de* 'never any'), of adjectives that can modify determinative nouns, and of adjectives with properties of determiners (e.g. *premier* 'first'). From Constant's grammar, we extracted the description of physical magnitudes and of approximate numerical expressions.

Then we enhanced the grammar with more constructions and more constraints, using the same two approaches as Gross, Silberztein and Constant to construct their grammars: the corpus-based bootstrapping method (Gross, 2000) and introspection. For example, we introduced combinations of adverbial determiners such as *un peu de* with adjectival determiners such as *chaque*. We also described constraints between successive determinative nouns, as in *trois sortes de parties de* 'three kinds of parts of'.

We mentioned above that the sequences described in the grammar are surface forms such as *au*, and not normalized forms such as *à le*. However, during the construction of the grammar, we

managed all the graphs in the normalized form, and we changed them to the surface form at the end of the construction, because this operation obfuscates considerably the grammar and makes it difficult to maintain. We saved the normalized version so that maintenance operations can be performed on it.

3.3 Classification of determinative nouns

The grammar of (Gross, 2001) contains a selection of about 20 frequent determinative nouns: *dose entièreté fraction groupe majorité maximum minimum minorité moitié morceau nombre paire part partie portion quantité restant reste sorte total totalité*. In addition to this list, it contains numbers: *dix* 'ten', and nouns derived from numbers: *dizaine* 'about ten', *dixième* 'one tenth'. The determinative nouns in (Silberztein, 2003) are a subset of those present in (Gross, 2001). We included all, and a few others such as *abondance catégorie classe couple ensemble espèce flopée floraison foison foisonnement foule kyrielle multitude myriade parcelle pléthore pourcentage profusion proportion ribambelle tas type volume*. The resulting list is far from exhaustive: (Buvet, 1994) mentions a list of about 3000 determinative nouns with a quantifying value, and (Buvet & Lim, 1996) deals with other determinative nouns with an aspectual value. However, our list was large enough to demonstrate that it was necessary to classify the determinative nouns before including them into the grammar. For example, these nouns impose different syntactic restrictions on the number of the head noun:

*Voici un (kilo + *groupe) de ce papier* 'Here is a (kg + *group) of this paper',

on its lexical value or countable feature:

*Voici un grand (nombre + *morceau) de bocaux* 'Here is a large (number + *piece) of bottles'
*Voici un grand (*nombre + morceau) de métal* 'Here is a large (*number + piece) of metal'

and on their own determiner:

*Voici trois (groupes + *quantités) de bocaux* 'Here are 3 (groups + *quantities) of bottles'

The typology of determinative nouns of (Buvet, 1994) was partly adequate for our purposes, but not entirely, because it does not take into account the immediate morpho-syntactic context of the nouns. This typology is based on paraphrases showing how the interpretation of the determinative noun is related to that of the same noun used in other contexts, as in *Ces bocaux forment trois groupes* 'These bottles make up three groups' or *Ce papier pèse un kilo* 'This paper weighs one kg'.

Thus, we devised another classification based on 5 syntactic properties directly exploitable in the grammar. Each property is the acceptability of a sentence type:

voici un Ndet:s (E + Modif) de ce N:s *Voici une partie de ce papier*
'Here is part of this paper'

voici Dnum:p Ndet:p de Det N	*Voici trois parties de ce bocal* 'Here are three pieces of this bottle'
voici un Ndet:s de Ncpt:s	*Voici une sorte de bocal* 'Here is a kind of bottle'
N_0 contenir un Ndet:s de Ncpt:p	*Ce mortier contient une partie de gravillons* 'This mortar contains a proportion of gravel'
N_0 être un Ndet:s de combien de Ncpt:p ?	*C'est un groupe de combien de personnes ?* 'This is a group of how many people?'

In these formulae, *:s* denotes the singular, *:p* the plural, *Modif* nominal modifiers, *Dnum* numeral determiners, and *Ncpt* countable nouns. The following decision tree distributes *Ndet* into 9 classes (the size of each class is given in parentheses):

voici un Ndet:s (E + Modif) de ce N:s	
voici Dnum:p Ndet:p de Det N	
voici un Ndet:s de Ncpt:s	
N_0 contenir un Ndet:s de Ncpt:p	*NdetPartie* (6)
** N_0 contenir un Ndet:s de Ncpt:p*	*NdetMorceau* (19)
** voici un Ndet:s de Ncpt:s*	*NdetMasse* (open)
** voici Dnum:p Ndet:p de Det N*	*NdetQuantité* (41)
** voici un Ndet:s (E + Modif) de ce N:s*	
voici Dnum:p Ndet:p de Det N	
voici un Ndet:s de Ncpt:s	*NdetSorte* (12)
** voici un Ndet:s de Ncpt:s*	
N_0 être un Ndet:s de combien de Ncpt:p ?	*NdetGroupe* (16)
** N_0 être un Ndet:s de combien de Ncpt:p ?*	*NdetDizaine* (32)
** voici Dnum:p Ndet:p de Det N*	*NdetNombre* (45)

Class *NdetMasse*, the largest, must be further divided into several subclasses, some of which are defined by (Buvet, 1994) as C2, C3, C4, C5, C6 and C13 (measurement units); and C7a, C7b, C8 and C9 (contents); the residual subclass contains *masse* 'mass'. The other classes of the decision tree above are in a complex relation with Buvet's typology.

3.4 Structure

The grammar is a network of 186 graphs. One of them id displayed in Fig. 1. There are 3 main graphs:

- *aDet* and *deDet* for determiners preceded respectively by the prepositions *à* and *de*,
- *Det* for determiners not preceded by prepositions or preceded by other prepositions.

The compilation of these main graphs produces automata with respectively 2143, 2223 and 2044 states. The grammar is strongly lexicalised: it contains 1206 lexical tokens.

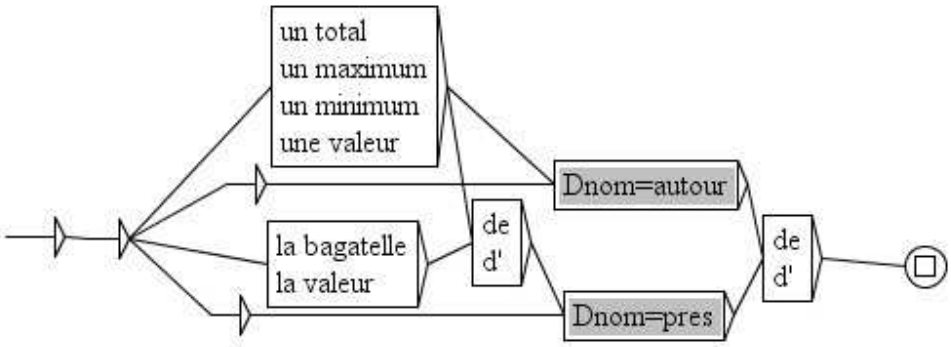


Figure 1: Graph 'Dnom=presDe' from the local grammar

4 Method of evaluation

Syntactic annotation of text is usually evaluated by comparison with reference treebanks. However, annotation derived from manually constructed grammars is often richer than the information found in golden standards. For example, the French Treebank (Abeillé & Barrier, 2004) analyses *J'ai appris un certain nombre d'exigences administratives* 'I got aware of a certain number of administrative requirements' with *nombre* 'number' as the head noun of the complement the verb. One of the motivations for building local grammars is the perspective of using them for the construction of treebanks with more informative syntactic-semantic annotation. Therefore, evaluation of our grammar by comparison to standard treebanks would have been inappropriate or even misleading.

In order to assess the quality of the grammar, we annotated a corpus with information on determiners, we ran the parser with the grammar on the raw version of the evaluation corpus, and we compared the output of a parser with the manual annotation. The evaluation corpus is made of journalistic texts from the newspaper *Le Monde* (1994). Its size is 8000 words.

4.1 Annotation guidelines

The evaluation corpus was annotated with XML tags in order to delimit the (generalised) determiners as defined in 3.1 above. The annotators were given the following guidelines.

Prepositions *à* 'to' and *de* 'of' immediately preceding a determiner are included in the delimited sequence. Other prepositions are not included. The XML tag is respectively <*ad*> or <*dd*> instead of <*d*> if the preposition was included. In case of a compound preposition ending in *de*, only the ending *de* is included in the sequence. For instance, *vis-à-vis de l'Est* 'towards the East' is annotated *vis-à-vis* <*dd*>*de l'*</*dd*>*Est*. When no determiner occurs between the preposition and the head noun, as in *un changement de concept* 'a change of concept', no annotation is inserted, except in two cases:

- if *de* is analysed as an indefinite determiner, as in *obtenir <d>de</d> meilleures conditions* 'obtain better conditions';

- if *de* is analysed as the surface form corresponding to an underlying sequence of the preposition *de* and an indefinite determiner, as in *sous <d>l'</d>effet <dd>de</dd> mesures draconiennes* 'under the effect of draconian measures'.

Numbers written in figures are annotated in the same conditions as numbers written in letters: *<d>100 000</d>, <d>dix</d>*.

Determiners occurring without a head noun are not annotated: compare *<d>peu de</d> temps* 'little time' with *Beaucoup semblent d'ailleurs commencer à la comprendre* 'Many indeed seem to begin to understand it'. Percentages not explicitly followed by a head noun are not annotated either: compare *qui couvre <d>environ 8 % des</d> besoins* 'that covers about 8% of the needs' with *accroitre de 40 % ses exportations* 'increase its exports by 40%'.

Determiners inside multi-word units are annotated only if they obey the general syntax of determiners. For example, the determiner is annotated in *<dd>de l'</dd>ordre de 9 à 10 %* 'about 9 to 10%', but not in *rectification d'ordre sémantique* 'correction of a semantic nature'.

The application of these guidelines led to the annotation of 1512 occurrences of determiners: 63% with *<d>*, 27% with *<dd>* and 11% with *<ad>*.

4.2 Parsing experiment

We ran a test transducer invoking the 3 main graphs *Det, aDet* and *deDet* on the raw, untagged version of the evaluation corpus, with the Unitex system (version 1.2). We wrote and used this test transducer, instead of using directly the 3 main graphs, in order to mitigate the influence of lexical ambiguity on the results. Since the evaluation corpus is not tagged for parts of speech, the parser matches variables with text words on the basis of their features found in the lexicons of the system. We used the Dela lexicons (Courtois, 1990). The large coverage of these lexicons tends to lower the precision in the recognition of words and constructions. However, this effect is mechanically mitigated by the length of the sequences described in local grammars: the longer the sequences, the smaller the influence of lexical ambiguity on precision. Since determiners employed without a noun were outside the scope of the experiment, we wrote a test transducer that associates a determiner (optionally preceded by *à* or *de*) with a core noun phrase composed of a noun preceded by optional adjectives, in turn preceded by optional adverbs. Thus, sequences recognised by the grammar are retained by the parser only if they are (immediately or not) followed by a word that can be a noun. These experimental conditions are fair, since the test transducer corresponds to the conditions of use of the grammar.

The test transducer produces an output text which consists of the input text with XML tags inserted before and after each determiner recognised by the local grammar. The XML tags identify whether the sequence was recognised by graph *Det, aDet* or *deDet*.

The parsing of the evaluation corpus on a Windows-XP PC took 12s, among which 10s were dedicated to the compilation of the grammar. With a Windows-2000 PC, Unitex parsed 20,452,000 words in 168 mn, which corresponds to 2029 words/second.

4.3 Comparison protocol

The annotation inserted by the parser was compared to the manual annotation. The annotation of a sequence in the two files was considered to agree only if both the opening tag and the closing tag occurred at the same place. Two comparisons were performed. In the first one, the three kinds of tags <*d*>, <*ad*> and <*dd*> were confused: for example, an annotation with <*d*> in the output of the parser was considered to agree with an annotation of the same sequence with <*dd*> in the reference corpus. In the second comparison, two annotations were considered to agree only if the value of the tag was the same.

5 Results

We computed the precision (proportion of sequences annotated in the reference corpus among those annotated by the parser) and the recall (proportion of sequences annotated by the parser among those annotated in the reference corpus). The results of the comparison are displayed in Table 1. The 'All' column corresponds to the comparison in which the three kinds of tags are considered equal.

	All	*Det*	*aDet*	*deDet*
Precision	86%	72%	97%	35%
Recall	92%	93%	91%	20%

Table 1. Comparison between parser output and manual annotation

These results show that the grammar is able to detect determiners with some accuracy, even on text which is not tagged for parts of speech. However, the grammar cannot discriminate whether a determiner is preceded by the preposition *de* or not. This is not a surprise, since the surface form *de* can be analysed either as a preposition, or as a determiner, or as a combination of a preposition and a determiner, and the choice depends on syntactic context.

6 Conclusion

We evaluated the quality of a grammar of French determiners by comparison with an independently annotated corpus. The application of the grammar gives deeper syntactic information than chunking or information available in treebanks: in particular, it contributes to a more accurate detection of heads of noun phrases. The grammar achieves 86% precision and 92%

recall. The analysis of errors showed directions for improvement of both figures. These facts suggest that the local grammar is worth using as a component of a deep syntactic grammar of French.

Acknowledgments

This work has been supported by CNRS. I thank Anastasia Yannacopoulou for her valuable contribution to the combination of Gross' and Silberztein's grammars.

References

Abeillé, A. & Barrier, N. 2004. “Enriching a French Treebank”, Lino et al. (eds.), Proceedings of the International Conference on Language Resources and Evaluation (LREC), Lisbon.

Blanc, O. & Constant, M. 2005. “Lexicalisation of grammars with parameterized graphs”, Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), Borovets (Bulgaria), p. 117-121.

Buvet, P.-A. 1994. “Détermination : les noms”, *Lingvisticae Investigationes* 18:121-150.

Buvet, P.-A. & Lim, J. 1996. “Les déterminants nominaux aspectuels”, *Lingvisticae Investigationes* 20:271-285.

Constant, M. 2000. “Description d'expressions numériques en français”, *Revue Informatique et Statistique dans les Sciences Humaines* 36:119-135.

Courtois, B. 1990. “Un système de dictionnaires électroniques pour les mots simples du français”, *Langue française* 87:11-22.

Danlos, L. 2005. “Automatic Recognition of French Expletive Pronoun Occurrences”, Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), Companion Volume, p. 73-78, Jeju, Korea.

Fairon, C., Paumier, S. & Watrin, P. 2005. “Can we parse without tagging? ”, Proceedings of the Language & Technology Conference: Human Language Technologies, Poznan, Poland, p. 473-477.

Gross, M. 1998-1999. “Lemmatization of Compound Tenses in English”, *Lingvisticae Investigationes* 22:71-122, Amsterdam/Philadelphia: Benjamins.

Gross, M. 2000. “A Bootstrap Method for Constructing Local Grammars”, Bokan, N. (ed.), Proceedings of the Symposium on Contemporary Mathematics, University of Belgrad, Serbia, p. 229-250.

Gross, M. 2001. "Grammaires locales de déterminants nominaux", *Détermination et formalisation*, LIS 23, Amsterdam/Philadelphia: Benjamins, p.177-193.

Laporte, E., Ranchhod, E. & A. Yannacopoulou 2006. "Syntactic variation of support verb constructions", Proceedings of the Lexis and Grammar Conference (LGC), Palermo, Italy.

Mason, O. 2004. "Automatic Processing of Local Grammar Patterns", Proceedings of the Annual Colloquium for the UK Special Interest Group for Computational Linguistics, Birmingham, p.166-171.

Nam, J. & Choi, K. 1997. "A Local-Grammar-based Approach to Recognizing of Proper Names in Korean Texts". Zhou & Church (eds.), Proceedings of the Workshop on Very Large Corpora, ACL/Tsing-hua University/Hong-Kong University of Science and Technology, p. 273-288.

Nenadic, G. 2000. "Local Grammars and Parsing Coordination of Nouns in Serbo-Croatian", Proceedings of Text, Speech and Dialogue (TSD), LNAI 1902, Springer, p. 57-62.

Paroubek, P., Robba, I., Vilnat, A. & Ayache, Ch. 2006. "Data, Annotations and Measures in EASY, the Evaluation Campaign for Parsers of French", Proceedings of the International Conference on Language Resources and Evaluation (LREC), Genoa, Italy.

Paumier, S. 2006. *The Unitex Manual*. http://igm.univ-mlv.fr/~unitex/.

Poibeau, Th. 2006. "Dealing with Metonymic Readings of Named Entities", Proceedings of the Annual Conference of the Cognitive Science Society (COGSCI), Vancouver, Canada.

Ranchhod, E., Carvalho, P., Mota, C. & Barreiro, A. 2004. "Portuguese Large-scale Language Resources for NLP Applications", Lino et al. (eds.), Proceedings of the International Conference on Language Resources and Evaluation (LREC), Lisbon, p.1755-1758.

Saetre, R. 2004. "GeneTUC - BioMolecular Information Retrieval", Computer Science Graduate Student Conference (CSGSC), Trondheim, Norway.

Senellart, J., Plitt, M., Bailly, Ch. & Cardoso, F. 2001. "Resource alignment and implicit transfer", Machine translation in the information age, MT Summit, p. 317-323.

Silberztein, M. 2003. "Finite-State Description of the French Determiner System", *Journal of French Language Studies* 13(2):221-246, Cambridge University Press.

Venkova, T. 2000. "A local grammar disambiguator of compound conjunctions as a pre-processor for deep analysers", Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, European Summer School in Logic, Language and Information (ESSLLI), Birmingham.

Structure de graphe de la base lexicale DiCo

Alain Polguère

OLST — Département de linguistique et de traduction,
Université de Montréal
C.P. 6128, succ. Centre-ville,
Montréal (Québec) H3C 3J7 Canada
alain.polguere@umontreal.ca

Résumé

Nous introduisons un nouveau type de structure appelé *système lexical*, qui est une architecture flexible pour les bases de données lexicales pouvant être intégrée aux systèmes de traitement automatique de la langue. Nous commençons par une caractérisation formelle des systèmes lexicaux en tant que graphes orientés simples, uniquement composés de nœuds correspondant aux entités lexicales et de liens connectant ces nœuds. Pour illustrer cette approche, nous présentons des données empruntées à un système lexical qui a été généré à partir de la base DiCo des dérivations sémantiques et collocations du français. Nous expliquons ensuite comment a été rendue possible la compilation de la base de données originelle dictionnairique en une structure de type graphe. Finalement, nous discutons du potentiel de la structure de données proposée pour la construction de ressources multilingues[1].

Mots-clés

base de données lexicale, modèle de type graphe, Lexicologie Explicative et Combinatoire, DiCo, fonction lexicale

1 Structure des systèmes lexicaux

Le but de cet article est d'introduire, justifier et illustrer un nouveau type de structuration des données lexicales permettant la construction de ressources appelées *systèmes lexicaux*. Notre présentation s'appuie sur une expérimentation visant la génération d'un système lexical à partir de tables extraites de la base de données DiCo. Cette expérimentation nous a permis de produire un système lexical qui est une structure plus riche que la base de données originelle dont elle est dérivée.

---

[1] Cet article est une adaptation française de Polguère (2006).

Les systèmes lexicaux, en tant que modèles formels des lexiques des langues naturelles, sont proches des bases de données de la famille *-Net*, dont le représentant le plus connu est sans aucun doute WordNet (Fellbaum, 1998). Cependant, les systèmes lexicaux possèdent des caractéristiques très particulières, qui les distinguent radicalement des autres structures lexicographiques. Dans cette section, nous commençons par présenter les deux approches principales de la structuration des modèles lexicaux, pour ensuite introduire les systèmes lexicaux de façon contrastive.

1.1 Bases de données lexicales dictionnairiques *vs* de la famille *-Net*

1.1.1 Nature textuelle des bases de données dictionnairiques

La façon la plus directe de construire des bases de données lexicales est d'utiliser des dictionnaires standard (c'est-à-dire des livres) et de les transformer en entités informatiques. C'est l'approche adoptée par la plupart des éditeurs de dictionnaires électroniques, avec divers niveaux de sophistication. Les produits qui résultent de cette informatisation peuvent être appelés *bases de données dictionnairiques*. De telles bases possèdent deux caractéristiques principales :

- Elles sont constituées d'une collection de descriptions de vocables, appelées *entrées de dictionnaires*.
- Leurs entrées peuvent être vues comme des « textes », au sens le plus général.

En conséquence, les bases de données dictionnairiques sont avant tout d'énormes textes, constitués d'une succession de textes beaucoup plus petits (les entrées).

Il semble naturel de voir les versions électroniques des dictionnaires standard comme étant des textes. Cependant, les bases lexicales formelles telles que la base multilingue JMDict (Breen, 2004), balisée en XML, sont aussi de nature textuelle. Ce sont des successions d'entrées, chaque entrée étant constituée d'un texte structuré qui « nous dit quelque chose » à propos d'un vocable (ou d'une acception de vocable, selon la perspective). Même les bases de données qui encodent des modèles relationnels du lexique peuvent être à 100% textuelles, et donc dictionnairiques. Tel est le cas de la base DiCo du français (Polguère, 2000a et 2000b), que nous avons utilisée pour compiler le système lexical qui nous servira ici d'illustration. Comme nous allons le voir plus bas, la base DiCo originelle n'est rien d'autre qu'une collection de fiches, chaque fiche étant subdivisée en champs, qui sont eux-mêmes en réalité de courts textes. Bien que le DiCo soit conçu dans le cadre de la Lexicologie Explicative et Combinatoire (Mel'čuk et coll., 1995) et se concentre sur la modélisation des liens lexicaux, il n'est clairement pas conçu comme une base de données de la famille *-Net*, comme nous le verrons plus loin.

1.1.2 Structure de graphe des bases de données de la famille **-Net**

La plupart des modèles lexicaux, même les dictionnaires standard, sont de nature relationnelle. Ainsi, tous les dictionnaires définissent les unités lexicales en termes d'autres unités lexicales et ils utilisent des pointeurs lexicaux du type « Synonyme » et « Antonyme ». Cependant, leur structure formelle ne reflète pas cette nature relationnelle. La situation est tout à fait différente avec les bases de données de la famille *-Net*. Ces dernières peuvent être caractérisées de la façon suivante :

- Ce sont des graphes — d'immenses ensembles d'entités connectées — plutôt que des collections de petits textes (les entrées).
- Elles ne sont pas nécessairement focalisées sur la description des vocables ou de leurs acceptions. Les nœuds de leur structure de graphe forment un ensemble potentiellement hétérogène d'entités lexicales ou, de façon plus générale, d'entités linguistiques.

Les bases de données de la famille *-Net* s'appuient vraisemblablement sur le type de représentation des connaissances le plus approprié pour la modélisation des lexiques. Cependant, elles posent un problème majeur : elles sont, dans la plupart des cas, structurées en fonction d'un petit ensemble de principes de hiérarchisation ou de classification. WordNet, par exemple, est avant tout de nature sémantique et impose une organisation hiérarchique des entités lexicales fondée, avant tout, sur deux relations sémantiques bien déterminées : la synonymie — à travers le regroupement des sens lexicaux au sein de *synsets* — et l'hyperonymie. De plus, la classification des unités lexicales par parties du discours crée une partition stricte de la base de données : WordNet est constitué de quatre hiérarchies de synsets distinctes et autonomes (pour les noms, les verbes, les adjectifs et les adverbes). Nous ne pensons pas que les modèles lexicaux doivent être conçus en fonction de quelques principes rigides qui imposent une hiérarchisation ou une classification des données. Ce type de structuration est bien entendu utile, même nécessaire, mais il devrait être projeté « sur demande » sur les modèles lexicaux. De plus, il ne devrait pas y avoir un ensemble prédéfini et fini de principes structuraux ; les structures de données devraient être capables d'accueillir n'importe quel type d'organisation, selon les besoins, et c'est justement une des caractéristiques principales des systèmes lexicaux présentées ci-dessous (section 1.2). Remarquons que le nouveau type de bases lexicales de la famille *-Net* que nous allons proposer présente certaines similarités avec d'autres structures adoptées en lexicographie informatisée (Trippel, 2006).

1.1.3 Structures textuelles vs *structures de graphes : les pour et les contre*

Remarquons tout d'abord que toute base de données dictionnairique peut-être compilée sous forme de base de données de la famille *-Net*, et *vice versa*. Bien entendu, les bases dictionnairiques qui s'appuient sur des modèles relationnels sont plus compatibles avec un encodage sous forme de graphe. Il y a cependant toujours des données relationnelles dans les dictionnaires, et ce type de données peut être extrait et encodé sous forme de nœuds et de liens les connectant. La véritable question n'est donc pas choisir entre deux structures mutuellement exclusives ; elle concerne la détermination de ce que chaque structure permet le mieux de faire. Selon nous, le potentiel de chaque type de structure peut être caractérisé de la façon suivante.

Les structures dictionnairiques sont des outils pour éditer (c'est-à-dire, rédiger) et consulter les descriptions lexicales. L'intuition linguistique des lexicographes ou des utilisateurs des modèles lexicographiques fonctionne bien mieux à partir de textes. Les lexicographes aussi bien que les utilisateurs ont besoin d'avoir une vue d'ensemble sur chaque unité lexicale et ont besoin du format d'entrée textuelle lorsqu'ils accèdent aux données lexicales. Bien entendu, d'autres façons de présenter l'information lexicale — comme la présentation tabulaire — peuvent aussi s'avérer nécessaires[2].

[2] On ne se surprendra pas de voir que les *lexicographer files* de WordNet offrent une perspective textuelle sur les items lexicaux, qui est tout à fait dictionnairique. L'unité de description est cependant le synset, et non l'unité lexicale. (Voir dans la documentation de WordNet, la présentation des *lexicographer files*.)

Les structures de *-Net* sont des outils pour implémenter la « dynamique lexicale » : navigation dans l'information lexicale, ajouts, révisions ou inférences de nouveau contenu informationnel à partir de l'information préexistante. À cause de cela, beaucoup pensent que les bases de la famille *-Net* possèdent une forme validité cognitive. Finalement, et cela n'est pas la moindre des caractéristiques, les bases de la famille *-Net* peuvent plus facilement absorber d'autres structures lexicales ou être absorbées par elles.

En conclusion, bien que les deux types de structures présentent un certain niveau de compatibilité et présentent des avantages propres dans des contextes d'utilisation bien déterminés, nous sommes particulièrement intéressé par le fait que les bases de la famille *-Net* sont plus disposées à vivre une « vie organique » en termes d'évolution (addition, soustraction, remplacement) et d'interaction avec d'autres structures de données (connexion avec des modèles d'autres langues, avec les grammaires, etc.).

1.2 Les systèmes lexicaux : un nouveau type de bases de données lexicales de la famille *-Net*

Tel qu'indiqué plus haut, la plupart des bases de données de la famille *-Net* semblent se focaliser sur la description de seulement quelques propriétés des lexiques des langues naturelles (quasi-synonymie, organisation hyperonymique des sens lexicaux, structures prédicatives avec leur expression au niveau syntaxique, etc.). En conséquence, les auteurs de ce type de bases sont souvent acculés à « étirer » à la limite la capacité descriptive de leurs modèles afin d'y ajouter la description de nouveaux types de phénomènes, dont ils ne se préoccupaient pas lorsque le design de leurs bases a été élaboré. On peut s'attendre à ce que ce type de greffe de nouveaux composants laisse des cicatrices sur le design initial des modèles lexicaux. Les structures lexicales que nous proposons, les systèmes lexicaux (dorénavant, *SL*), ne posent pas ce type de problème, pour deux raisons. Tout d'abord, ils ne sont pas bridés par le fait qu'ils ne se concentreraient que sur la description de quelques phénomènes lexicaux bien déterminés ; ils sont au contraire motivés par une vision globale du lexique en tant que composante centrale de la connaissance linguistique. Ensuite, ils possèdent une organisation très simple et « plate », qui n'impose aucune structure hiérarchique ou classifiante *a priori*. Expliquons comment tout cela fonctionne.

Le design de tout SL doit respecter quatre principes de base, qui ne peuvent en aucun cas être enfreints. Nous allons brièvement examiner chacun de ces principes.

1. Simple structure de graphe orienté. Un SL est un graphe orienté, et rien d'autre. Cela signifie que, d'un point de vue formel, il est uniquement constitué de nœuds et de liens orientés connectant ces nœuds.

2. Caractère non hiérarchique. Un SL est une structure non hiérarchique, bien qu'il puisse contenir un ensemble de nœuds hiérarchiquement connectés. Par exemple, nous verrons plus bas que le SL DiCo contient des nœuds qui correspondent à un ensemble hiérarchiquement organisé d'étiquettes sémantiques ; mais le SL lui-même n'est en aucun cas organisé en fonction d'une ou plusieurs hiérarchies particulières.

3. Hétérogénéité. Un SL est un ensemble hétérogène de nœuds. On peut trouver trois familles principales de nœuds en son sein : (i) les entités lexicales véritables telles que les lexèmes, les

locutions, les mots-formes, etc. ; (ii) les entités quasi lexicales, telles que les collocations, les fonctions lexicales[3], les expressions libres dont le stockage dans la base présente un intérêt particulier (par exemple, les exemples lexicographiques) ; (iii) les entités lexicogrammaticales, telles que les affixes, les régimes, etc.

Les nœuds de SL typiques sont avant tout des entités lexicales, mais il faut prévoir que les SL peuvent contenir en tant que nœuds des entités qui appartiennent à l'interface entre le lexique et la grammaire de la langue. Tel est le cas des cadres de sous-catégorisation, appelés *schémas de régime* dans la Lexicologie Explicative et Combinatoire. En tant que règles spécifiant des patrons de structures syntaxiques, ils appartiennent à la grammaire de la langue. Cependant, en tant qu'entités formelles préconstruites sur lesquelles viennent se fixer les lexèmes dans la phrase, ils relèvent bien plus du domaine lexical que de véritables règles grammaticales, telles que les règles de construction du passif, d'accord grammatical, etc.

4. Caractère flou (angl. *fuzziness*). Chaque composant d'un SL, que ce soit un nœud ou un lien, porte une valeur de confiance, c'est-à-dire une mesure de sa validité. Bien entendu, il existe bien des façons d'attribuer et manipuler les valeurs de confiance afin d'implémenter le flou en représentation des connaissances. Ainsi, dans nos expérimentations avec le SL DiCo, nous avons adopté une approche tout à fait simpliste, qui était suffisante pour nos présents besoins, mais qui devrait être grandement sophistiquée au fur et à mesure que nous avançons dans la construction et l'utilisation des SL. Dans notre implémentation présente, nous n'utilisons que trois valeurs de confiance : « `1` » signifie que pour autant qu'on puisse en juger — c'est-à-dire, en prenant pour argent comptant ce qui est explicitement encodé dans le DiCo — l'information est valide ; « `0.5` » signifie que l'information résulte d'une inférence faite à partir des données traitées en entrée dans le processus de compilation du DiCo sous forme de SL DiCo et n'était pas explicitement encodée en tant que telle par les lexicographes du DiCo ; « `0` » signifie que l'information doit être non valide — par exemple, dans le cas où la compilation du DiCo nous a permis d'identifier un renvoi non valide à une unité lexicale dans les données importées.

L'encodage du flou est un trait essentiel des SL, en tant que structures sur lesquelles peuvent s'effectuer des inférences ou en tant que structures qui sont, au moins en partie, inférées d'autres structures (dans le cas de la génération de SL à partir de bases lexicales existantes). Bien entendu, toute valeur de confiance n'est pas une valeur absolue. « `1` » ne signifie pas que l'information est valide quoiqu'il advienne, et « `0` » qu'elle est nécessairement fausse. L'information contenue dans les SL, et l'évaluation de cette information, n'est pas plus absolue que n'importe qu'elle information qui peut être stockée dans le lexique mental d'un individu. Cependant, si nous voulons qu'il soit possible d'effectuer des calculs sur le contenu des SL, il est essentiel que nous puissions distinguer entre les données dont nous avons toutes les raisons de penser qu'elles sont valides et celles dont nous avons toutes les raisons de penser qu'elles ne le sont pas.

Il est maintenant grand temps de donner des exemples concrets de données contenues dans un SL. Mais auparavant, insistons sur le fait qu'aucun appareillage formel autre que ce que nous venons de présenter n'est autorisé dans les SL. Tout autre élément formel que nous pourrions vouloir ajouter doit relever des autres composants du modèle linguistique, de la grammaire par exemple.

[3] Pour les notions de collocations et de fonctions lexicales, voir la section 2 ci-dessous.

Remarquons, cependant, que nous n'excluons pas la nécessité d'introduire une mesure du « poids relatif » des nœuds et des arcs. Cette mesure, distincte de la valeur de confiance, refléterait le degré d'activation de chaque élément du SL. Par exemple, l'article de DiCo de DÉFAITE énumère un assez grand nombre de verbes supports prenant ce nom comme complément, parmi lesquels CONNAÎTRE and SUBIR (*une défaite*). La pondération pourrait indiquer que le premier verbe est utilisé de façon beaucoup moins courante (ou libre) que le second dans ce contexte particulier.

2 Exemples empruntés au SL DiCo

Le DiCo est une base de données lexicale du français qui met l'accent sur la modélisation des liens paradigmatiques et syntagmatiques contrôlés par les unités lexicales. Les liens paradigmatiques correspondent à ce que la Lexicologie Explicative et Combinatoire appelle les liens de *dérivation sémantique* (synonymie, antonymie, verbalisation, nominalisation, nom d'actant ou de circonstant typique, etc.). Les liens syntagmatiques correspondent aux collocations contrôlées par les unités lexicales (intensificateurs, verbes supports, etc.). Ces propriétés lexicales sont encodées au moyen d'un système d'entités métalexicales appelées *fonctions lexicales*. Pour une présentation du système des fonctions lexicales, on pourra se reporter à Mel'čuk (1996) et Kahane & Polguère (2001). Bien qu'il ne contienne pas de définitions véritables, le DiCo décrit de façon partielle le contenu sémantique de chaque unité lexicale au moyen de deux outils formels complémentaires : (i) une étiquette sémantique, qui correspond au genre prochain (composant central) de la définition de l'unité lexicale et (ii) une forme propositionnelle, qui décrit la nature prédicative de l'unité (sens non prédicatif ou prédicat à un, deux ou plus actants). Chaque article spécifie aussi le schéma de régime (en gros, le cadre de sous-catégorisation) de l'unité lexicale et énumère les locutions qui la contiennent formellement. Finalement, chaque article présente un ensemble d'exemples extraits des corpus ou d'Internet. Comme on le voit, le DiCo couvre une part importante des propriétés intrinsèques de chaque unité lexicale ; pour une présentation plus détaillée du DiCo, on pourra consulter Polguère (2000a et 2000b).

Pour l'instant, le DiCo est élaboré sous la forme d'une base de données FileMaker®. Chaque article de DiCo correspond à une fiche (un enregistrement) dans la base de données, et le noyau de chaque fiche est le champ qui contient la description des liens de fonctions lexicales contrôlés par le mot-clé (c'est-à-dire, l'unité lexicale décrite dans la fiche). Les données en (1) ci-dessous correspondent à un élément du champ de fonctions lexicales de la fiche DiCo pour RANCUNE.

(1)
```
      /*[X] éprouver ~*/
{Oper12} avoir, éprouver, nourrir, ressentir [ART ~ Prép-envers N=Y]
```

On peut isoler dans l'exemple ci-dessus cinq types distincts d'entités à encoder dans un SL.

- `Oper12` est le nom d'une fonction lexicale désignant un type particulier de verbes supports[4].
- `{Oper12}`, pris comme un tout, dénote `Oper12`(RANCUNE), c'est-à-dire l'application de la fonction lexicale `Oper12` à son argument (le mot-clé de la fiche).

[4] Plus précisément, `Oper12` dénote les verbes supports qui prennent le premier actant du mot-clé comme sujet, le mot-clé lui-même comme premier complément et le deuxième actant du mot-clé comme deuxième complément ; par exemple : *X éprouve de la rancune envers Y.*

- La formule qui précède — incluse entre les deux symboles `/*…*/` — est la *formule de vulgarisation* de `Oper12`(RANCUNE). Cet encodage métalinguistique du contenu exprimé par l'application de la fonction lexicale est destiné à l'utilisateur du DiCo qui ne maîtrise pas le système formel des fonctions lexicales.
- Immédiatement après le nom de la fonction lexicale vient la liste des valeurs de l'application de la fonction, chaque élément de valeur étant une entité lexicale donnée. Dans le cas présent, il s'agit de collocatifs du mot-clé, du fait de la nature syntagmatique de `Oper12`.
- L'expression entre crochets encode la structure syntaxique contrôlée par les collocatifs. Elle correspond à une entité lexicogrammaticale. Ce type d'entité n'a pas été traité dans notre expérimentation sur le SL DiCo et il n'en sera pas question dans la suite de la discussion.

Les données présentes en (1) correspondent à un minuscule sous-graphe du SL global que nous avons généré, sous-graphe qui est visualisé dans la Figure 1 ci-dessous. Notons que les représentations graphiques que nous utilisons ici ont été générées automatiquement en format GraphML à partir du SL, puis affichées au moyen de l'éditeur de graphes yEd.

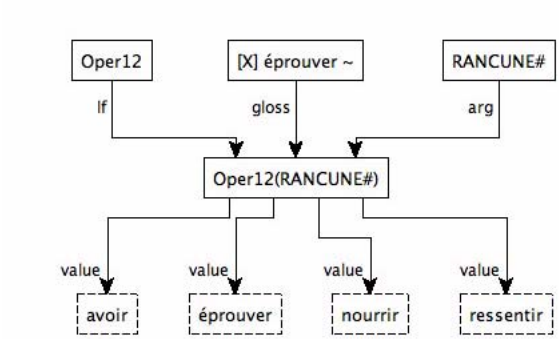


Figure 1: Interprétation sous forme de SL des données de (1)

Ce graphe montre comment les données du DiCo présentées en (1) ont été modélisées dans le SL DiCo en termes d'entités lexicales et de liens. On voit ici qu'une application de fonction lexicale est elle-même une entité lexicale : un contenu à communiquer, qui pointe vers des moyens d'expression de ce contenu. L'argument de la fonction lexicale (lien `arg`) — l'unité lexicale RANCUNE — est bien entendu aussi une entité lexicale (bien que de nature toute différente). Cela vaut aussi pour les valeurs (liens `value`). Aucune de ces valeurs, cependant, n'a été diagnostiquée comme possédant un article correspondant dans le DiCo. En conséquence, le processus de compilation leur a attribué de façon temporaire le statut de simples mots-formes, avec une valeur de confiance de `0.5`, visualisée ici au moyen de boîtes aux bordures discontinues. (Les lignes continues, pour les liens ou les boîtes, indiquent une valeur de confiance de `1`.) Il reviendra aux lexicographes d'ajouter au DiCo des articles pour les sens correspondants de AVOIR, ÉPROUVER, NOURRIR et RESSENTIR.

On pourrait être surpris de voir les fonctions lexicales (telles que `Oper12`) apparaître comme entités lexicales dans notre SL, du fait de leur nature très « abstraite ». Deux arguments justifient ce choix descriptif. Premièrement, les unités lexicales sont aussi des entités de nature plutôt abstraite. Alors que les mots-formes *cheval* et *chevaux* pourraient être considérés comme plutôt « concrets », leur regroupement au sein d'une unité lexicale CHEVAL représente une forme d'abstraction qui est loin d'être triviale. Deuxièmement, les fonctions lexicales ne sont pas que des

outils descriptifs dans la Lexicologie Explicative et Combinatoire. Elles sont aussi conceptualisées comme des généralisations d'unités lexicales, qui jouent un rôle très important dans la production du texte, dans les règles générales de paraphrasage par exemple.

Notre première illustration démontre comment la matérialisation sous forme de SL du DiCo reflète la véritable nature relationnelle de ce dernier, contrairement à sa matérialisation dictionnairique originelle sous forme de base de données FileMaker. Cette illustration montre aussi à quel point les entités lexicales peuvent être de natures variées et comment les valeurs de confiance peuvent nous aider à rendre explicite la distinction entre ce qui a été explicitement modélisé par les lexicographes et ce qui peut être inféré de leur modélisation.

L'illustration suivante va s'appuyer sur la précédente et montrer comment les fonctions lexicales dites *non standard* sont intégrées dans le SL. Jusqu'à présent, nous n'avons parlé que de fonctions lexicales standard, c'est-à-dire de fonctions lexicales qui appartiennent au petit noyau universel de relations lexicales identifié par la Lexicologie Explicative et Combinatoire (ou, de façon plus globale, par la théorie Sens-Texte). Cependant, tous les liens lexicaux paradigmatiques ou syntagmatiques ne sont pas nécessairement standard. En voici une illustration, empruntée à l'article de DiCo de l'unité lexicale CHAT.

(2) `{Ce qu'on dit pour appeler ~} « Minet ! », « Minou ! », « Petit ! »`

Ici, une fonction lexicale entièrement non standard, `Ce qu'on dit pour appeler ~`, a été utilisée pour connecter le mot-clé CHAT aux expressions du type *Minou !* Comme on peut le voir, aucune formule de vulgarisation n'a été introduite, et cela, parce que les noms de fonctions lexicales non standard représentent déjà un encodage explicite, non formel des relations lexicales. L'interprétation de (2) sous forme de SL est de ce fait une structure plus simple que celle utilisée dans l'illustration précédente, comme le montre la Figure 2.

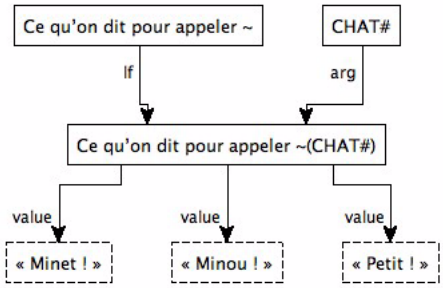


Figure 2: Interprétation sous forme de SL des données de (2)

La dernière illustration, ci-dessous, va montrer comment il est possible de projeter une structuration sémantique sur le SL DiCo lorsque cela s'avère nécessaire, et seulement à ce moment-là.

La hiérarchie des étiquettes sémantiques utilisées pour caractériser sémantiquement les unités lexicales du DiCo a été compilée dans le SL DiCo en même temps que la base lexicale elle-même. Chaque étiquette sémantique est connectée à son ou ses étiquettes plus génériques (la hiérarchie permettant l'héritage multiple) au moyen d'un lien `is_a`. De plus, elle est connectée aux unités lexicales qu'elle étiquette au moyen du lien `label`. Il est donc possible de tout simplement

extraire la hiérarchie des étiquettes du SL et « pêcher » par là même toutes les unités lexicales du SL, organisées hiérarchiquement par la relation d'hyperonymie. Notons que cette procédure est toute différente de celle consistant à extraire du DiCo toutes les unités lexicales possédant une étiquette donnée : nous extrayons ici toutes les unités dont l'étiquette sémantique appartient à une sous-hiérarchie donnée du système des étiquettes sémantiques. La Figure 3 ci-dessous montre le résultat de l'extraction de la sous-hiérarchie dominée par l'étiquette `accessoire`. Pour éviter d'avoir à utiliser des étiquettes sur les arcs, nous avons programmé la génération de cette classe de structures GraphML avec les liens encodés de la façon suivante : les liens `is_a` (entre étiquettes sémantiques) apparaissent sous forme de flèches continues en gras et les liens `label` (entre les étiquettes sémantiques et les lexies qu'elles étiquettent) apparaissent comme des flèches en pointillés.

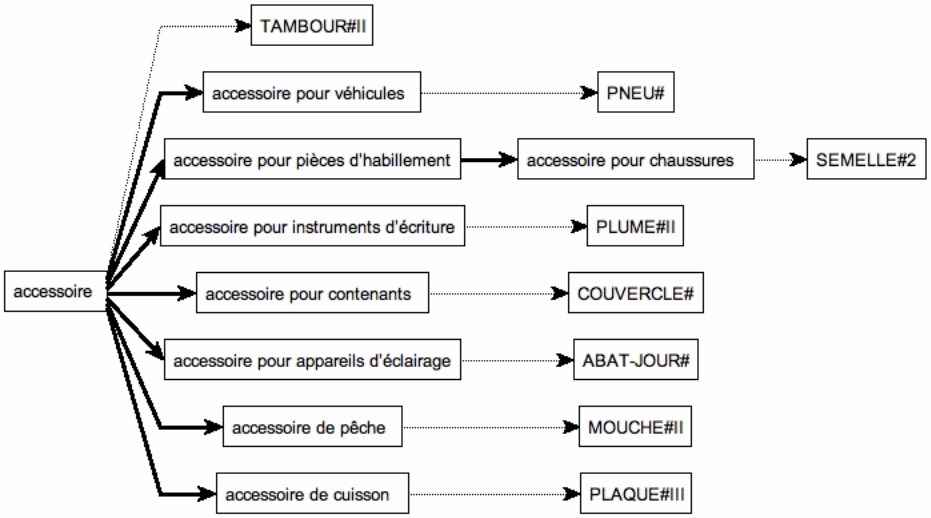


Figure 3: La sous-hiérarchie sémantique `accessoire` dans le SL DiCo

Toute la « beauté » de la structuration des SL ne réside pas dans le fait qu'elle nous permet de générer automatiquement d'élégantes représentations graphiques. De telles représentations ne sont qu'une façon commode de rendre explicite la structure interne des SL. Ce qui nous intéresse véritablement, c'est ce que l'on peut faire avec les SL une fois qu'on les considère dans une perspective fonctionnelle.

L'avantage fonctionnel principal des SL est que ces structures sont à la fois cannibales et toutes disposées à être cannibalisées. Expliquons les deux facettes de cette métaphore un peu morbide.

Premièrement, les graphes orientés sont des structures très puissantes, qui peuvent virtuellement encoder n'importe quel type d'information et sont particulièrement adaptés pour les connaissances lexicales. Si l'on est convaincu qu'un lexique est avant tout une entité relationnelle, on peut postuler que toute information présente dans n'importe quelle forme de dictionnaire ou de base de données peut éventuellement être compilée dans une structure de SL. L'expérimentation que nous avons pratiquée en compilant le DiCo (voir les détails dans la section suivante) démontre bien cette propriété des structures de SL.

Deuxièmement, à cause de leur extrême simplicité formelle, les structures de SL peuvent à l'inverse toujours être « digérées » par d'autres types de structures plus contraintes, telles que les

versions XML des bases dictionnairiques ou de la famille *-Net*. Par exemple, nous avons regénéré à partir du SL un DiCo en format HTML (avec des liens hypertextuels pour les références croisées entre articles et un encodage couleur des valeurs de confiance associées à l'information linguistique). Il est intéressant de remarquer que ce produit dérivé HTML du SL DiCo contient des articles qui n'existaient pas dans le DiCo originel. Ils sont produits pour chaque valeur d'application de fonction lexicale qui ne correspond par à un article de DiCo préexistant. Le contenu des ces articles est constitué de liens de fonctions lexicales « inverses » : des pointeurs vers des applications de fonctions lexicales pour lesquelles l'entité lexicale en question est une valeur. Ces nouveaux articles peuvent être vus comme des ébauches grossières, qui pourront être utilisées par les lexicographes afin d'écrire de nouveaux articles. Nous en dirons plus à ce sujet à la fin de la prochaine section.

3 Compilation de la base dictionnairique DiCo sous forme de système lexical

Le DiCo est pour l'instant disponible soit en format FileMaker soit sous forme de tables SQL, qui peuvent être notamment consultées au moyen de l'interface DiCouèbe[5]. Ce sont ces tables qui ont été utilisées comme données d'entrée pour la génération du SL DiCo[6]. Elles présentent l'avantage d'être déjà le résultat d'un traitement du DiCo qui découpe son contenu en segments élémentaires d'information lexicographique (Steinlin et coll., 2005). Il est donc plus aisé d'analyser ces tables de façon un peu plus poussée pour effectuer la restructuration sous forme de modèle de type SL.

La tâche consistant à inférer de l'information nouvelle, de l'information qui n'est pas explicitement encodée dans le DiCo, est la partie délicate du processus de compilation, du fait de la grande richesse de la base de données. Pour l'instant, nous n'avons implémenté qu'un petit sous-ensemble de toutes les inférences pouvant être effectuées. Par exemple, nous avons inféré des lexèmes (unités monolexicales) à partir des locutions apparaissant à l'intérieur des fiches DiCo. (Le nom de locution COUP DE SOLEIL implique l'existence probable de trois lexèmes COUP, DE et SOLEIL.) Nous avons aussi distingué les entités lexicales qui sont de véritables unités lexicales de celles qui sont des signifiants (des formes linguistiques). Les signifiants, qui n'ont pas nécessairement à être associés avec un unique sens, jouent un rôle important dans le contexte de la navigation à l'intérieur d'un SL (par exemple, lorsque l'on veut établir une distinction entre l'accès à une unité lexicale à partir de sa forme et à partir de son sens).

Nous ne pouvons donner ici tous les détails du processus de compilation. Mentionnons simplement le fait que, pour l'instant, de l'information importante contenue dans le DiCo n'a pas été traitée. Par exemple, nous n'avons pas encore implémenté la compilation des schémas de régime et des exemples lexicographiques. D'un autre côté, toutes les applications de fonctions lexicales et l'étiquetage sémantique des unités lexicales sont correctement traités. On se rappellera que nous importons, avec le DiCo, la hiérarchie des étiquettes sémantiques utilisée par les lexicographes du DiCo, ce qui nous permet de tisser des liens d'hyperonymie entre les unités lexicales, tel que mon-

---

[5] http://www.olst.umontreal.ca/dicouebe

[6] Le code permettant la compilation du DiCo en un SL, la génération des sorties graphiques en GraphML et la génération d'une version HTML du DiCo a été écrit en SWI-Prolog.

tré dans la Figure 3 ci-dessus[7]. Pour ce qui est du code proprement dit, le SL DiCo est tout simplement une base de données Prolog plate, comportant des clauses pour deux uniques prédicats :

```
entity( <Numerical ID>, <Name>, <Type>, <Trust> )
link( <Numerical ID>, <Source ID>, <Target ID>, <Type>, <Trust> )
```

Voici quelques statistiques sur le contenu du SL DiCo généré lors de notre expérimentation.

Nœuds : 37,808 au total

780 étiquettes sémantiques
1,301 vocables (= entrées dans la « nomenclature » du SL)
1,690 unités lexicales (= acceptions de vocables)
6,464 mots-formes
2,268 expressions non lexicalisées
7,389 signifiants monolexicaux
948 signifiants multilexicaux
3,443 fonctions lexicales
9,417 applications de fonctions lexicales
4,108 formules de vulgarisation pour les applications de fonctions lexicales

Liens : 61,714 au total

871 `is_a`, entre étiquettes sémantiques
775 `sem_label`, entre étiquettes sémantiques et unités lexicales
1,690 `sense`, entre vocables et unités lexicales qui sont leurs acceptions
2,991 `basic_form`, entre signifiants mono- ou multilexicaux et des vocables ou unités lexicales
6,464 `signifier`, entre mots-formes et signifiants monolexicaux
4,135 `used_in`, entre signifiants monolexicaux et signifiants multilexicaux
9,417 `lf`, entre fonctions lexicales et applications de fonctions lexicales
6,064 `gloss`, entre applications de fonctions lexicales et formules de vulgarisation
9,417 `arg`, entre applications de fonctions lexicales et unités lexicales (qui sont leur argument)
19,890 `value`, entre applications de fonctions lexicales et les éléments de valeur retournés

Nous allons maintenant faire quelques commentaires à propos de ces chiffres, afin de mieux faire comprendre comment fonctionne la génération du SL à partir de la base DiCo originelle.

La base de données FileMaker (ou SQL) du DiCo qui a été utilisée ne contenait que 775 fiches d'unités lexicales (acceptions de vocables). Cela est reflété dans les statistiques ci-dessus par le nombre de liens `sem_label` entre étiquettes sémantiques et unités lexicales : seules les unités lexicales qui étaient des mots-clés de fiches DiCo possèdent une étiquette sémantique. Les statistiques montrent que le SL DiCo contient en tout 1,690 unités lexicales. D'où viennent donc les 915 (1,690 – 775) unités lexicales additionnelles ? Elles ont été extrapolées à partir du champ *phraséologie* (`ph`) des fiches DiCo, où les lexicographes énumèrent les locutions qui sont formellement construites à partir des mots-clés des fiches. Par exemple, la fiche DiCo pour BARBE contient (entre autres) un pointeur vers la locution BARBE À PAPA dans son champ `ph`. Cette locution ne

[7] La hiérarchie des étiquettes sémantiques est construite au moyen de l'éditeur d'ontologies Protégé. Nous utilisons un fichier d'exportation XML de Protégé pour injecter cette hiérarchie à l'intérieur du SL DiCo. Ceci est une autre illustration du caractère cannibale des SL.

possédait pas sa propre fiche dans le DiCo d'origine et a donc été « réifiée » lors de la génération du SL DiCo, parmi 914 autres locutions de ce type.

La « nomenclature » de notre SL est donc beaucoup plus riche que celle du DiCo dont le SL est dérivé. Cela est encore plus vrai si on inclut dans cette nomenclature les 6,464 entités lexicales du type mots-formes. Tel qu'expliqué à la fin de la section 2, il est possible de regénérer à partir du SL des descriptions lexicales pour toute entité lexicale qui est soit une unité lexicale soit un mot-forme ciblé par une application de fonction lexicale, en remplissant les descriptions des mots-formes avec des liens de fonctions lexicales inverses. Afin de tester cette possibilité, nous avons regénéré un DiCo complet en format HTML à partir du SL, avec un total de 8,154 (1,690 + 6,464) entrées lexicales, stockées individuellement sous forme de pages HTML. Les pages décrivant les mots-clés du DiCo source contiennent la spécification hypertexte des liens de fonctions lexicales d'origine, avec en plus tous les liens lexicaux inverses qui ont été trouvés dans le SL ; les pages des mots-formes ne contiennent que des liens inverses inférés. Par exemple, la page HTML de METTRE (qui n'est pas un mot-clé dans le DiCo source) contient 71 liens inverses, tels que[8] :

```
CausOper1( À L'ARRIÈRE-PLAN# ) ->
Labor12( ACCUSATION#I.2 ) ->
Caus1[1]Labreal1( ANCRE# ) ->
Labor21( ANGOISSE# ) ->
Labreal12( ARMOIRE# ) ->
```

Bien entendu, la plupart des articles qui n'étaient pas dans le DiCo source sont relativement pauvres et demanderaient un travail lexicographique intensif pour être transformés en descriptions DiCo complètes et valides. Elles représentent cependant un point de départ utile pour les lexicographes ; de surcroît, plus le DiCo s'enrichira, plus le SL deviendra productif en terme de génération automatique d'ébauches d'articles.

Nous allons, pour terminer, proposer quelques réflexions à propos du potentiel d'utilisation des modèles de type SL pour la construction de ressources multilingues.

4 Systèmes lexicaux et données multilingues

L'approche de l'implémentation de ressources lexicales multilingues que permettent les SL est compatible avec les stratégies utilisées dans des ressources multilingues déjà existantes comme Papillon (Sérasset & Mangeot-Lerebours, 2001) : il s'agit de modéliser les ressources multilingues comme des mises en relation de ressources individuelles monolingues.

Une ressource lexicale multilingue fondée sur une architecture de SL devrait être composée de plusieurs **SL complètement autonomes**, c'est-à-dire de SL qui n'ont pas été conçus ou adaptés pour la connexion multilingue. Ils fonctionnent comme des modules indépendants, qui peuvent être connectés tout en préservant leur intégrité. Les connexions entre SL monolingues sont implémentées sous forme de liens interlangues spécialisés entre entités lexicales équivalentes (du point de vue de la traduction). À une exception près : les fonctions lexicales standard (`A1`, `Magn`, `Bon`, `Oper1`, etc.). Parce que ce sont des entités lexicales universelles, ces fonctions doivent être stoc-

[8] Le soulignement indique ici les liens hypertextuels.

kées dans un module interlangue spécial ; en tant qu'universaux, elles jouent un rôle crucial dans la connectivité interlangue (Fontenelle, 1997). Cependant, il s'agit ici de fonctions lexicales « pures ». Les **applications** de fonctions lexicales, comme `Oper12`(RANCUNE) mentionnée plus haut, ne sont en aucun cas universelles et doivent être connectées à leurs contreparties dans les autres langues. Examinons brièvement cet aspect de la question.

Il convient de distinguer au moins deux cas de connexions lexicales interlangues dans les SL : les connexions lexicales directes et les connexions établies par le biais des applications de fonctions lexicales.

Les connexions directes, telles que fr. RANCUNE ⇔ ang. RESENTMENT, devraient être implémentées — manuellement ou en utilisant des ressources bilingues existantes — comme de simples liens interlangues (ou inter-SL) entre deux entités lexicales. Les choses ne sont pas toujours aussi simples, cependant, du fait de l'existence de connexions interlangues partielles ou multiples. Par exemple, quel lien interlangue devrait partir de ang. SIBLING si nous voulons qu'il pointe vers un équivalent français ? Comme il n'existe pas d'équivalent français lexicalisé, on pourrait être tenté d'inclure dans le SL français une entité comme *frère ou sœur*. Nous avons deux objections très fortes à ce type de solution. Toute d'abord, cette entité complexe ne serait pas une traduction acceptable de *sibling* dans la plupart des contextes : on ne peut pas traduire *He killed all his siblings* par *Il a tué tous ses frères ou sœurs* — la conjonction *et* est requise dans ce contexte ainsi que dans bien d'autres. Ensuite, et cela est encore plus problématique, cette solution nous force à intégrer dans le SL français des entités à seule fin de traduction, ce qui serait une transgression de l'intégrité monolingue originelle de la base[9]. Nous devons admettre que nous ne disposons pas de solution toute faite pour traiter ce problème, tout particulièrement si nous insistons sur le fait qu'il faut l'idée d'introduire des traductions périphrastiques en tant qu'entités lexicales *ad hoc* dans les SL. Il doit considérer la possibilité qu'un regroupement de SL ne puisse pas être entièrement connecté à fin de traduction sans l'introduction de SL « tampons » qui assure une connectivité interlangues complète. Par exemple, le SL tampon pour les SL français et anglais pourrait contenir des entités lexicales phrastiques comme *frères et sœurs*, *être de mêmes parents* et *être frère*(*s*) *et sœur*(*s*). Cette stratégie pourrait même être très productive et nous amener à réaliser que ce qui apparaît tout d'abord comme une solution *ad hoc* est finalement justifié d'un point de vue linguistique. Le fait de considérer le cas de la traduction française de *sibling*, par exemple, nous a forcé à réaliser qu'alors que *frère*(*s*) *et sœur*(*s*) semble très naturel en français, *sœur*(*s*) *et frère*(*s*) a un côté un peu étrange or, tout du moins, semble avoir été construit ainsi par le locuteur de façon tout à fait intentionnelle. Cela est un argument très fort pour considérer qu'existe véritablement en français une **entité** lexicale *frère*(*s*) *et sœur*(*s*)[10], indépendamment du problème de traduction que nous pose *sibling*. Cette entité phrastique devrait probablement être présente dans tout SL complet du français.

Le cas de la connexion par le biais d'applications de fonctions lexicales est encore plus délicat. Une approche simpliste serait de considérer qu'il est suffisant de connecter de façon interlinguistique les applications de fonctions lexicales afin d'obtenir toutes les connexions lexicales pour les

---

[9] Il est utile de remarquer que les bons dictionnaires anglais-français, comme le *Collins-Robert*, offriront plusieurs possibilités de traductions dans ce type de situation. De plus, leurs traductions ne s'appliqueront pas à *sibling* comme tel, mais plutôt à *siblings* ou à des expressions comme *someone's siblings*, *to be siblings*, etc.

[10] Nous n'avons pas dit, bien entendu, qu'il s'agirait d'une **unité** lexicale.

éléments de la valeur retournée par ces applications. Pour les fonctions lexicales standard, cela pourrait être fait automatiquement en utilisant la stratégie ci-dessous, pour deux langues A et B.

> Si l'entité lexicale L_A est connectée à L_B au moyen d'un lien `traduction`,
> toutes les entités lexicales liées à l'application de fonction lexicale **`f`**(L_A) par un lien `value` devraient être connectées par un lien `value_translation`, avec une valeur de confiance de `0.5`, à toutes les entités lexicales liées à **`f`**(L_B) par un lien `value`.

La distinction entre les liens `translation` et `value_translation` permet les connexions interlangues contextuelles : une entité lexicale L'_B pourrait être une bonne traduction de L'_A uniquement dans le cas où elle apparaît en tant que collocatif dans une collocation donnée. Mais cela n'est pas suffisant. Il est aussi nécessaire d'exercer un filtrage des connexions `value_translation` qui sont systématiquement générées en utilisant la stratégie ci-dessus. Par exemple, chacune des valeurs particulières données dans l'exemple (1) de la section 2 devrait être associée à son équivalent **sémantiquement le plus proche** parmi les valeurs de **`Oper12`**(RESENTMENT) : HAVE, FEEL, HARBOR, NOURISH, etc. Pour l'instant, nous ne voyons pas comment cela pourrait être effectué automatiquement, à moins que l'on puisse faire usage des bases de données multilingues de collocations déjà existantes. Pour l'anglais et le français, par exemple, nous prévoyons de faire des expérimentations avec la base de données des paires de collocations anglais-français de T. Fontenelle (Fontenelle, 1997).

5 Conclusions

Nous avons effectué la production automatique d'un SL de taille significative, si l'on considère le nombre absolu d'entités et de liens qu'il contient ainsi que la richesse de la connaissance lexicale qu'il encode. Nous prévoyons de compléter l'absorption de toute l'information contenue dans la base DiCo (y compris l'information qui peut être inférée). Nous aimerions aussi pouvoir intégrer des bases de données complémentaires du français dans le SL et débuter l'implémentation de connexions multilingues. Un autre développement serait la construction d'un éditeur permettant d'accéder aux données du SL et de les modifier. Cet outil pourrait aussi être utilisé pour développer des SL du type DiCo pour d'autres langues que le français.

Remerciements

Les travaux lexicographiques menés à l'OLST sont en partie rendus possibles par une subvention de recherche équipe du FQRSC.

Bibliographie

Breen, J. W. 2004. JMdict: a Japanese-Multilingual Dictionary. In *Proceedings of COLING Multilingual Linguistic Resources Workshop*, Genève.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database,* Cambridge, MA: MIT Press.

Fontenelle, T. 1997. *Turning a bilingual dictionary into a lexical-semantic database*, Niemeyer, Tübingen.

Kahane, S. & A. Polguère. 2001. Formal foundation of lexical functions. In *Proceedings of ACL/ EACL 2001 Workshop on Collocation*, 8-15, Toulouse.

Igor Mel'čuk. 1996. Lexical Functions: A Tool for the Description of Lexical Relations in the Lexicon. In L. Wanner (ed.): *Lexical Functions in Lexicography and Natural Language Processing*, 37-102, Amsterdam/Philadelphie, Benjamins.

Mel'čuk, I., A. Clas & A. Polguère. 1995. *Introduction à la lexicologie explicative et combinatoire*, Louvain-la-Neuve: Duculot.

Polguère A. 2000a. Towards a theoretically-motivated general public dictionary of semantic derivations and collocations for French. In *Proceedings of EURALEX'2000*, 517-527, Stuttgart.

Polguère A. 2000b. Une base de données lexicale du français et ses applications possibles en didactique, *Revue de Linguistique et de Didactique des Langues* (*LIDIL*), 21:75-97.

Polguère, A. 2006. Structural properties of Lexical Systems: Monolingual and Multilingual Perspectives. In *Proceedings of the Workshop on Multilingual Language Resources and Interoperability (COLING/ACL 2006)*, 50-59, Sydney.

Steinlin, J., Kahane, S. & A. Polguère. 2005. Compiling a "classical" explanatory combinatorial lexicographic description into a relational database. In *Proceedings of the Second International Conference on the Meaning Text Theory*, 477-485, Moscou.

Sérasset, G. & M. Mangeot-Lerebours. 2001. Papillon lexical database project: Monolingual dictionaries and interlingual links. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, 119–125, Tokyo.

Trippel, T. 2006. *The lexicon graph model: a generic model for multimodal lexicon development*, Thèse de doctorat, Université Bielefeld.

De la lexicographie formelle pour la terminologie : projets terminographiques de l'Observatoire de linguistique Sens-Texte (OLST)

Marie-Claude L'Homme

Observatoire de linguistique Sens-Texte (OLST)
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
H3C 3J7
mc.lhomme@umontreal.ca

Résumé

Le présent article décrit une méthodologie élaborée par l'équipe de terminologie de l'Observatoire Sens-Texte pour construire des ressources dans lesquelles sont décrits des termes spécialisés. Ces ressources ont ceci de particulier qu'elles veulent donner un portrait complet des propriétés lexico-sémantiques des unités terminologiques. J'illustrerai ce travail à partir d'exemples extraits d'une base de données portant sur des termes appartenant au domaine de l'informatique et de l'Internet. Quelques autres projets en cours sont également évoqués.

Mots-clés

Terminologie, lexicographie, terme, base de données lexicales, liens sémantiques, fonctions lexicales, structure actancielle

1 Pourquoi des bases de données *lexicales* en terminologie ?

Généralement, les terminologues abordent les unités linguistiques en tenant pour acquis que les unités « terminologiques » sont celles qui représentent des concepts, à savoir des entités définies en fonction des liens qu'elles tissent avec d'autres entités dans l'organisation des connaissances d'un domaine de spécialité. Ils reproduisent ainsi la démarche des spécialistes et tentent d'établir une correspondance entre le système conceptuel du domaine et les unités linguistiques servant à étiqueter des nœuds dans ce système (ces unités seront alors des termes). Cette optique les amène presque naturellement a opter, afin de modéliser les structures terminologiques, pour des représentations de nature ontologique.

Cette démarche, que nous qualifions de *conceptuelle*, est nécessaire, mais non suffisante pour appréhender pleinement les unités appelées *termes,* notamment parce qu'elle impose qu'on dépouille les termes d'une grande partie de leurs propriétés linguistiques. Nous rappelons ci-dessous quelques conséquences linguistiques de la démarche, conséquences qui serviront de levier pour défendre un traitement lexico-sémantique des termes (certains de ces arguments sont également présentés dans L'Homme, 2005, 2007b) [1] :

- L'exhaustivité de la description faite à l'intérieur d'un domaine dépend d'une délimitation préalable de la structure conceptuelle et non d'une appréhension de la structure lexicale. Ainsi, dans une démarche conceptuelle, on jugera utile de retenir *fichier de configuration du gestionnaire de communications* (sans doute parce qu'il s'agit d'une sorte de « fichier de configuration » – lui-même étant une sorte de « fichier » – qui se distingue d'autres sortes de fichiers). En revanche, il apparaîtra moins essentiel de retenir tous les termes reliés sémantiquement à un premier (par exemple, même si on juge utile de retenir *compilateur*, il est moins certain qu'on retiendra tous les termes reliés sémantiquement à celui-ci : *compiler, compilation, compilable, traduire*, etc.).

- La question de la polysémie n'est abordée que dans la mesure où elle explique qu'une forme étiquette deux concepts différents (dans le même domaine ou dans deux domaines distincts). Ainsi, il n'apparaît pas indispensable de faire des distinctions fines comme celle existant entre *configuration*$_1$ « activité consistant à configurer » et *configuration*$_2$ « résultat de cette activité ».

- Les termes décrits sont essentiellement de nature nominale (ex. en informatique, on jugera utile de décrire *fichier, ordinateur portable, pixel*; en revanche, des verbes comme *copier* ou *supprimer* ou, encore, des adjectifs comme *robuste* ou *sensible* risquent d'échapper à l'analyse).

- La plupart des termes sont multilexémiques et cela, même s'ils ont un sens compositionnel (voir l'exemple de *fichier de configuration du gestionnaire de communications* cité ci-dessus).

Pour les raisons évoquées ci-dessus, il apparaît indispensable de proposer des descriptions des propriétés linguistiques des termes, notamment de leurs propriétés lexico-sémantiques. C'est à ce vaste chantier que s'est attaqué une équipe de terminologie de l'Observatoire de linguistique Sens-Texte (OLST). Nous ne prétendons pas ainsi épuiser tous les dimensions que peut revêtir le terme dans tous les domaines d'application où on fait appel à cette notion. Toutefois, nous pensons que nos descriptions permettent de saisir la dimension lexicale du terme et d'en offrir un portrait fidèle.

[1] On retrouvera, par ailleurs, des arguments de nature différente dans Bourigault et Slodzian (1999), Cabré (2003), Gaudin (2003).

Ce travail repose sur des modèles de sémantique lexicale (notamment, Cruse 1986 ; et la Lexicologie explicative et combinatoire, désormais *LEC*, Mel'čuk et al., 1984-1999, 1995) qui offrent un cadre conceptuel ainsi que des principes méthodologiques permettant de mettre au jour les propriétés lexico-sémantiques des unités lexicales. Nous énumérons, ci-dessous, quelques-uns des éléments que nous empruntons à ces modèles et que nous appliquons dans nos projets :

- Une définition de la notion d' « unité lexicale » reposant sur les principes d'autonomie de fonctionnement, de sens lexical et de non-compositionnalité ;
- Une série de critères permettant de baliser les intuitions que peut avoir un analyste quant à la polysémie d'une forme lexicale ;
- Un modèle descriptif reposant sur une définition de la structure actancielle des unités lexicales ;
- En ce qui concerne la LEC, un système complet d'appréhension et de description des liens existant entre unités lexicales, système appelé *fonctions lexicales*, désormais *FL*, qui permet de représenter les relations paradigmatiques et syntagmatiques.

Dans la suite de l'article, je me focaliserai sur la méthodologie que nous avons mise au point ainsi que sur l'encodage des descriptions réalisées par l'équipe. J'illustre la démarche au moyen d'exemples tirés d'une base de données lexicales décrivant des termes français appartenant aux domaines de l'informatique et de l'Internet. J'évoquerai, dans la dernière section, quelques autres projets menés par l'équipe.

2 Méthodologie et encodage

Notre méthodologie se décline en cinq étapes principales que je décris brièvement ci-dessous. L'encodage est réalisé par les terminologues dans le système de gestion de bases de données *Access*[2]. Une partie des tables est exportée dans *MySQL* afin de permettre un accès en ligne.

2.1 Recours aux corpus

Toutes les composantes de la méthodologie s'appuient sur un corpus spécialisé assemblé en ayant recours à des critères de sélection des textes (représentativité des spécialisations d'un domaine, genre textuels, langue de rédaction, niveau de spécialisation, etc.). Les corpus utilisés ont une taille variant entre 300 000 et 1 000 000 de mots. Le corpus permet d'effectuer une première sélection des termes, d'obtenir des concordances et de repérer automatiquement des paires de termes reliés sémantiquement.

---

[2] Un projet est en cours (en collaboration avec Guy Lapalme, du Département d'informatique et de recherche opérationnelle) afin de convertir les tables Access dans un formalisme XML et de permettre un encodage directement dans ce format.

Bien que le corpus soit indispensable, notamment parce qu'il permet au terminologue qui n'est pas spécialiste d'acquérir des connaissances sur le domaine, il n'est pas suffisant. Des descriptions exhaustives nécessitent forcément un recours à des ressources plus importantes et de nombreuses distinctions fines exigent le recours à des spécialistes.

2.2 Établissement de la nomenclature

L'établissement de la nomenclature s'effectue en deux temps : d'abord, un extracteur de termes génère une liste d'unités spécifiques ; ensuite, un terminologue parcourt cette liste afin de retenir les candidats valables.

2.2.1 Extraction automatique de termes

L'extracteur terminologique utilisé, appelé *TermoStat* (Drouin 2003) génère une liste de candidats à partir d'un calcul permettant de saisir la spécificité des unités en opposant leurs occurrences dans un corpus spécialisé à celles observées dans un autre corpus. Le tableau 1 présente la première partie d'une liste de candidats obtenue en comparant un corpus d'informatique (d'environ 600 000 mots) à un corpus composé de textes journalistiques (*Le Monde*)[3].

Candidat	Étiquette (Winbrill)	Fréquence	Valeur-test
fichier	SBC	3956	315,535
commande	SBC	1902	177,526
internet	SBC	1102	170,618
serveur	SBC	1166	168,286
utiliser	VB	1993	163,323
utilisateur	SBC	1117	162,823
logiciel	SBC	1166	161,681
option	SBC	1486	158,922
ordinateur	SBC	1283	153,731
système	SBC	2699	143,549
configuration	SBC	845	140,704
répertoire	SBC	1003	139,748
touche	SBC	855	124,923
disquette	SBC	609	124,514
windows	SBP	613	124,412
votre	DT	1365	121,361
disque	SBC	1093	120,167
réseau	SBC	1511	119,167
imprimante	SBC	537	117,294
mémoire	SBC	1240	112,42
clavier	SBC	579	111,162

Tableau 1: Première partie d'une liste de candidats-termes

[3] Cette expérimentation et son évaluation ont été réalisées par Lemay (2003). L'expérimentation a été effectuée sur des unités monolexémiques et toutes parties du discours confondues.

2.2.2 Critères de sélection des termes

Les listes de candidats sont ensuite étudiées par les terminologiques qui sélectionnent les noms, verbes, adjectifs ou adverbes valables, c'est-à-dire qui feront l'objet d'un ou de quelques articles dans la base de données. Ils s'aident dans cette tâche des quatre critères suivants :

- Dénotation d'une entité liée au domaine : ex. *fichier, logiciel.*
- Présence d'actants de nature spécialisée : ex. *créer* qui s'emploie systématiquement avec des termes d'informatique, à savoir *fichier, répertoire, base de données* et ainsi de suite.
- Lien morphologique accompagné d'un lien sémantique : ex. *modérer* et *modération* seront retenus si *modérateur* a été sélectionné.
- Autre lien paradigmatique : ex. *supprimer* et *effacer* seront retenus puisque leur antonyme, à savoir *créer*, est sélectionné.

Les termes retenus sont placés dans une table de la base de données accompagnés d'un certain nombre de contextes apparaissant dans le corpus ou recueillis dans des sites d'informatique publiés dans le Web. La figure 1 illustre cet encodage au moyen du terme *fichier*.

Table 1 : Termes

Terme	**Inf. gramm.**	**Statut**
fichier 1	n. m.	3

Table 2 : Contextes

Terme	**Contexte**	**Source**
fichier 1	Ce fichier peut être stocké pour garder une trace de ces informations. Un fichier texte est un fichier composé de caractères stockés sous la forme d'octets.	SYSINT
fichier 1	La fonction de compression de fichiers consiste à réduire leur taille en appliquant un algorithme qui enregistre l'information sous une forme plus compacte.	GIRI
fichier 1	Il est plus rapide de se servir de la souris pour copier ou déplacer des fichiers ou des répertoires.	PCPREM
fichier 1	A chaque fois que vous supprimez un fichier ou un répertoire sur votre disque dur, celui-ci est stocké provisoirement dans la Corbeille de Windows afin d'être éventuellement récupéré si vous l'avez effacé accidentellement.	ETERNIR

Figure 1: Encodage résultant de la sélection du terme *fichier*

Notre méthode, s'appuyant sur une liste de termes générés automatiquement, ne permet pas de retrouver immédiatement l'ensemble des termes des domaines étudiés. Certains peuvent être absents des corpus ou, encore, de la liste de spécificités. Des termes sont ajoutés plus tard lorsque des liens sémantiques importants entre un premier terme sélectionné et un autre sont découverts. Par exemple, les termes *blogue*, *bloguer*, *bloguer*, *blogage*, *blogosphère*$_1$ et *blogosphère*$_2$ ont été ajoutés plus tard à la nomenclature. *Blogue* a été ajouté puisqu'il s'agit d'une sorte de site (application du critère de lien paradigmatique) ; les autres termes ont été retenus en appliquant le critère de lien morphologique. Par ailleurs, comme la liste de spécificités ne referme que des unités monolexémiques, certaines locutions doivent être reconstruites en cours d'analyse (ex. *volée -> à la volée* ; *dur -> disque dur*).

2.3 Distinctions sémantiques

Certains des formes stockées dans la base de données doivent faire l'objet d'une distinction sémantique. Chaque article est consacré à la description d'un sens différent. Ici, quatre nouveaux critères servent à baliser les intuitions des terminologues :

- Cooccurrence compatible ou différentielle (Mel'čuk et al. 1995) : ex. *formatage*$_{1a}$ *d'un disque ; formatage*$_{2a}$ *d'un texte ; *formatage d'un texte ou d'un disque dur.*

- Substitution par un synonyme : *sélectionner*$_1$ *une option, choisir une option ; sélectionner*$_2$ *du texte, *choisir du texte*.

- Dérivation morphologique différentielle : *adresse*$_1$ (d'une mémoire) (lié à *adresser, adressage, adressable*); *adresse*$_2$ (URL, Web) (pas de dérivés morphologiques répertoriés).

- Autres liens paradigmatiques différentiels : *page*$_1$ (partie d'un document) ; *page*$_2$ (partie d'un site) ; *page*$_3$ (division d'une mémoire).

Les distinctions sémantiques peuvent être faites au moment de la sélection ou plus tard lors de la rédaction des articles. Dans certains cas, la numérotation des acceptions et des informations associées à ces acceptions doit être revue dans la base de données.

2.4 Structure actancielle

La première étape de la rédaction des articles commence par une définition provisoire de la structure actancielle des termes. Ici, nous privilégions une approche visant à travailler simultanément sur des acceptions voisines plutôt que de travailler sur les multiples acceptions que peut avoir une forme polysémique. Par exemple, *compiler, compilateur, compilation, compilable, recompiler*, etc., seront décrits en parallèle. En revanche, les deux acceptions de *formater* ont été décrites à des moments éloignés dans le temps.

Notre encodage de la structure actancielle tient compte de trois paramètres (un exemple est donné à la figure 2) :

- Les rôles actanciels (ou thématiques) : AGENT, PATIENT, etc. ce qui nous permet d'utiliser la même notation pour des termes de sens proche même si les actants occupent une position différente.

- Les termes prototypiques réalisant un rôle actantiel.

- Les autres termes réalisant un rôle actanciel faisant l'objet d'un article dans la base de données.

Table 1 : Termes

Terme	**Inf. gramm.**	**Statut**	**Structure actancielle**
compilable 1	adj.	1	PATIENT{programme 1} est ~
compiler 1	v. tr.	1	INSTRUMENT{compilateur 1} ~ PATIENT{programme 1}
compiler 2	v. tr.	1	AGENT {utilisateur 1} ~ PATIENT{programme 1} avec INSTRUMENT{compilateur 1}
compilateur 1	n. m.	1	~ utilisé par AGENT{programmeur 1} pour convertir PATIENT{programme 1}

Table 3 : Termes reliés

Terme	**Explication**	**Terme relié**
compiler 1	Nom du patient	programme 1
compiler 1	Nom de l'agent	programmeur 1
compiler 1	Nom de l'agent	informaticien 1
compiler 1	Nom du patient	script 1
compiler 1	Nom du patient	code 2

Figure 2 : Encodage de la structure actancielle pour le terme *compiler*

2.5 Listes de liens lexicaux

La dernière étape, la plus longue, consiste à découvrir, puis, à encoder tous les liens sémantiques partagés par un terme faisant l'objet d'un article avec d'autres termes et unités lexicales retrouvés dans les corpus. Ces liens sont paradigmatiques (synonymie, antonymie, hyperonymie, méronymie, etc.) ou syntagmatiques (collocations). La notion de « lien lexical » et sa représentation s'appuie sur le système des fonctions lexicales mis au point en LEC.

La découverte des liens lexicaux est réalisée essentiellement à la main, mais des expérimentations ont permis de montrer qu'une partie de ces liens peut être acquise automatiquement. Une première méthode (Claveau & L'Homme, 2005) permet d'acquérir des termes apparentés morphologiquement (ex. *blogue, bloguer, blogage, blogueur*) et de faire des hypothèses quant au lien sémantique qu'ils partagent. Ces hypothèses s'appuient sur des exemples déjà encodés (par exemple, si le lien entre *programmer* et *programmeur* a été encodé comme étant un lien entre un prédicat et un agent, on en déduira que le lien entre *bloguer* et *blogueur* est de même nature). Une seconde méthode permet d'extraire des collocations dont les composantes (mot clé et collocatif) partagent une relation syntaxique et sémantique spécifique. Une série d'expérimentations (Claveau & L'Homme, 2006) a été réalisée pour extraire des collocations composées d'un nom et d'un verbe dans lesquelles le verbe véhicule un sens de « réalisation » (ex. le lien observé dans *exécuter un programme* ou *utiliser un outil*, par opposition au lien observé dans *procéder à une installation* ou *avoir un mot de passe*).

L'encodage des liens lexicaux dans la base de données se fait de deux manières. D'abord, une notation en FL permet de systématiser l'encodage dans la base de données pour des termes appartenant à des classes sémantiques très variées. Ensuite, une explication (s'inspirant des formules de vulgarisation proposées par Polguère, 2003) est donnée afin de permettre un décodage plus rapide. Une partie des liens relevés pour le terme *compilateur* est reproduite à la figure 3.

Table 1 : Termes

Terme	**Inf. gramm.**	**Statut**	**Structure actancielle**
compilateur 1	n. m.	1	~ utilisé par AGENT{programmeur 1} pour convertir PATIENT{programme 1}

Table 3 : Termes reliés

Terme	**FL**	**Explication**	**Valeur**	**Terme relié**	**No d'ordre**
compilateur 1	S2 - patient	Nom du patient	programme 1	programme 1	.1
compilateur 1	Sres	Résultat	exécutable 1	exécutable 1	.9
compilateur 1	Syn	Synonyme	programme de compilation	----	0
compilateur 1	Syn	Synonyme	programme compilateur	----	0
compilateur 1	Cf	Sens voisin	interpréteur 1	interpréteur 1	0.0.3
compilateur 1	Bon/Ver	Qui est très efficace	~ performant 1	performant 1	0.3.6
compilateur 1	CausFunc0	Qqn. crée un « mot clé »	développer 1 un ~	développer 1	1
compilateur 1	CausFunc0	Qqn. crée un « mot clé »	écrire 2 un ~	écrire 2	1
compilateur 1	Labreal12	L'agent utilise un « mot clé » pour intervenir sur le patient	compiler 2 … avec un ~	compiler 2	2.3
compilateur 1	Fact2	Le « mot clé » intervient sur le patient	le ~ compile 1 …	compiler 1	2.3
compilateur 1	Fact2	Le « mot clé » intervient sur le patient	le ~ traduit 1 …	traduire 1	2.3

Terme : Vedette d'un article
FL : Encodage du lien lexical au moyen des fonctions lexicales
Explication : Explication du lien lexical au moyen d'une paraphrase
Valeur : Valeur de la fonction lexicale
Terme relié : Terme important dans la valeur permettant d'établir un lien avec l'article décrivant ce terme
Numéro d'ordre : Numéro permettant d'ordonner des liens lexicaux à l'affichage

Figure 3 : Encodage de liens lexicaux de *compilateur*

3 Autres chantiers

Jusqu'à présent, beaucoup d'efforts ont été consacrés au développement de la base de données lexicales portant sur les termes d'informatique et de l'Internet. Ce travail a permis de mettre à l'épreuve et de perfectionner une méthodologie ainsi qu'un modèle d'encodage qui peuvent maintenant être étendus à d'autres domaines de spécialité et à d'autres langues que le français.

Dans ce qui suit, nous donnons un aperçu des différents projets reliés à l'élaboration de ressources terminologiques sur lesquels nous travaillons actuellement[4].

- Version multilingue de la base de données portant sur l'informatique et l'Internet : nous comptons préparer des descriptions portant sur d'autres langues que le français (travail déjà commencé en anglais, en espagnol et en coréen : pour le travail sur le coréen, voir Bae et L'Homme, 2006). Ce travail permettra de développer des stratégies afin de relier les équivalents lexicaux et non lexicaux dans la base de données.

[4] Signalons que d'autres bases de données terminologiques sont élaborées à l'OLST sous la direction de Jeanne Dancette : une première base de données portant sur le domaine de la distribution ; une seconde, sur le domaine de la mondialisation du travail.

- Modèle d'encodage reposant sur XML : ce travail a déjà été commencé en collaboration avec Guy Lapalme, du Département d'informatique et de recherche opérationnelle de l'Université de Montréal. Cet encodage permettra, notamment, de vérifier la cohérence des descriptions et d'effectuer des annotations en corpus et de mieux arrimer les données du corpus aux descriptions réalisées par les terminologues.

- Réalisations syntaxiques des termes d'informatique et d'Internet : les articles, dans leur forme actuelle, ne comportent pas de rubrique sur le comportement des termes. Nous développerons, toujours en collaboration avec Guy Lapalme, une méthodologie consistant à annoter des exemples extraits de corpus et à générer des descriptions à partir de ces annotations (cette méthodologie s'inspire des travaux effectués dans le cadre de FrameNet, 2007).

- Technique d'acquisition des liens morphologiques pour de nouveaux domaines et de nouvelles langues : ce travail sera réalisé en collaboration avec Patrick Drouin (de l'Université de Montréal) et Vincent Claveau (de l'IRISA, Université de Rennes). La méthode devra être modulée de manière à permettre une interaction avec le terminologue. Contrairement aux expérimentations réalisées sur le français – dont nous avons parlé plus haut – nous ne pouvons pas nous appuyer sur des liens préalablement décrits.

- Base de données lexicales portant sur le domaine de l'environnement : un projet vient de démarrer afin de développer une nouvelle ressource décrivant les termes du domaine de l'environnement au moyen de la méthodologie mise au point pour construire la base de données de l'informatique et de l'Internet. Toutefois, contrairement à cette première ressource qui a été construite d'abord en français, les descriptions anglaises et françaises seront réalisées en parallèle.

- Base de données lexicales servant à décrire les adjectifs spécialisés (Carrière et al., 2007) : cette ressource diffère sensiblement de celles qui ont été évoquées plus haut. Elle vise à proposer un modèle d'encodage dans lequel les adjectifs sont décrits en tenant compte de la variation sémantique qu'ils subissent lorsqu'ils modifient des noms appartenant à une classe sémantique spécifique. Pour l'instant, la base de données renferme des adjectifs appartenant aux domaines de la médecine et de l'informatique. La mise en ligne des premiers articles est prévue pour la fin de 2007.

Remerciements

Le travail décrit dans cet article a bénéficié du soutien financier du Conseil de recherche en sciences humaines (CRSH) et du Fonds québécois de recherche sur la société et la culture (FQRSC).

Bibliographie

Bae, H.S. & M.C. L'Homme. 2007, à paraître. Converting A Monolingual Lexical Database into a Multilingual Specialized Dictionary. In *Proceedings of the Conference on Multilingualism and Applied Comparative Linguistics*.

Bourigault, D. & M. Slodzian. 1999. Pour une terminologie textuelle. *Terminologie nouvelles* 19:29-32.

Cabré, M.T. 2003. Theories of Terminology: Their description, Prescription and Explanation. *Terminology* 9(2):163-199.

Carrière, I., M.C. L'Homme & P. Drouin. 2007. Modèle de description des adjectifs terminologiques. In *Terminologie, approches transdisciplinaires*, Gatineau (Québec), 2-4 mai 2007.

Claveau, V. & M.C. L'Homme. 2005. Apprentissage par analogie pour la structuration de terminologie - Utilisation comparée de ressources endogènes et exogènes. In *Terminologie et intelligence artificielle, TIA 2005. Actes*, Université de Rouen, Rouen (France), 4-5 avril 2005.

Claveau, V. & M.C. L'Homme. 2006. Discovering and Organizing Noun-Verb Collocations in Specialized Corpora using Inductive Logic Programming" *International Journal of Corpus Linguistics* 11(2):209-243.

Cruse, D.A. 1986. *Lexical Semantics,* Cambridge: Cambridge University Press.

Drouin, P. 2003. Term Extraction Using Non-Technical Corpora as a Point of Leverage. *Terminology* 9(1):99–117.

FrameNet (http://framenet.icsi.berkeley.edu/) Consulté le 3 septembre 2007.

Gaudin, F. 2003. *Socioterminologie. Une approche sociolinguistique de la terminologie,* Bruxelles : De Boeck/Duculot.

L'Homme, M.C. 2005. Sur la notion de 'terme. *Meta* 50(4):1112-1132.

L'Homme, M.C. (dir.). 2007a. *DiCoInfo, le dictionnaire fondamental de l'informatique et de l'Internet* http://olst.ling.umontreal.ca/dicoinfo/.

L'Homme, M.C. 2007b. Using Explanatory and Combinatorial Lexicology to Describe Terms. In Wanner. L. (ed.). *Selected Lexical and Grammatical Topics in the Meaning-Text Theory. In Honour of Igor Mel'čuk*. Amsterdam/Philadelphia: John Benjamins.

Lemay, C. 2003. *Identification automatique du vocabulaire caractéristique de l'informatique fondée sur la comparaison de corpus*. Mémoire de maîtrise. Montréal : Université de Montréal.

Mel'čuk, I. et al. 1984-1999. *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques 1-IV*, Montréal: Presses de l'Université de Montréal.

Mel'čuk, I., A. Clas & A. Polguère. 1995. *Introduction à la lexicologie explicative et combinatoire*, Paris: Duculot.

Polguère, A. 2003. Collocations et fonctions lexicales : pour un modèle d'apprentissage. In F. Grossmann et A. Tutin (ed.). *Les collocations : analyse et traitement*. 23-44. Amsterdam : Éditions de Werelt.

Structuration de lexiques pour Antidote

Bertrand Pelletier, Éric Brunelle, Jean Saint-Germain
Jean Fontaine, Jasmin Lapalme, Simon Charest

Druide informatique inc.
1435 St-Alexandre, bureau 1040
Montréal, Québec
H3A 2G4
developpement@druide.com

Résumé

Cet article décrit la structure et le traitement des principaux lexiques utilisés par Druide informatique pour l'élaboration de ses produits et services, dont Antidote. Les lexiques sont en particulier mis en relation avec leur utilisation dans les dictionnaires d'Antidote. Une comparaison des avantages et inconvénients des structures est présentée.

Mots-clés

Antidote, lexique, dictionnaire, fichier, SGBD.

1 Introduction

Les produits et services développés par Druide informatique[1] reposent sur une grande quantité d'information lexicographique – les lexiques[2]. Ces lexiques sont utilisés pour élaborer des produits variés, tels Antidote (logiciel d'aide à la rédaction), WebElixir (service de veille-qualité pour sites Web), PDS (Petit Druide des synonymes – version papier) et GDS (Grand Druide des synonymes – version papier).

[1] Entreprise québécoise spécialisée dans l'édition et la distribution de logiciels grand public.

[2] Lexique : ensemble d'unités lexicales accompagnées de données systématisées.

Le maintien de la qualité de ces produits tout au long de leur évolution et la création de nouveaux produits exige une évolution des lexiques et un maintien de leur qualité. De plus, comme il s'agit de produits commerciaux, la contrainte d'efficacité s'ajoute.

La structuration des lexiques est par conséquent critique, tout comme leurs outils de gestion.

Dans cet article, nous allons brièvement présenter les lexiques et leur structuration, en mettant particulièrement en correspondance les lexiques avec leur utilisation dans les dictionnaires[3] d'Antidote (Fig. 1). Nous allons par la suite présenter les actions qui y sont effectuées, ainsi que les avantages et inconvénients de leur structure.

Figure 1 : Antidote – Dictionnaire de définitions – fiche « ciel »

[3] Les dictionnaires : les 10 dictionnaires contenus dans le logiciel Antidote.

2 Lexiques

Les dictionnaires d'Antidote (définitions, cooccurrences, …) proviennent de lexiques dont la structure et le format varient (DicoDefLoc, DicoCooc, …). Il y a presque correspondance biunivoque entre lexiques et dictionnaires. La figure 2 illustre notamment la transformation (compilation) des lexiques en fichiers binaires intermédiaires, lesquels sont recompilés en fichiers finaux directement utilisés par les logiciels (Antidote, WebElixir ou les outils internes de génération des dictionnaires de papier).

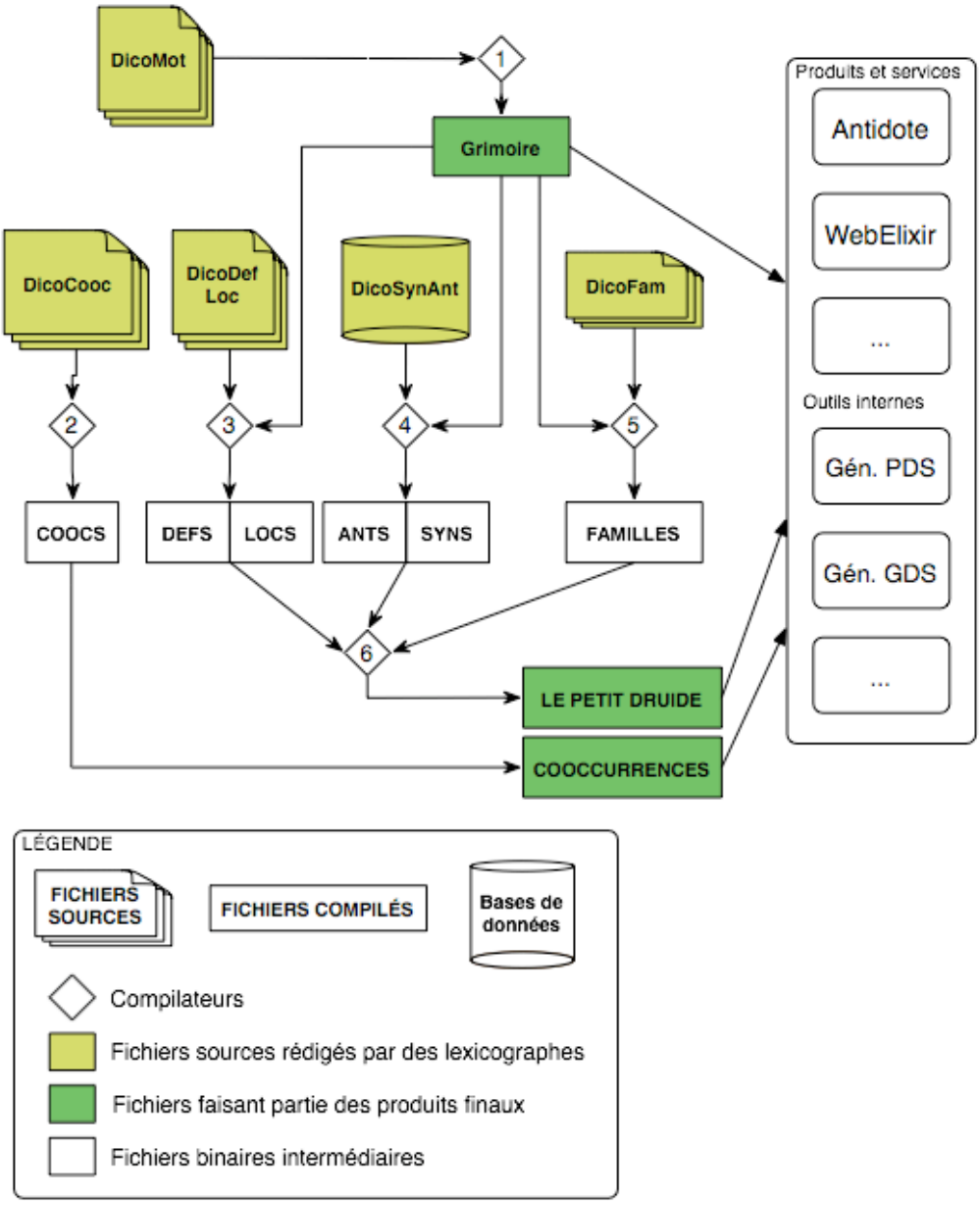


Figure 2 : Lexiques et logiciels de Druide informatique.

2.1 Le DicoMot

C'est le lexique pivot : l'information qu'il contient est requise pour relier et traiter tous les autres lexiques. Il fournit la liste des mots, et pour chacun d'eux, ses informations syntaxiques, morphologiques et graphiques, ainsi que les données grammaticales ou définitoires utilisées dans Antidote pour la correction ou pour l'affichage d'alertes. Dans Antidote, son contenu est notamment utilisé pour les dictionnaires de conjugaison et d'anagrammes, ainsi que pour le panneau de droite du dictionnaire de définitions (Fig. 1). Le DicoMot consiste en un ensemble de fichiers texte gérés par un éditeur standard.

2.2 Le DicoCooc

Il contient les cooccurrences et les citations affichées dans les dictionnaires correspondants dans Antidote. Il est également utilisé pour la recherche multimot (identification dans différents lexiques des fiches où figure simultanément une liste de mots). Le DicoCooc est constitué d'un grand nombre de fichiers texte. Contrairement aux autres lexiques texte, il est géré par les lexicographes depuis une application de type SGBD (Système de gestion de base de données) développée à l'interne.

2.3 Le DicoDefLoc

Il contient les définitions et les locutions. Dans Antidote, il sert non seulement à la production des dictionnaires de définitions et de locutions, mais aussi à la recherche multimot. Il consiste en un ensemble de fichiers texte gérés par un éditeur standard.

2.4 Le DicoSynAnt

Il contient les synonymes, les hyponymes et les hyperonymes (dictionnaire de synonymes) ainsi que les antonymes (dictionnaire d'antonymes).

À l'origine, il consistait tout comme le DicoDefLoc en un ensemble de fichiers. Cependant, pour satisfaire certains besoins lexicographiques (notamment l'édition simultanée par plusieurs lexicographes), il a été converti en base de données.

2.5 Le DicoFam

Il s'agit d'un fichier texte unique qui regroupe les mots en familles (*neige*, *neigeux*, *reneiger*…). Il est géré par un éditeur standard.

3 Traitement des données – Actions

3.1 Entrée initiale

C'est la genèse : il faut créer l'information, généralement à partir de rien. Les fichiers texte se prêtent assez bien à cette entrée massive et répétitive de données. Généralement, ce sont plusieurs lexicographes qui se partagent la tâche en simultané, mais chacun doit se limiter à un fichier, car les risques de conflits sont plus grands au début.

3.2 Recherche dans les données source

Régulièrement, lexicographes et linguistes ont besoin d'effectuer des recherches sur les données brutes. Elles peuvent être simples : « y a-t-il beaucoup de mots qui ont le trait X ? » Elles peuvent être complexes : « quelle intersection y a-t-il entre les mots qui ont le trait X et le trait Y, et ceux qui ont le trait Z ou le trait W ? » La nature et l'étendue de ces requêtes est imprévisible. Il est arrivé fréquemment qu'on ne puisse pas répondre aux interrogations complexes.

3.3 Modification

Il faut régulièrement modifier les données, pour ajouter de nouvelles fiches, de nouvelles informations sur les fiches existantes ou corriger l'information existante. Ce sont des manipulations ponctuelles, généralement réparties un peu partout dans l'ensemble des fichiers d'un lexique donné. Souvent, ces modifications doivent être appliquées à plusieurs lexiques à la fois. Nous avons par exemple une norme qui dicte que, lors de l'ajout d'un nouveau mot au lexique central (DicoMot), l'information complémentaire à ce mot doit être ajoutée immédiatement à tous les autres lexiques pertinents (DicoDefLoc, DicoSynAnt…). Ces ajouts peuvent engendrer des collisions (modifications simultanées par plusieurs lexicographes sur une même fiche ou dans un même fichier) qu'il faut gérer correctement et efficacement.

3.4 Compilation

Les lexiques ne peuvent pas être utilisés directement dans leur format source. Ils doivent être compilés (Fig. 2). La compilation extrait l'information source, en vérifie la conformité syntaxique et la cohérence selon plusieurs règles, crée les liens entre les divers dictionnaires, et génère un format binaire optimisé en temps d'accès et en espace. Les fichiers texte peuvent être compilés directement. Pour les SGBD, l'information est d'abord extraite dans un fichier texte, lequel est reformaté, puis finalement compilé.

3.5 Validation

Plusieurs types de validations sont souhaitables et, dans certains cas, requis. Voici la description des deux principales.

3.5.1 Syntaxe

La validation syntaxique est fondamentale et s'effectue en général automatiquement à l'étape de la compilation des données. La compilation est généralement effectuée une fois par semaine et, par conséquent, après un grand nombre de modifications effectuées par plusieurs lexicographes. Dans ce cas, la compilation fournit une liste des erreurs syntaxiques, que les lexicographes corrigent. Dans certains cas (rares, heureusement), plusieurs « corrections syntaxiques » d'une même erreur sont possibles, et il faut identifier le lexicographe ayant effectué la modification, afin qu'il puisse appliquer « sa » bonne correction. Il faut alors espérer qu'il se souvienne de l'information linguistique qu'il désirait exprimer. S'il s'avère que le lexicographe en question est absent au moment de la validation syntaxique, il faut effectuer une correction temporaire, pour ne pas interrompre le processus de génération, et laisser une note qui sera vérifiée par le lexicographe à son retour. Devant la lourdeur de ce processus, il serait préférable que la validation syntaxique s'effectue aussitôt qu'une information est complétée, permettant ainsi au lexicographe de corriger l'information (fraiche dans son esprit) qu'il vient d'entrer.

3.5.2 Cohérence

Il peut s'agir de vérifier la cohérence au sein d'une même fiche (par exemple, la présence de deux traits incompatibles), à travers tout un lexique (par exemple, la présence d'une fiche double), ou parmi plusieurs lexiques (par exemple, l'utilisation dans une définition du DicoDefLoc d'un mot qui ne figure pas encore au DicoMot). L'application des critères de cohérence s'effectue en général lors de la compilation. Cependant, les critères restreints à une fiche peuvent également être vérifiés immédiatement suite à la modification.

4 Traitement des données – Outils / Structure

Il n'existe pas d'outil parfait (ni donc de structure parfaite) pour effectuer les actions mentionnées à la section précédente. Chacun a ses avantages et inconvénients, et cela vaut également pour les outils que nous avons développés à l'interne. En ne considérant que les actions les plus fréquentes, soit la recherche et la modification, voici certaines comparaisons entre la structure « fichiers texte » et la structure « bases de données ».

4.1 Avantages des fichiers texte

En utilisant un système de gestion de versions de fichiers (CVS – Concurrent Versions System, Subversion…), il est possible de retracer tout l'historique d'un fichier, depuis le jour de sa création. Par exemple, il est aisé de restaurer l'état d'un fichier à une date spécifique, ou encore d'identifier qui a indiqué que le pluriel de « bateau » est « bateaus ». Les bases de données courantes ne permettent pas aisément cette granularité ou la restauration.

Plusieurs éditeurs de texte conservent la séquence des modifications effectuées pendant une session de travail, de sorte qu'il est possible de défaire les modifications. Les bases de données ne conservent en général que la dernière modification.

Certains éditeurs de texte sont très efficaces (puissants et rapides) pour effectuer des recherches et remplacements basés sur les expressions régulières, même à travers plusieurs dizaines de fichiers à la fois.

Lors d'une recherche textuelle, il est possible d'itérer sur chaque occurrence trouvée, et chacune d'elles est facilement localisable (sélectionnée dans le texte). Une recherche similaire dans une base de données produit une liste de fiches, sans indiquer l'emplacement de la chaine dans chaque fiche.

4.2 Avantages des bases de données

Les bases de données permettent l'accès et l'édition d'un fichier en simultané par plusieurs lexicographes. Les fichiers texte ne le permettent pas, à moins d'utiliser un système de gestion de versions de fichiers.

Les fichiers étant souvent chargés et sauvegardés en bloc (selon le logiciel d'édition), ils doivent être de taille limitée. Sinon, la lenteur des opérations rend la tâche du lexicographe irritante et inefficace. Il s'ensuit une répartition de l'information lexicale en plusieurs fichiers, habituellement regroupés selon leur catégorie grammaticale. Cette répartition alourdit quelque peu l'accès à l'information, car il faut premièrement déterminer la catégorie grammaticale à laquelle appartient le mot recherché, localiser et charger en mémoire le fichier correspondant, puis accéder à la section du fichier où se trouve la « fiche ». À l'inverse, les bases de données disposent de méthodes de chargement et d'indexation permettant une recherche rapide et transparente.

Les bases de données permettent de présenter une information différente, selon les besoins de chaque lexicographe, ou encore de présenter une même information mais de manière différente. Le fichier texte, quant à lui, offre toujours la même « structure » linéaire.

Les bases de données permettent des recherches complexes sur plusieurs champs, lesquels peuvent être de types différents. Par exemple : « identifier toutes les fiches qui ont le trait X dans le champ C1 ET (le trait Y dans le champ C2 OU le trait Z dans le champ C3) ». Les fichiers texte ne permettent que de lister les occurrences du trait (texte) X, Y ou Z.

Il est aussi possible d'appliquer une modification à l'ensemble des fiches extraites suite à une recherche complexe. Par exemple : « remplacer par la valeur 25 l'argument du trait X de toutes les fiches satisfaisant le critère P ». Il est virtuellement impossible d'accomplir cette opération avec des fichiers texte.

5 Avenir

À cause des grands avantages des bases de données (requêtes complexes, regroupement de l'ensemble des données sur un mot), nous caressons le projet de convertir l'ensemble de nos données lexicographiques en une seule base à vues multiples.

Toutefois, plusieurs faits nous font hésiter. Premièrement, la lourdeur du processus de conversion, notamment la création d'une interface spécifique, le risque de commettre des erreurs de masse (ou ponctuelle) qui pourraient n'être décelées que trop tard, et le besoin d'interrompre le travail des lexicographes pendant la migration.

Ensuite, les problèmes d'outils d'édition et d'interface, qui irritent déjà sur une base réduite comme le DicoSynAnt, ne ralentiront-ils pas tout le reste ? Enfin, pour profiter pleinement des capacités des SGBD, il faut formaliser l'information et ses liens; or certaines données ne sont peut-être pas facilement « découpables en champs ».

Peut-être une structure hybride SGBD-texte fait-elle partie de la solution ?

Application of Finite Graphs LGG in Information Extraction

Jeesun NAM

DICORA, Department of Linguistics and Cognitive Science
Hankuk University of Foreign Studies

89-Wangsan Mohyun Yongin-Si Kyunggi-Do KOREA[1]
namjs@hufs.ac.kr

Abstract

This paper describes an LGG(Local Grammar Graph)-driven methodology primarily based on accurate linguistic description to automatically extract information from the full-text documents. As a domain-specific document, we use a Korean electronic newspaper, JUNJA SINMUN, specialized in IT domain. This method demands well-defined linguistic patterns in lexico-syntactic levels that should be conceived by linguist-experts. Our research is not concerned with information extraction, ontology construction, nor a system implementation, but with detailed descriptions of LGG graphs that underline the importance of accurate linguistic knowledge for IE systems. In this paper, we first summarize semantic information types in domain-specific texts and describe the LGG-driven approach. We present some specific examples of LGG graphs and conclude with a discussion on problematic issues and future works. Our concern is less on theoretical discussion about NLP or IE systems, and more on the realities of the linguistic complexity and how they can be managed for application.

Keywords

Information Extraction, LGG graph, Semantic Knowledge Pattern, UNITEX GraphEditor

[1] This work was supported by Hankuk University of Foreign Studies Research Fund of 2007.

1 Introduction

The importance of Information Extraction(IE) is increasing in recent application areas, especially related to Information Retrieval Research, Semantic Web Technology or Ontology Constructing Methodology. The goal of IE is to automatically extract structured information understandable by machine (i.e. semantic information presented in an explicit and formal way) from unstructured machine-readable texts. Many algorithms or statistical methodologies have been proposed to extract relevant information from documents of well-defined domain. However the success of IE depends on the availability of high-quality linguistic knowledge that allows the recognition of accurate information from the given documents. As the linguistic knowledge cannot be accumulated in a short time, researchers have investigated systems for automatically extracting semantic information mostly by using a domain-dependant wrapper conceived with machine learning techniques.

This paper describes an LGG(Local Grammar Graph)-driven methodology primarily based on accurate linguistic description to automatically extract information from full-text documents. As a domain-specific document, we use a Korean electronic newspaper, JUNJA SINMUN, specialized in IT domain. This method requires well-defined linguistic patterns in lexico-syntactic levels that should be conceived by linguist-experts. Compared to the methodologies based on machine learning techniques, LGG-driven IE does not cover a large corpus but extracts more reliable semantic knowledge from given documents. While the approaches appear to be complementary, we note here several benefits of the LGG-driven approach. First, semantic information extracted by the LGG-driven approach can be much more complex than other approaches, since the LGG enables developers to describe very sophisticated semantic patterns in a flexible way by the finite local nature of graph representation. Second, effective improvement of the first result of IE is guaranteed, since it is easily performed according to the modifications and upgrades of the existing LGGs. Interaction between LGG developers and LGG graphs is controlled perfectly by UNITEX GraphEditor (http://www-igm.unv-mlv.fr/~unitex). Finally, the LGGs presented as Finite-State Graphs equivalent to FSA or FST can be used for other applications, for instance for building semantic patterns for RDF triples or Ontology schema as well as in extracting information from documents.

In this paper we will discuss our approach to extracting semantic information from texts, present a few LGG graphs constructed for domain-specific documents, and finally describe some remaining problems. Our research is not concerned with information extraction, ontology construction, nor a system implementation, but with detailed descriptions of LGG graphs that underline the importance of accurate linguistic knowledge for IE systems. The next section summarizes semantic information in domain-specific texts followed by a description of the LGG-driven approach. In Section 4 we present some specific examples of LGG graphs and Section 5 discusses some problematic issues and future works. Our concern is less on theoretical discussion about NLP or IE systems, and more on the realities of the linguistic complexity and how they can be managed for application.

2 Semantic Information in Domain-Specific Texts

In domain-specific texts such as JUNJA SINMUN, the Korean electronic newspaper about IT technology, the knowledge patterns are of limited numbers and explicitly described. We consider an example of the argument structures like <*Developer-To develop-Technique/Product*> below:

[2] <LGG-Develop> =:
{*Samsung Junja*(Samsung) – *Gaibalhada*(to develop) – *Cameraphone E800*(Cameraphone E800)}

The above semantic structure can be expressed in several linguistic patterns such as:

[3] *Samsung Junja-ga Cameraphone E800-eul gaibalha-ssda*
Samsung-ga(Nom) cameraphone E800-eul(Acc) gaibalha(develop)-essda(Past)
(Samsung has developed a Cameraphone E800)

[4] *Samsung-eso Cameraphone E800-eul seonboi-essda*
Samsung-eso(Nom) cameraphone E800-eul(Acc) senboi(present)-essda(Past)
(Samsung has presented a Cameraphone E800)

[5] *Cameraphone E800-i SamsungJunja-eso chulsidoi-essda*
Cameraphone E800-i(Nom) Samsung-eso(Agent) chulsideo(be made)-essda(Past)
(Cameraphone E800 has been made by Samsung)

When linguistic information is described at the maximum lexical level, the recognition of semantic knowledge will be performed more successfully than in systems where they are controlled by limited rules or abstract conceptual frames. The lexicalized description of linguistic units is not unattainable, since the vocabulary(terminology) and knowledge patterns in a given domain are of limited numbers. We describe these lexico-syntactic variations by LGG finite graphs.

Within our approach, we can make use of some domain-specific lexicons such as those of the names of persons, organizations, or locations that are easily acquired, but we can also generate a lexicon of the names of new products or new technologies that are scarcely available at the present time. Our LGG graphs are crucial in detecting the names of new products or new technologies by virtue of the linguistically well-defined contexts for these names.

3 Overview of LGG-driven Approach

3.1 Finite-State Local Automata

LGG-driven Approach we introduce here is proposed by Maurice Gross of LADL/IGM[2]. The goal of LGG(Local Grammar Graph) is to account for all the possible sentences in a given corpus, and this, with no exception(M. Gross 1997). The apparent obstacle to the realization of such an operation is avoided by the low complexity of the various automata necessary to the description of the domain-specific texts. These automata can be reused to describe the texts of other domains. This is similar to the way small molecules combine to produce larger ones in organic chemistry.

We consider here some typical sentences where the finite constraints can be exhaustively described in a local way (M. Gross 1997):

[6] *The Dow Jones industrial average (gained + lost) 15.40 points at 3,398.37*
[7] *The Dow Jones Industrial average finished with a (gain + loss)*
[8] *The Dow Jones Industrial average broke an all-time record of 5,000 points*

The above sentences present some variations of a Stock Exchange Index, the Dow Jones indexes. These sentences that indicate positive or negative values present common lexico-syntactic features such as:

- The subject is an Index,
- The verb expresses the direction of the variation,
- Two complements contain numerals which provide:
  - A relative variation, namely the difference with the previous quotation day: 15.40 in [6]. The variation is always a positive number, the sign being expressed by the verb,
  - And then, the full value of the Index: 3,398.37 or 5,000.

Such a point of departure is highly subjective. Firstly, the choice of the domain is completely semantic and secondly, it is determined by the intuition that the set of expressions is restricted, or perhaps closed. When we observe the texts of a given domain in a newspaper over a period of several months, we are given the impression that the vocabulary, the syntactic structures and the style of the domain are limited. Such a hypothesis needs to be verified carefully, and can only be confirmed experimentally. Once these local grammars are built, it is easy to use them to parse the texts and to verify their rate of success in areas of application.

[2] *Laboratoire d'Automatique Documentaire et Linguistique* and *Institut Gaspard Monge*.

3.2 Compiling LGG Graphs in UNITEX Platform

UNITEX is a corpus processing system designed particularly for building LGG-based resources and applying these resources and an automata-oriented technology to corpora. The concept of this software was conceived at LADL under the direction of Maurice Gross. Electronic resources such as electronic dictionaries and grammars can be accumulated and applied to corpora in UNITEX platform. Resource developers can work at the different levels of morphology, lexicon and syntax. The main functions are the following (http://www-igm.univ-mlv.fr/~unitex):

- Building, checking and applying electronic dictionaries
- Pattern matching with regular expressions and recursive transition networks
- Applying lexicon-grammar tables
- Handling ambiguity via the text automaton

UNITEX is a multilingual platform that allows users to virtually handle all the characters of all languages, including Asian languages. UNITEX is a multi-system software whose interface is written in Java and where all other programs are written in C/C++. Finally, UNITEX is a free software that is distributed under the terms of the General Public License (GPL). Everyone has access to the source code of all the UNITEX programs, which is included in the zip file he or she downloads. Users can modify it freely and include it in any GPL-licensed program.

4 Constructing LGG Graphs

In this section, we present the LGG graphs constructed to extract information about the development activities of new products or new technologies in IT domain. As we mentioned above, we are focused on the description of constructing LGG graphs rather than the discussion of theoretical importance. Most systems for IE work via a set of simple abstract patterns, where the results obtained by these patterns are either too large or lack precision. Therefore they do not extract complex semantic information, especially expressed with sophisticated linguistic variations in texts.

4.1 LGG-DevelopITProduct

Consider the examples mentioned above in [3]~[5]. For these three sentences we can build, in the first step, a very simple LGG as followings:

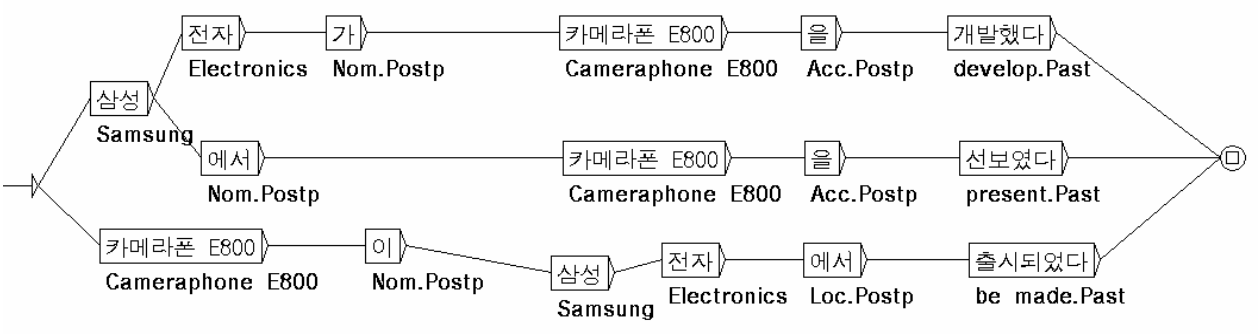


Figure 1. LGG-Develop 1.0

The above LGG can be ameliorated so that the following represents 10 sentence variations rather than 3:

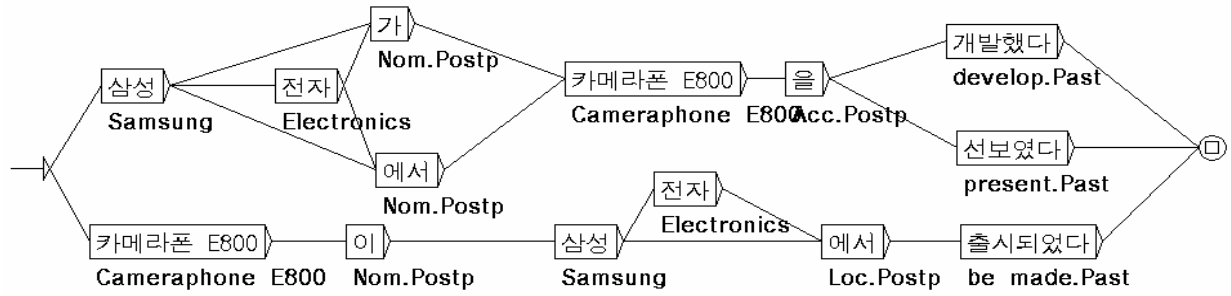


Figure 2. LGG-Develop 1.1

These graphs represent finite-state automata. Each has one initial state (left-most arrow) and one final state (right-most square). This representation can be managed in order to facilitate the effective construction of grammars by linguists. It corresponds to classical representations, even though in the latter states are represented as nodes, whereas nodes represent the transitions in the former. In these graphs there is no overt symbol for states. Nodes or arrows are labeled by the alphabet of the automaton, that is natural language words and/or their grammatical categories. To avoid multiplying parallel paths with words having identical roles, labels are grouped in boxes, hence a box of *n* words labels *n* arrows in the classical representation. Shaded boxes contain subautomata that are called into the graph by their name. For instance, instead of using a specific name *Samsung Electronics*, we can insert a shaded box such as *NameOfCompany* to represent or to recognize other names of companies from texts such as the following:

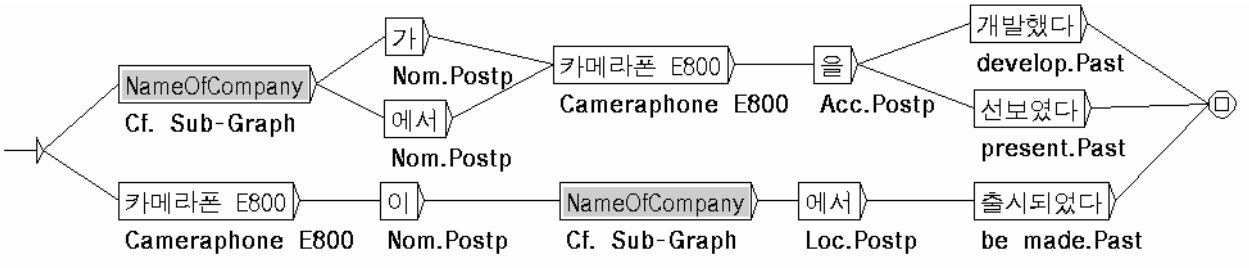


Figure 3. LGG-Develop 1.2

The shaded boxes titled *NameOfCompany* in Figure 3 are in fact assigned to the following graph:

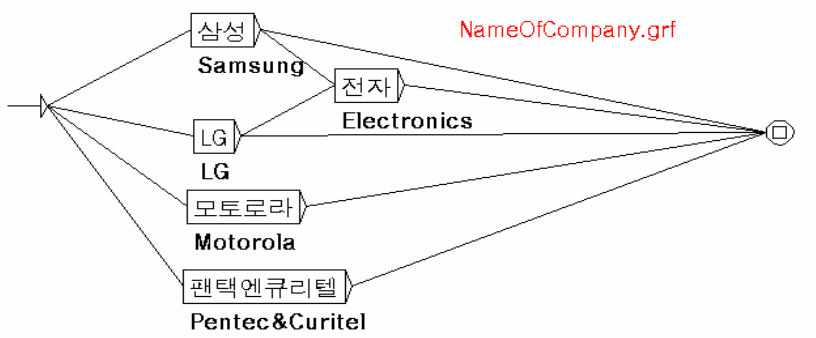


Figure 4. LGG-NameOfCompany 1.0

Now the graph in Figure 3 can be modified to create the following:

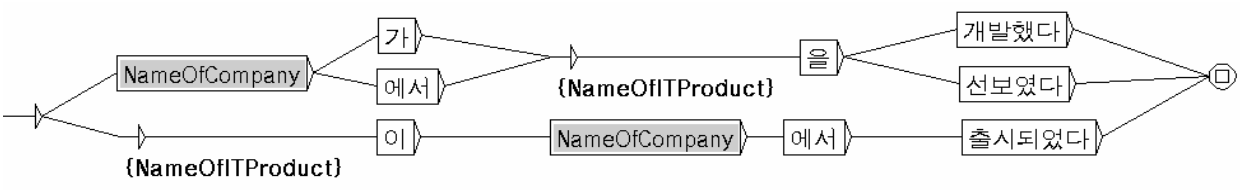


Figure 5. LGG-DevelopProduct 1.0

The LGG in Figure 5 allows us to generate a list of the names of new products or new technologies via the sentence patterns described around predicative terms such as to *develop*, *present* or *be made*, that are semantically equivalent. The noun phrases in the accusative or nominative position in these structures will be recognized as the names of new products or new technologies as predefined in these LGG graphs.

4.2 Extracting Information with LGGs

The LGG graphs constructed in our work do not only recognize semantic information in a given domain, but also allow the extraction of structured knowledge such as RDF triples. RDF syntax is based on a mathematical abstraction. It is defined as a set of triples. The following graph contains semantic annotation marks in the outputs of each box(transition) and enables us to indicate semantic information in RDF triple:

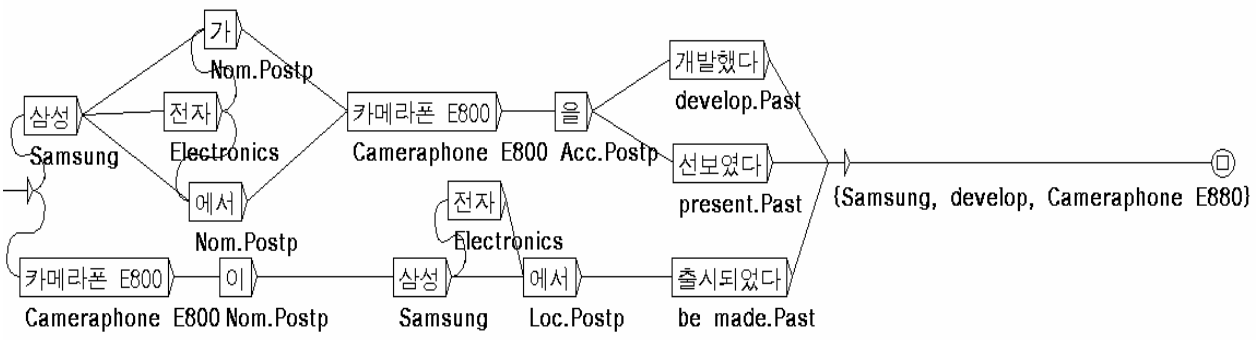


Figure 6. LGG-TripleSamsungProduct 1.0

The above LGG can be generalized in the following way:

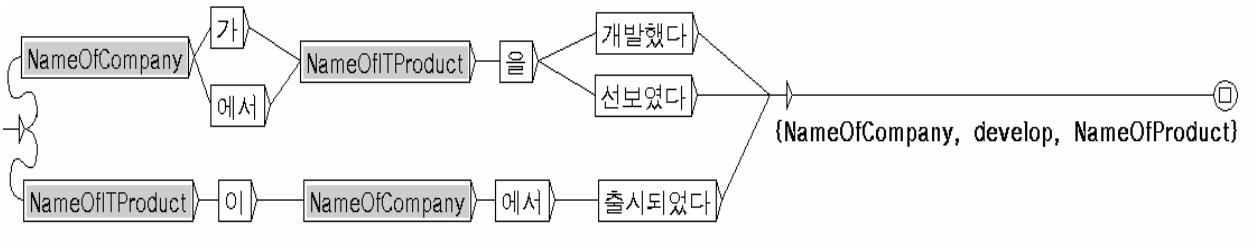


Figure 7. LGG-Triple{NameOfCompany,develop,NameOfProduct} 1.0

When the graph in Figure 7 is applied to the texts of IT domain, the generation of structured semantic information in RDF triples is performed according to annotation marks figured at the end of the graph. In UNITEX platform we can compile the LGG graphs into Finite-State Transducers(FST) that will be applied in any corpora of Unicode text type. The results we obtain via this processing will constitute the important part of semantic knowledge for domain-specific ontologies and other concrete applications as well as the IE system.

5 Problematic Issues and Future Work

We note that within the LGG-driven approach complex semantic knowledge can be extracted by virtue of the explicit and flexible descriptive nature of LGG graphs. However, even though complex LGG graphs have been well-designed, extracting candidate semantic information from web newspaper can result in a noise or, inaccurate information. We can get, as result, incorrect information due to noun phrase segmentation errors, incorrect parsing of adverbials or sentence modifiers, or material problems of web texts. The precision of linguistic description is extremely high compared to statistic or machine-learning approaches. However, we have come across some undesirable results. Consider for instance:

- *Geu bunya-eso Tobesoft-ga singisul-eul gaibalha-essda*
  This domain-in TOBESOFT-Nom.Postp new-technology-Acc.Postp develop-Past
  (TOBESOFT has developed a new technology in this domain)

Contrary to most situations, the noun phrases in the object position are not the names of new products, but generic terms such as *a new technology*. These kinds of nouns should not be extracted as a name of new products. Unless we use a ready-made list for the names of IT products, this problem cannot be avoided without some refinement of the existing LGGs.

The following example presents another type of problems:

- *Swach gerub-i chulsiha-n 65man dollar jjali chogoga sigye-ga…*
  SWACH-Nom.Postp present-Sfx.Modif 650,000$ expensive watch-Postp
  (The very expensive watch of 650,000$ that SWACH Corp. has presented…)

In the above example, the type of nouns in the object position is a syntactic sequence of several nouns such as *the very expensive watch of 650,000$*, whereas a predetermined list certainly does not contain this type of complex sequences as a name of product. Therefore, to resolve the present problematic issues we need to ameliorate the structure of our graphs.

For obvious reasons, computer science engineers or system developers have always attempted to describe the general features of human languages. But beyond these generalities lies an extremely complex set of dependencies between individual sequences, that is huge in size. The LGG-driven approach allows the explicit description and the realistic construction of classes of semantically equivalent utterances so that we can expect to obtain a much larger set of grammars of a given language by accumulating these well-defined local finite grammars.

Acknowledgements

We would like to thank Se-Young Park, Kwon-Yang Kim, Seung-Mi Chun, Seung-Hee Gho, So-Yun Kim, Ivan Berlocher and Tanya Stein for their helpful comments and for providing resources necessary for the completion of this work.

References

Gross, M. 1989. The Use of Finite Automata in the Lexical Representation of Natural Language, in *Electronic Dictionaries and Automata in Computational Linguistics*, Lecture Notes in Computer Science 377, pp.34-50, Berlin/New York Springer.

Gross, M. 1997. The Construction of Local Grammars, in *Finite-State Language Processing*, Ed. by E. Roche & Y. Schabes, The MIT Press.

Laporte, E., Monceaux, A. 1998/1999. Elimination of lexical ambiguities by grammars: The ELAG system, in *Lingvisticae Investigationes*: *Analyse lexical et syntaxique: Le système INTEX*, Cédric Fairon (ed.), Tome 22, John Benjamins Publishing Company, Amsterdam.

Paumier, S. 2000. Nouvelles methodes pour la recherche d'expressions dans de grands corpus, in A. Dister (ed.), *Actes des 3ème Journée INTEX, Revue Informatique et Statistique dans les Sciences Humaines*, 36ème année, 4-1.

Richardson, S., Dolan, W., Vanderwende, L. 1998. MindNet : acquiring and structuring semantic information from text, in *COLING*.

Roche, E. 1997. Parsing with Finite-State Transducers, in *Finite-State Language Processing*, Ed. by E. Roche & Y. Schabes, The MIT Press.

Silberztein, M. 1993. *Dictionnaires électroniques et analyse automatique de texts: Le système INTEX*, Paris, Masson.

Soderland, S. 1997. Learning to extact text-based information from the World Wide Web, in *Knowledg Discovery and Data Mining*.

WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture

François-Régis Chaumartin

Société Proxem
7 impasse Dumur
92110 Clichy – France
frc@proxem.com

Lattice/Talana – Université Paris 7
30 rue du château des rentiers
75013 Paris - France
fchaumartin@linguist.jussieu.fr

Résumé

Vous connaissez tous WordNet, mais en connaissez-vous tout ? Nous vous proposons ici, d'une part de redécouvrir WordNet (notamment en présentant les spécificités des versions les plus récentes) et d'autre part de découvrir d'autres ressources (lexicales, syntaxiques et sémantiques) qui s'y rattachent. Nous présentons également des techniques d'enrichissement automatique de WordNet, et des applications de TALN l'utilisant.

Mots-clés

WordNet, eXtended WordNet, VerbNet, FrameNet, SentiWordNet, WordNet Domains, WordNet-Affect, SemCor, Wikipédia, SUMO, Cyc, Web sémantique, ontologie

1 Introduction

WordNet est une ressource lexicale de large couverture, développée depuis plus de 20 ans pour la langue anglaise. Elle est utilisable librement, y compris pour un usage commercial, ce qui en a favorisé une diffusion très large. Plusieurs autres ressources linguistiques ont été constituées (manuellement ou automatiquement) à partir de, en extension à, ou en complément à WordNet. Des programmes issus du monde de l'Intelligence Artificielle ont également établi des passerelles avec WordNet.

L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Combinées, ces ressources fournissent un point de départ intéressant pour des développements sémantiques en TAL ou dans le cadre du Web sémantique, tels que la recherche d'information, l'inférence pour la compréhension automatique de textes, la désambiguïsation lexicale ou la résolution d'anaphores.

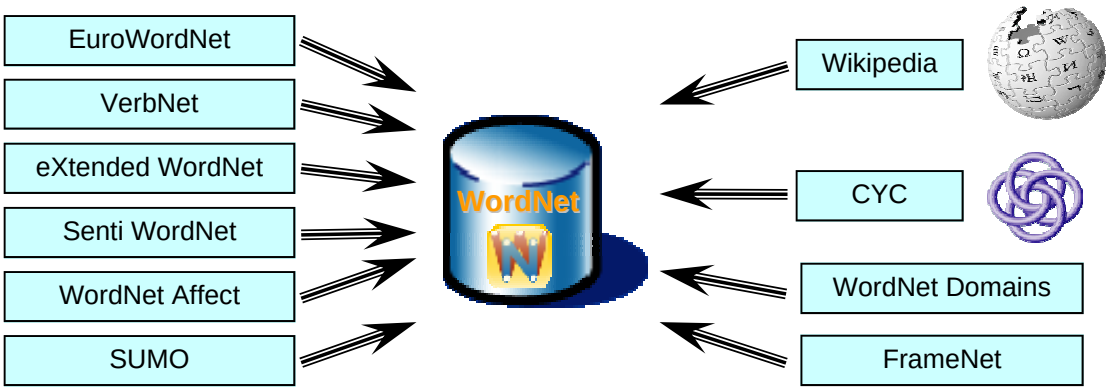


Figure 1 : Ressources disposant d'une traçabilité vers WordNet (liste non exhaustive)

2 WordNet

WordNet (Miller, 1995) est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991.

Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages.

S'il n'est pas exempt de critiques (granularité très fine, absence de relations paradigmatiques…), WordNet n'en reste pas moins l'une des ressources de TAL les plus populaires.

2.1 Notion de synset

Le *synset* (ensemble de synonymes) est la composante atomique sur laquelle repose WordNet. Un synset correspond à un groupe de mots interchangeables, dénotant un sens ou un usage particulier. Un synset est défini d'une façon différentielle par les relations qu'il entretient avec les sens voisins.

Les **noms** et **verbes** sont organisés en hiérarchies. Des relations d'hyperonymie (« est-un ») et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs « spécialisations ». Au niveau racine, ces hiérarchies sont organisées en types de base. Le réseau des noms est bien plus profond que celui des autres parties du discours. A titre indicatif, les deux premiers niveaux de la hiérarchie des noms se constituent des concepts abstraits suivants :

- **ABSTRACTION**: ATTRIBUTE, MEASURE/QUANTITY/AMOUNT, RELATION, SET, SPACE, TIME…
- **HUMAN ACTION**: ACTIVITY, COMMUNICATION, DISTRIBUTION, INACTIVITY, JUDGMENT, LEANING, LEGITIMATION, MOTIVATION, PROCLAMATION, PRODUCTION, SPEECH ACT…

- **ENTITY**: ANTICIPATION, CAUSAL AGENT, ENCLOSURE, EXPANSE, LOCATION, PHYSICAL OBJECT, SKY, SUBSTANCE, THING…
- **EVENT**: GROUP ACTION, NATURAL EVENT, MIGHT-HAVE-BEEN, MIGRATION, MIRACLE, NONEVENT, SOCIAL EVENT…
- **GROUP, GROUPING**: ASSOCIATION, BIOLOGICAL GROUP, PEOPLE, COLLECTION, AGGREGATION, COMMUNITY, ETHNIC GROUP, KINGDOM, MULTITUDE, POPULATION, RACE, RARE-EARTH ELEMENT…
- **PHENOMENON**: EFFECT/RESULT, LEVITATION, FORTUNE/CHANCE, REBIRTH, NATURAL PHENOMENON, PROCESS, PULSATION…
- **POSSESSION**: ASSETS, CIRCUMSTANCES, PROPERTY/MATERIAL POSSESSION, TRANSFERRED PROPERTY, TREASURE…
- **PSYCHOLOGICAL FEATURE**: COGNITION/KNOWLEDGE, FEELING, MOTIVATION/NEED…
- **STATE**: ACTION/ACTIVITY, EXISTENCE, STATE OF MIND, CONDITION, CONFLICT, DAMNATION, DEATH, DEGREE, DEPENDENCY, DISORDER, EMPLOYMENT, END, FREEDOM, ANTAGONISM, IMMATURITY, IMMINENCE, IMPERFECTION, INTEGRITY, MATURITY, OMNIPOTENCE, PERFECTION, PHYSIOLOGICAL STATE, RELATIONSHIP, STATE OF AFFAIRS, STATUS, TEMPORARY STATE, NATURAL STATE…

L'organisation des **adjectifs** est différente. Un sens « tête » joue un rôle d'attracteur ; des adjectifs « satellites » lui sont reliés par des relations de synonymie. On a donc une partition de l'ensemble des adjectifs en petits groupes. Les **adverbes** sont le plus souvent définis par les adjectifs dont ils dérivent. Ils héritent donc de la structure des adjectifs.

La version 3.0, la plus récente (janvier 2007) compte 117 597 synsets et 207 016 lemmes.

2.2 Relations

2.2.1 Relations sémantique (entre synsets)

Le tableau suivant présente un comptage des relations sémantiques de WordNet 2.1 par catégorie.

Relation	Entre	Nombre	Exemple
Hypernym/Hyponym	Verbe / verbe	13 124	EXHALE / BREATHE
	Nom / nom	75 134	CAT / FELINE
Instance Hyponym	Nom / nom	8 515	EIFFEL TOWER / TOWER
Part	Nom / nom	8 874	FRANCE / EUROPE
Member	Nom / nom	12 262	FRANCE / EUROPEAN UNION
Substance	Nom / nom	793	SERUM / BLOOD
Attribute	Adjectif / nom	643	INACCURATE / ACCURACY
Verb Group	Verbe / verbe	1 748	GELATINIZE#1 / GELATINIZE#2
Verb Entailment	Verbe / verbe	409	DREAM / SLEEP
Verb Cause	Verbe / verbe	219	ANESTHETIZE / SLEEP
Adjective Similar	Adjectif / adjectif	22 622	DYING / MORIBUND

Topic Domain	Nom / adjectif	1 108	COMPUTER SCIENCE / ADDRESSABLE
	Nom / nom	4 146	COMPUTER SCIENCE / COMPUTER
	Nom / adverbe	37	
	Nom / verbe	1 236	COMPUTER SCIENCE / CASCADE
Region Domain	Nom / adjectif	75	
	Nom / nom	1 246	FRENCH / FRANCE
Usage Domain	Nom / adjectif	227	
	Nom / nom	563	NEUTRALIZATION / EUPHEMISM
	Nom / adverbe	73	
	Nom / verbe	14	
See Also	Adjectif / adjectif	2 683	BLACK / DARK

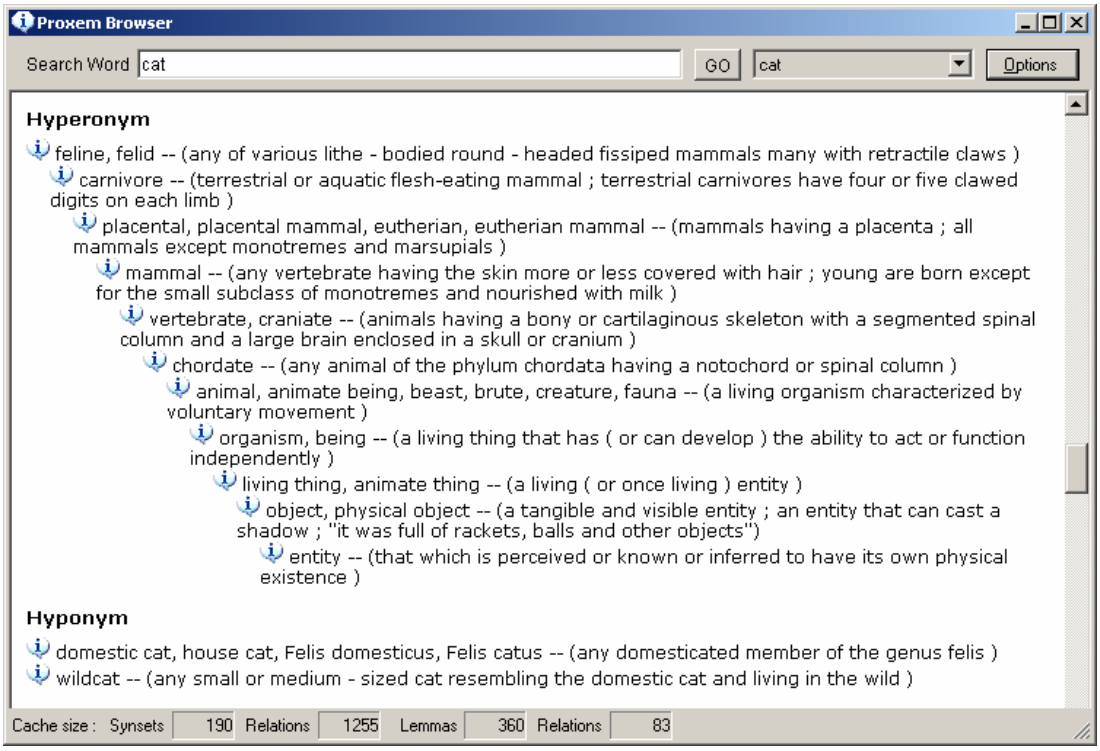
2.2.2 Relations lexicales (entre lemmes)

Le tableau suivant présente un comptage des relations lexicales de WordNet 2.1 par catégorie.

Relation	Entre…	…et	Nombre	Exemple
Usage Domain	nom	nom	379	
See Also	verbe	verbe	582	SLEEP LATE / SLEEP
Adjective Participle	adjectif	verbe	124	APPLIED / APPLY
Antonym	adjectif	adjectif	4 080	GOOD / BAD
	adverbe	adverbe	718	POORLY / WELL
	nom	nom	2 142	WINNER / LOOSER
	verbe	verbe	1 089	DIE / BE BORN
Pertainym	adjectif	nom	4 814	ACADEMIC / ACADEMIA
	adverbe	adjectif	3 213	BOASTFULLY / BOASTFUL
	adjectif	adjectif	38	
Derivation	nom	verbe	21 579	KILLING / KILL
	adjectif	nom	11 401	DARK / DARKNESS
	nom	nom	2 931	AUTOMOBILE / AUTOMOBILIST
	verbe	adjectif	1 508	KILL / KILLABLE
Adjective Cluster	adjectif	adjectif	1 290	STRIDENT / NOISY

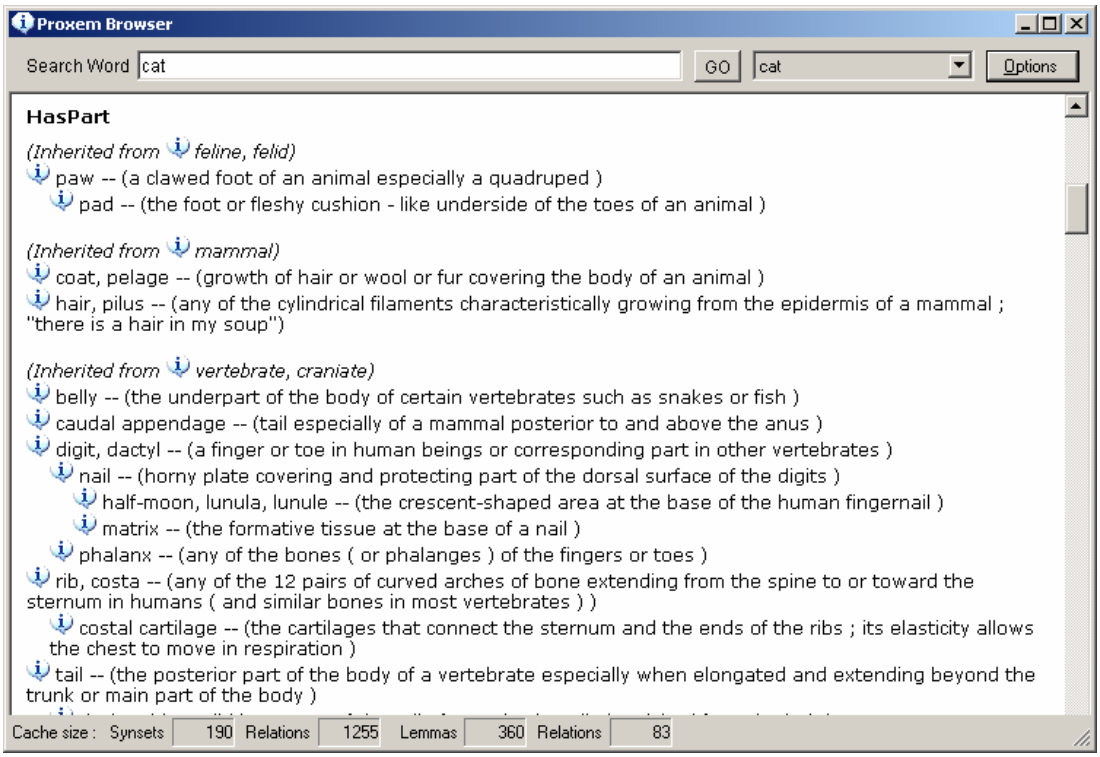
2.2.3 Exemples de relations d'hyperonymie et d'hyponymie

Par exemple, partant du sens le plus général du mot CAT#1 *(*le « chat » félin), on obtient une liste ordonnée d'ancêtres et de descendants, permettant de déterminer qu'un chat est un carnivore, un mammifère, un animal, etc.



2.2.4 Exemples de relations d'holonymie et de méronymie

Grâce à ces relations, on peut déterminer qu'un chat a des pattes, un pelage, une queue…



2.2.5 Notion d'instance hyponyme

La version 2.1 a introduit la notion d' « instance hyponyme », qui désigne une instance (et non une sous-classe) d'un synset (une Entité Nommée). Par exemple, GEORGE WASHINGTON est une instance hyponyme de PRESIDENT OF THE UNITED STATES. De même, le nom TOWER#1 a pour hyponymes SILO, MINARET, PYLON… et TOUR EIFFEL comme instance hyponyme.

2.3 Limites de WordNet

2.3.1 Informations manquantes

WordNet ne précise pas l'étymologie, la prononciation, les formes de verbes irréguliers et ne contient que des informations limitées sur l'usage des mots.

2.3.2 Profusion de sens pour un mot donné

La contrepartie de son importante couverture est que WordNet est très précis dans le sens des définitions. On a une granularité très (trop ?) fine des sens. Par exemple, le verbe *to give* (« donner ») n'a pas moins de 44 sens. Une telle profusion ne facilite pas une tâche de désambiguïsation lexicale.

2.3.3 Absence de relations pragmatiques

WordNet ne matérialise pas d'une façon formelle tout le sens contenu dans les définitions des termes. Par exemple, l'information qu'un chat ne rugit pas figure dans la définition, mais ne se retrouve formalisée dans aucune relation. De même, des relations pragmatiques telles que savon / bain (SOAP#1 / BATH#2) sont absentes de WordNet.

2.4 Mappage entre différentes versions

Il existe une correspondance des identifiants de synsets entre versions de WordNet. Ce mappage est indispensable pour assurer une traçabilité avec la version la plus récente. En effet, plusieurs ressources complémentaires à WordNet, et dignes d'intérêt, ont été définies pour la version 1.7 ou 2.0. Curieusement, le site Web de Princeton n'offre de mappage « officiel » que pour les noms et les verbes. Heureusement, d'autres sites proposent également des correspondances (construites automatiquement) pour les adjectifs et adverbes.

2.5 Corpus étiquetés par rapport à WordNet

A notre connaissance, peu de corpus sont étiquetés manuellement par rapport aux sens de WordNet. Nous pouvons citer le corpus SemCor (un sous-ensemble du corpus Brown), composé de 352 documents, comptant 2000 mots chacun approximativement. Plus précisément, le corpus

SemCor compte au total 676 546 mots (hors ponctuations). 234 135 noms, verbes, adjectifs et adverbes ont fait l'objet d'une désambiguïsation lexicale manuelle par rapport à WordNet 1.6, puis d'un mappage automatique vers les versions suivantes de WordNet (jusqu'à la 2.1). Ce corpus permet par exemple un début d'apprentissage automatique pour des tâches de désambiguïsation lexicale.

2.6 Fréquence des lemmes

WordNet donne une fréquence d'apparition pour chaque lemme définissant un synset. Ce nombre indique combien de fois un mot apparaît dans un sens spécifique. Pour un nom ou un verbe, la somme cumulée des fréquences d'un synset et de ses hyponymes au sein d'un sous-arbre de la hiérarchie permet de calculer son Contenu Informationnel.

2.7 Mesures de similarité

Une utilisation possible de l'ontologie fournie par WordNet est la définition de métriques heuristiques de « distance sémantique » entre les synsets. Cette métrique est basée sur la distance à parcourir dans le graphe, combinée ou non avec le Contenu Informationnel. Elle permet de quantifier la similarité de deux concepts. Elle peut également servir dans un cadre de désambiguïsation lexicale.

(Pedersen, Patwardhan, Michelizzi, 2004) présentent plusieurs de ces algorithmes de similarité entre mots, et une implémentation basée sur WordNet en Perl appelée WordNet::Similarity.

2.8 WordNets pour d'autres langues que l'anglais

2.8.1 EuroWordNet

EuroWordNet est une base de données pour plusieurs langues européennes. La phase initiale du projet s'est achevée en 1999, avec la conception de la base de données, ainsi que la définition de types de relations, d'un haut d'ontologie (63 éléments partagé par toutes les langues) et d'un Index-Inter-Langues (basé sur la version 1.5 du WordNet de Princeton).

EuroWordNet a produit des wordnets pour le néerlandais, l'italien, l'espagnol, l'allemand, le français, le tchèque et l'estonien. (À notre connaissance, les ressources pour le français ont été fournies par la société MemoData sur la base de son Dictionnaire Intégral.)

Les langues sont reliées ensemble par l'intermédiaire de l'Index-Inter-Langues. Il est ainsi possible de passer des mots dans une langue aux mêmes mots dans n'importe quelle autre langue. EuroWordNet permet donc une recherche d'information monolingue ou multilingue.

Langue	Synsets	Sens de mots	Relations internes à une langue	Relations d'équivalence entre langues différentes
WordNet 1.5	94 515	187 602	211 375	0

Ajouts à l'anglais	16 361	40 588	42 140	0
néerlandais	44 015	70 201	111 639	53 448
espagnol	23 370	50 526	55 163	21 236
italien	40 428	48 499	117 068	71 789
allemand	15 132	20 453	34 818	16 347
français	22 745	32 809	49 494	22 730
tchèque	12 824	19 949	26 259	12 824
estonien	7 678	13 839	16 318	9 004

Plusieurs autres groupes de recherche ont développé des wordnets dans d'autres langues en se basant sur les spécifications d'EuroWordNet (suédois, norvégien, danois, grec, portugais, basque, catalan, roumain, lithuanien, russe, bulgare et slovène).

On peut regretter que, contrairement à la version de Princeton, EuroWordNet ne soit pas distribué librement. Cela explique certainement sa diffusion beaucoup moins importante.

2.8.2 BalkaNet

BalkaNet prolonge la base de données d'EuroWordNet avec d'autres langues européennes : tchèque, roumain, grec, turc, bulgare, et serbe.

	bulgare	tchèque	grec	roumain	turc	serbe
Synsets	21 441	28 456	18 461	19 839	14 626	8 059
Noms	14 174	21 009	14 426	13 345	11 059	5 919
Verbes	4 169	5 155	3 402	4 808	2 725	1 803
Adjectifs	3 088	2 128	617	852	802	324
Adverbes	9	164	16	834	40	13
Lemmes	44 956	43 918	24 366	33 690	20 310	13 295

3 Autres ressources

3.1 VerbNet

VerbNet est un lexique des classes de verbes anglais. C'est un projet mené sous l'impulsion de Martha Palmer (d'abord à l'Université de Pennsylvanie, puis à Boulder au Colorado). VerbNet regroupe par classe les verbes partageant les mêmes comportements syntaxiques et sémantiques. C'est un prolongement des travaux de (Levin, 1993). (Chaumartin, 2006) décrit comment mettre en œuvre WordNet et VerbNet pour implémenter une interface syntaxe-sémantique.

Une classe de verbes regroupe plusieurs verbes, et identifie des rôles thématiques avec d'éventuelles contraintes de sélection. Elle décrit plusieurs constructions typiques (des « *frames* ») des verbes membres. La sémantique de l'action ou de l'événement est également précisée. Des sous-classes permettent de décrire d'éventuelles spécialisations d'une classe. On peut en trouver une description dans (Kipper-Schuler, 2003). La ressource la plus proche pour les verbes français nous semble être le lexique-grammaire du LADL (Gross, 1994).

La version la plus récente (VerbNet 2.1) distingue 237 classes de verbes qui regroupent 4991 sens de verbes. Un verbe membre d'une classe est souvent accompagné d'une précision sur le *synset* correspondant, qui permet d'identifier dans WordNet le sens précis du verbe. VerbNet dispose aussi d'un mappage vers FrameNet. Une API en Java est également disponible.

3.1.1 Structure d'une description de classe de verbes

Chaque fichier de VerbNet décrivant une classe de verbes est représenté en XML, et découpé en sections balisées selon une structure arborescente :

- `<MEMBERS>` décrit les verbes membres qui appartiennent à la classe, en précisant l'identifiant vers le(s) synset(s) correspondant(s) de WordNet,
- `<THEMROLES>` indique les rôles thématiques de la classe :
  - `<SELRESTRS>` précise leurs éventuelles contraintes de sélections,
- `<FRAMES>` indique chacune des constructions typiques, en donnant à chaque fois :
  - `<SYNTAX>` sa syntaxe,
  - `<SEMANTICS>` sa sémantique,
  - `<EXAMPLES>` un ou plusieurs exemples,
- `<SUBCLASSES>` regroupe éventuellement en sous-classes :
  - `<VNSUBCLASS>` les cas particulier d'une classe de verbes.

3.1.2 Un exemple : la classe de verbe "murder"

Par exemple, le fichier *murder.xml* décrit trois constructions typiques :

- *Agent* élimine *Patient* (« Brutus tua Jules César »),
- *Agent* élimine *Patient* avec *Instrument* (« Brutus tua César avec un poignard »),
- *Instrument* élimine *Patient* (« le pesticide tua les insectes »).

Chaque description de classe de verbes déclare des contraintes de sélection sur les rôles thématiques. Par exemple, pour "**murder**", l'*Agent* et le *Patient* doivent avoir un trait *Animé* (en pratique, *Humain* ou *Organisation*) et l'*Instrument* doit être *Concret*.

3.1.2.1 Description de la syntaxe

La deuxième *frame* de la classe de verbe "**murder**" décrit :

- ```
  <SYNTAX>
        <NP value="Agent"/>
        <VERB/>
        <NP value="Patient"/>
        <PREP value="with"/>
        <NP value="Instrument"/>
  </SYNTAX>
  ```

- ```
  <EXAMPLES>
        <EXAMPLE>"Brutus killed Caesar with a knife"</EXAMPLE>
  </EXAMPLES>
  ```

3.1.2.2 Description de la sémantique

Par exemple, pour "**murder"** :

- au démarrage de l'événement, *Patient* est vivant : *alive(start(E), Patient),*
- à la fin de l'événement, *Patient* n'est plus vivant : ***!*** *alive(result(E), Patient).*

3.1.3 Prise en compte de l'héritage entre classes

La balise <SUBCLASSES> déclare les éventuelles sous-classes qui spécialisent une classe de verbe donnée. Une sous-classe permet :

- De raffiner les contraintes de sélection portant sur les rôles thématiques,
- De déclarer de nouveaux rôles thématiques,
- D'associer des verbes à la sous-classe,
- De créer de nouvelles *frames*.

3.2 FrameNet

FrameNet (Baker, Fillmore & Lowe, 1998), projet mené à Berkeley à l'initiative de Charles Fillmore, est fondé sur la sémantique des cadres ("*frame semantics*"). FrameNet a pour objectif de documenter la combinatoire syntaxique et sémantique pour chacun des sens d'une entrée lexicale à travers une annotation manuelle d'exemples choisis dans des corpus sur des critères de représentativité lexicographique. Les annotations sont ensuite synthétisées dans des tables, qui résument pour chaque mot les cadres avec leurs actants sémantiques et arguments syntaxiques.

FrameNet II compte actuellement 825 cadres sémantiques, 10 000 unités lexicales (dont 6 100 complètements annotées) ainsi que 130 000 phrases d'exemples annotés. La totalité des outils et données est (en principe) distribuée librement.

Un mappage entre les verbes de FrameNet II et ceux de WordNet peut être trouvé sur http://www.cs.unt.edu/~rada/downloads.html#verbmap.

3.2.1 Exemple de description du cadre "Crime_scenario"

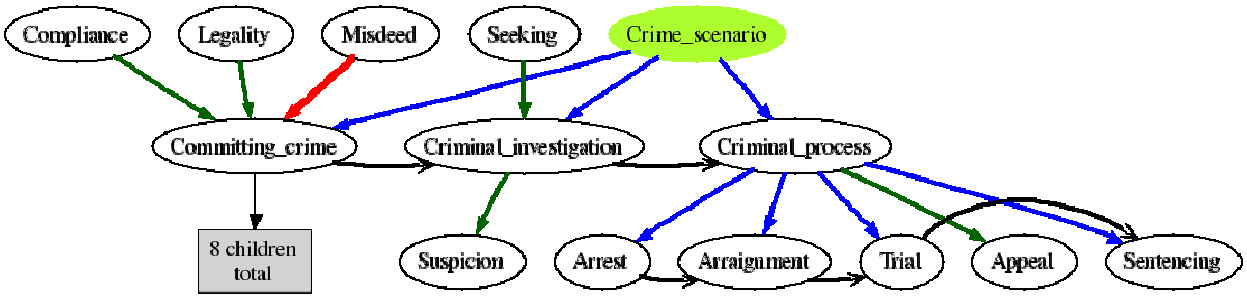
3.2.1.1 Description

A (putative) Crime is committed and comes to the attention of the Authorities. In response, there is a Criminal_investigation and (often) Arrest and criminal court proceedings. The Investigation, Arrest, and other parts of the Criminal_Process are pursued in order to find a Suspect (who then may enter the Criminal_process to become the Defendant) and determine if this Suspect matches the Perpetrator of the Crime, and also to determine if the Charges match the Crime. If the Suspect is deemed to have committed the Crime, then they are generally given some punishment commensurate with the Charges.

3.2.1.2 Frame Elements

Authorities []	The group which is responsible for the maintenance of law and order, and as such have been given the power to investigate Crimes, find Suspects and determine if a Suspect should be submitted to the Criminal_process.
Charge []	A description of a type of act that is not permissible according to the law of society.
Crime []	An act, generally intentional, that matches the description that belongs to an official Charge.
Perpetrator []	The individual that commits a Crime.
Semantic type	Sentient
Suspect []	The individual which is under suspicion of having committed the Crime.

3.2.2 Exemple de relations entres cadres



3.3 eXtended WordNet

3.3.1 Présentation

eXtended WordNet (XWN) est un projet mené en 2003 à l'Université de Dallas, qui enrichit WordNet 2.0. XWN produit une analyse syntaxique de la définition de chaque synset, la désambiguïsation lexicale de chaque mot de la définition, puis un passage en forme logique.

(Moldovan & Novischi, 2002) décrivent comment XWN permet d'améliorer sensiblement les résultats d'un système de Questions-Réponses.

3.3.2 Exemple

Par exemple, le nom COUSIN#1, dont la définition est "*the child of your aunt or uncle*" (« l'enfant de votre tante ou de votre oncle »), a pour analyse syntaxique :

```
(TOP (S (NP (NN cousin) )
        (VP (VBZ is)
                (NP (NP (DT the) (NN child) )
                        (PP (IN of)
                                (NP (PRP$ your) (NN aunt) (CC or) (NN uncle) ) ) ) )
(. .) ) )
```

Ainsi que la forme logique suivante :

```
cousin:NN(x1) -> child:NN(x1) of:IN(x1, x4) aunt:NN(x2) or:CC(x4, x2, x3) uncle:NN(x3)
```

3.3.3 Caractéristiques

Les informations présentes dans XWN sont de qualité *gold* (validé humainement), *silver* (accord entre deux analyseurs syntaxiques) ou *normal*. Si on considère l'analyse des définitions, on a :

Synsets (WN 2.0)	Nombre de définitions	Mots de classe ouverte	Mots mono-sémiques	Qualité *gold*	Qualité *silver*	Qualité *normal*
Noms	79 689	505 946	138 274	10 142	45 015	296 045
Verbes	13 508	48 200	6 903	2 212	5 193	30 813
Adjectifs	18 563	74 108	14 142	263	6 599	50 359
Adverbes	3 664	8 998	1 605	1 829	385	4 920

Du fait de la complexité de la tâche de désambiguïsation lexicale, et de l'absence de validation humain systématique, il est sage de penser que seule les mots étiquetés avec la qualité *gold* sont correctement désambiguïsés (ils ne représentent que 3,2% des mots polysémiques), et que les autres mots contiennent une proportion importante de contresens.

3.4 WordNet Domains

WordNet Domains (Magnini et Cavaglià, 2000) est une extension multilingue de WordNet 2.0, développée à l'Instituto Trentino di Cultura (ITC-irst). La notion de domaine a été employée aussi bien en linguistique qu'en lexicographie pour marquer des usages des mots. Les domaines sémantiques offrent une manière naturelle d'établir des relations sémantiques entre les sens des mots, qui peuvent être utilisée avec profit en informatique linguistique. Dans WordNet Domains, chaque synset est annoté avec au moins une étiquette de domaine (par exemple *Sport*, *Politique*, *Médecine*, *Economie*…), choisie dans un ensemble d'environ deux cents étiquettes organisées hiérarchiquement.

Un domaine peut inclure des synsets de différentes parties du discours et de différentes sous-hiérarchies de WordNet. Par exemple le domaine *Médecine* regroupe des sens de noms tels que DOCTOR#1 (le 1er sens du mot docteur) et HOSPITAL#1, et de verbes comme OPERATE#7.

L'information apportée par ces domaines est complémentaire à celles déjà présentes dans WordNet. Les domaines peuvent créer des regroupements homogènes des sens d'un même mot, avec comme effet secondaire de réduire la polysémie des mots dans WordNet.

3.4.1 Exemple : les domaines associés aux différents sens du nom "bank"

Le mot *bank*, par exemple, a dix sens dans WordNet 2.0. Trois d'entre eux (BANK#1, BANK#3 et BANK#6) sont regroupés au sein du domaine *Economie*, tandis que deux (BANK#2 et BANK#7) sont regroupés avec les étiquettes de domaine *Géographie* et *Géologie*.

Sens	Synset (Définition)	Domaines
#1	depository financial institution, bank, banking concern, banking company (a financial institution...)	*Economy*
#2	bank (sloping land ...)	*Geography, Geology*
#3	bank (a supply or stock held in reserve...)	*Economy*
#4	bank, bank building (a building...)	*Architecture, Economy*
#5	bank (an arrangement of similar objects...)	*Factotum*
#6	savings bank, coin bank, money box, bank (a container...)	*Economy*
#7	bank (a long ridge or pile...)	*Geography, Geology*
#8	bank (the funds held by a gambling house...)	*Economy, Play*
#9	bank, cant, camber (a slope in the turn of a road...)	*Architecture*
#10	bank (a flight maneuver...)	*Transport*

3.4.2 Intérêt

L'utilisation de WordNet Domains permet par exemple d'améliorer l'efficacité d'algorithmes de désambiguïsation lexicale et d'expansion de requêtes.

3.5 WordNet-Affect

La détection de connotations affectives dans les textes a des intérêts économiques réels : par exemple, une société peut chercher à détecter, en analysant la blogosphère ou les *news*, s'il se dit du bien ou du mal de ses produits.

Basé sur WordNet Domains, WordNet-Affect (Strapparava & Valitutti, 2004) est une ressource linguistique pour la représentation lexicale de connaissances sur les affects.

Un sous-ensemble de synsets de WordNet appropriés est choisi pour représenter des concepts affectifs. On ajoute des informations additionnelles aux synsets affectifs, en leur associant une ou plusieurs étiquettes qui précisent une signification affective. Par exemple, les concepts affectifs représentant un état émotif sont représentés par des synsets marqués par l'étiquette *Émotion*. Le tableau suivant liste ces étiquettes affectives, avec des exemples de synsets associés :

Etiquette affective	Exemples de synsets associés
Emotion	nom ANGER#1, verbe FEAR#1
Mood	nom ANIMOSITY#1, adjectif AMIABLE#1
Trait	nom AGGRESSIVENESS#1, adjectif COMPETITIVE#1
Cognitive State	nom CONFUSION#2, adjectif DAZED#2
Physical State	nom ILLNESS#1, adjectif ALL IN#1

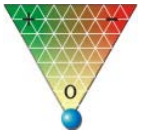
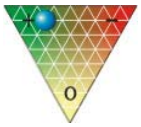
Edonic Signal	nom HURT#3, nom SUFFERING#4
Emotion-Eliciting Situation	nom AWKWARDNESS#3, adjectif OUT OF DANGER#1
Emotional Response	nom COLD SWEAT#1, verbe TREMBLE#2
Behaviour	nom OFFENSE#1, adjectif INHIBITED#1
Attitude	nom INTOLERANCE#1, nom DEFENSIVE#1
Sensation	nom COLDNESS#1, verbe FEEL#3

WordNet-Affect a été développé en deux étapes. La première a consisté à identifier manuellement un premier « noyau » de synsets affectifs. La deuxième étape a permis, en suivant les relations définies dans WordNet, de propager les informations de ce noyau à son voisinage.

3.6 SentiWordNet

SentiWordNet (Esuli & Sebastiani, 2006) est une ressource lexicale permettant le sondage d'opinion. SentiWordNet assigne à chaque synset de WordNet 2.0 trois valeurs : Positivité, Négativité, Objectivité (respectant l'égalité : Positivité + Négativité + Objectivité = 1). Cette ressource a été créée d'une façon semi-automatisées, en mixant des techniques linguistiques et statistiques (utilisation de classifieurs).

Avec cette classification, on a par exemple pour les trois sens de l'adjectif "*estimable*" :

	P = 0 N = 0 O = 1	COMPUTABLE#1 **ESTIMABLE**#3 *may be computed or estimated; "a calculable risk"; "computable odds"; "estimable assets"*
	P = 0,75 N = 0 O = 0,25	**ESTIMABLE**#1 *deserving of respect or high regard*
	P = 0,625 N = 0,25 O = 0.125	HONORABLE#5 GOOD#4 RESPECTABLE#2 **ESTIMABLE**#2 *deserving of esteem and respect; "all respectable companies give guarantees"; "ruined the family's good name"*

3.7 SUMO (Suggested Upper Merged Ontology)

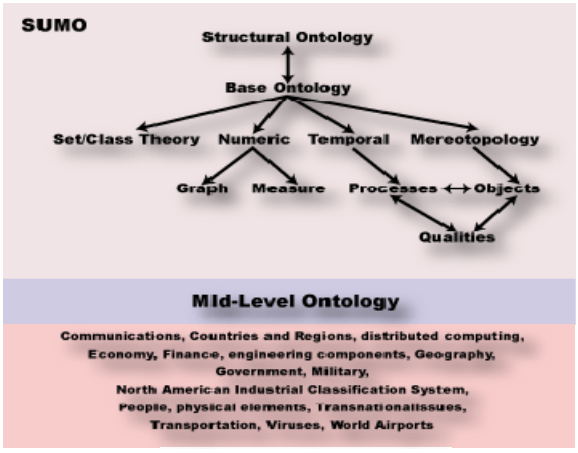
3.7.1 Notion de « haut » d'ontologie

Les ontologies sont des artefacts construits en fonction d'une tâche précise. Force est de constater qu'une ontologie donnée ne semble pas pouvoir être facilement réutilisée pour une tâche autre que celle qui a motivé sa construction originelle.

Il découle de ce constat de nombreuses recherches sur la réutilisabilité du « haut » des ontologies, avec pour argumentaire : puisqu'il est difficile, voire impossible, de réutiliser directement des ontologies, trop proches de vues détaillées qu'on peut avoir sur un domaine, intéressons-nous au « haut » de l'ontologie. Cette *Upper Ontology* répertorie et organise de grandes catégories de la

pensée ou de la société humaine qui devraient pouvoir être réutilisables dans de très nombreuses applications et être alors « génériques ».

L'objectif du groupe *Standard Upper Ontology* est de réfléchir, puis soumettre à la normalisation, la constitution d'un haut d'ontologie qui se voudrait universel pour les grandes catégories d'objets et de pensées. Le résultat est SUMO (*Suggested Upper Merged Ontology*), qui vise à s'imposer en tant que standard, et commence à être utilisée notamment pour le Web sémantique. MILO (*MId-Level Ontologies*) est un ensemble d'ontologies multi domaines, de niveau intermédiaire, créées en se basant sur SUMO.



SUMO (Niles & Pease, 2003) est écrit en langage SUO-KIF, dérivé simplifié de KIF (*Knowledge Interchange Format*), qui est un langage équivalent à la logique du premier ordre. Une traduction vers OWL (le langage du Web sémantique) est également disponible.

L'ensemble compte 20 000 termes et 60 000 axiomes. Il existe un mappage complet de SUMO vers les différentes versions de WordNet (jusqu'à la version 2.1), y compris pour MILO (les ontologies de niveau intermédiaire).

3.7.2 Exemple : le concept « beverage »

Définition : Any food that is ingested by drinking. Note that this class is disjoint with the other subclasses of food, i.e. meat and fruit or vegetable.

Sous-classes : Milk, AlcoholicBeverage, Coffee, Tea

Axiomes (traduits automatiquement en anglais à partir de l'expression en KIF) :

Food is disjointly decomposed into Meat, Beverage

for all beverage *?BEV* holds Liquid is an attribute of *?BEV*

for all drinking *?DRINK* holds if *?BEV* is a patient of *?DRINK*, then *?BEV* is an instance of Beverage

for all Cup *?CUP* holds if contains(*?CUP*, *?STUFF*), then *?STUFF* is an instance of Beverage

for all Tavern *?COMPANY* holds there exist CommercialService *?SERVICE*, beverage *?BEVERAGE* so that *?SERVICE* is an agent of *?COMPANY* and *?BEVERAGE* is a patient of *?SERVICE*

3.8 Cyc

Cyc est un projet d'Intelligence Artificielle lancé en 1984 par Doug Lenat. Cyc vise à regrouper une ontologie et une base de données complètes sur le sens commun, pour permettre à des applications d'I.A. d'effectuer des raisonnements similaires à ceux des humains.

Des fragments de connaissances typiques sont par exemple : « les chats ont quatre pattes » ; « Paris est la capitale de la France ». Elles contiennent des termes (PARIS, FRANCE, CHAT…) et des assertions (« Paris est la capitale de la France ») qui relient ces termes entre eux. Grâce au moteur d'inférence fourni avec la base Cyc, il est possible d'obtenir une réponse à une question comme « Quelle est la capitale de la France ? ».

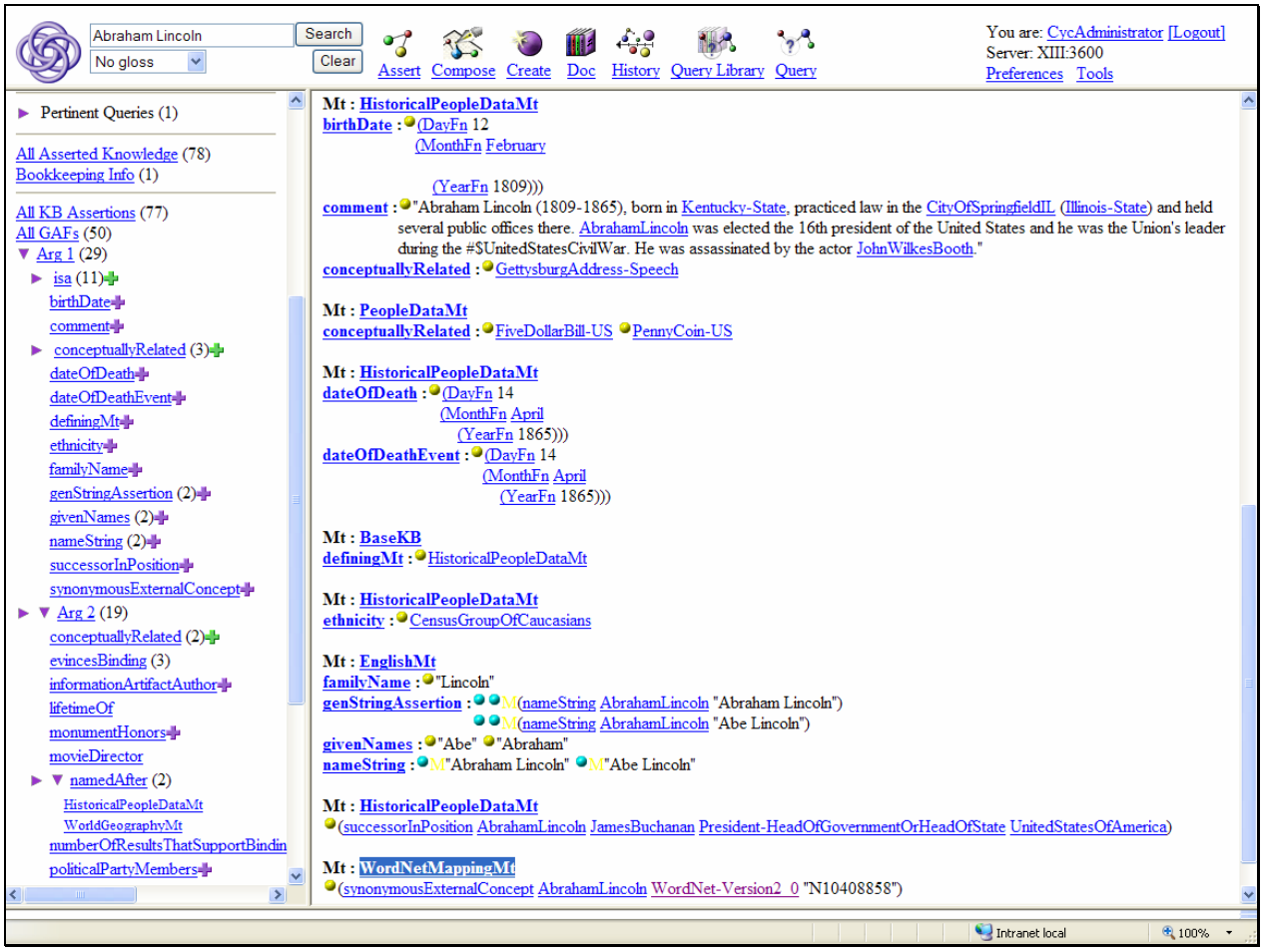


Figure 2 : Interface Web du serveur ResearchCyc (page de description de ABRAHAMLINCOLN)

La base Cyc contient des millions d'assertions (faits et règles) rentrées à la main. Elles sont écrites en langage CycL, qui est un langage logique avec une syntaxe proche de celle de LISP.

La base de connaissance est divisée en plusieurs milliers de micro-théories (Mt), collections de concepts et faits concernant typiquement un domaine particulier de la connaissance. Une micro-théorie est donc un ensemble d'assertions qui partagent le même point de vue : un domaine

particulier, un certain niveau de détail, un certain intervalle de temps, etc. À la différence de la base de connaissance dans son ensemble, chaque micro-théorie doit être exempte de contradictions. Par exemple, Philadelphie était la capitale des Etats-Unis de 1790 à 1800. Dans une micro-théorie couvrant l'intervalle de temps 1790-1800, l'assertion (#$CAPITALCITY #$UNITEDSTATES #$PHILADELPHIA) sera vraie, et dans une micro-théorie couvrant le XX^{ème} siècle, (#$CAPITALCITY #$UNITEDSTATES #$WASHINGTON) sera également vraie.

ResearchCyc 1.0 est la version réservée au monde de la recherche. Elle compte 300 000 concepts et 3 000 000 d'assertions (faits et règles) utilisant 26 000 relations. Des modules en langage naturel permettent de poser des questions et de rentrer de nouveaux faits sans avoir besoin de connaître CycL. La version OpenCyc 1.0 est librement accessible, mais ne contient qu'un sous ensemble de ces règles et assertions.

Les deux versions contiennent à ce jour une correspondance partielle entre les concepts de Cyc et les synsets de WordNet 2.0. Approximativement 11 300 synsets (8800 noms, 2110 verbes, 330 adjectifs et 35 adverbes) sont liés aux concepts de Cyc.

3.9 Wikipédia

Wikipédia est une encyclopédie libre et multilingue écrite de façon collaborative sur Internet avec la technologie wiki. Plusieurs projets visent à établir automatiquement des liens entre la Wikipédia et WordNet.

(Ruiz-Casado, Alfonseca, Castells, 2005) présentent l'implémentation d'un algorithme rapide permettant de réaliser la correspondance entre un article de la *Simple Wikipedia*[1] et le synset correspondant de WordNet. Si aucun synset n'a de lemme en commun avec le titre de l'article, ce dernier est ignoré. Si un seul synset de WordNet a un lemme égal au titre, l'article y est lié sans autre analyse. En cas d'ambiguïté, l'article fait l'objet d'un étiquetage morphosyntaxique (après un filtrage des marqueurs syntaxiques spécifiques à la *Wikipedia*), pour ne conserver que les noms, verbes et adjectifs. Le système analyse les définitions de WordNet, et construit pour chacune d'entre elles un vecteur booléen (contenant « 1 » pour chaque terme en commun avec l'article et « 0 » pour chaque mot en disjonction). L'algorithme calcule alors une mesure de type cosinus entre les vecteurs, et retient le meilleur article, au sens de cette mesure de similarité. Les auteurs revendiquent une précision de 91,11% (83.89% sur les mots polysémiques).

(Chaumartin, 2007) présente une généralisation de ce type d'approche, où WordNet est en plus enrichi avec des nouveaux synsets, avec une identification automatique du bon hyperonyme. La précision de l'appariement entre WordNet 2.1 et un sous-ensemble de la Wikipedia anglaise (autour de 15 800 articles) est de 92% ; en cas de création de nouveau synset, l'hyperonyme est correctement identifié dans 85% des cas.

[1] Une version en anglais simplifié de la Wikipédia.

4 Conclusion

Nous avons présenté en détail WordNet, ainsi que plusieurs autres ressources de nature lexicale, syntaxique et sémantique, qui s'y rattachent. Le fait de mettre en commun plusieurs ressources de large couverture permet d'espérer des progrès dans les applications de TAL. Pour finir, citons quelques projets qui combinent plusieurs de ces ressources.

(Shi, Mihalcea, 2005) revendique la construction d'un analyseur sémantique robuste en langue anglaise, en utilisant WordNet, VerbNet et FrameNet.

Notre projet, ISIDORE[2] (en cours de réalisation), combine WordNet, VerbNet, eXtended WordNet et SUMO. Il vise à extraire des connaissances d'une encyclopédie. Nous espérons disposer fin 2008 d'une indexation sémantique de 15 000 articles de la Wikipedia en anglais.

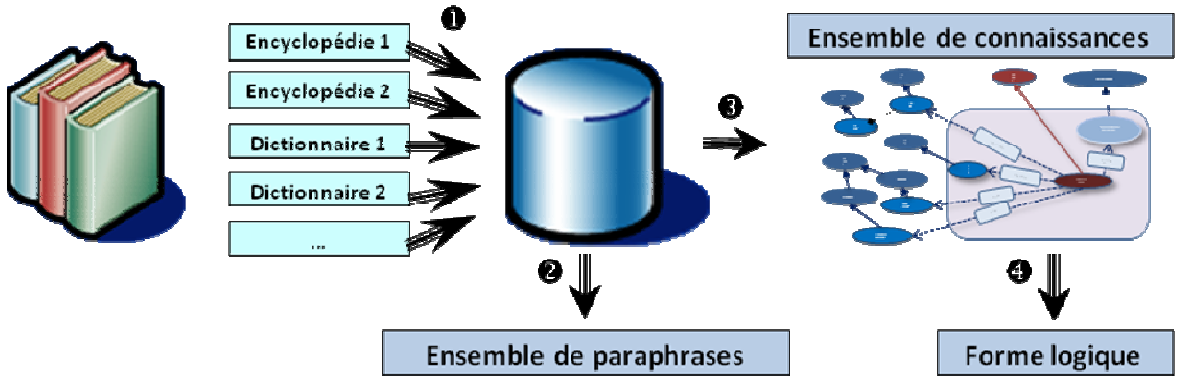


Figure 3 : Architecture d'ensemble du projet ISIDORE

Bibliographie

Andreevskaia A., Bergler S. 2006. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. Actes de *EACL-06, 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italie.

Baker C., Fillmore C., Lowe J. 1998. The Berkeley FrameNet project. Actes de *17th international conference on Computational linguistics.*

Bentivogli L., Forner P., Magnini B., Pianta E. 2004. Revising WordNet Domains Hierarchy: Semantics, Coverage, and Balancing. In *COLING 2004 Workshop on "Multilingual Linguistic Resources"*, Genève, Suisse, pp. 101-108.

Chaumartin F. 2006. Construction automatique d'interface syntaxe-sémantique utilisant des ressources de large couverture en langue anglaise. Actes de *TALN 2006*, 729-735.

[2] Saint-Isidore (560-636), patron des informaticiens, fut l'auteur des *Etymologies*, une encyclopédie en 20 livres.

Chaumartin F. 2007. Extraction de paraphrases désambiguïsées à partir d'un corpus d'articles encyclopédiques alignés automatiquement. Actes de *TALN 2007* (à paraître).

Esuli A., Sebastiani F. 2006. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. Actes de *LREC 2006, fifth international conference on Language Resources and Evaluation*, pp. 417-422.

Gross M. 1994. Constructing Lexicon-grammars. In *Computational Approaches to the Lexicon*, Atkins and Zampolli (eds.), Oxford Univ. Press, pp. 213-263.

Kipper-Schuler K. 2003. *VerbNet: a broad coverage, comprehensive, verb lexicon.* Ph.D. Thesis, University of Pennsylvania.

Levin B. 1993. *English Verb Classes and Alternation: A Preliminary Investigation.* Chicago, IL: University of Chicago Press.

Magnini B., Cavaglià G. 2000. Integrating Subject Field Codes into WordNet. Actes de *LREC-2000, Second International Conference on Language Resources and Evaluation*, Athènes, Grèce, pp. 1413-1418.

Moldovan D., Novischi A. 2002. Lexical Chains for Question Answering, Actes de *COLING 2002*.

Miller G., Fellbaum C., Miller K. 1993. *Five Papers on WordNet*.

Miller G. 1995. Wordnet: A lexical database. Actes de *ACM 38*, pp. 39-41.

Niles I., Pease A. 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. Actes de *2003 International Conference on Information and Knowledge Engineering (IKE '03)*, Las Vegas, Nevada.

Pedersen T., Patwardhan S., Michelizzi J. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. Actes de *Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, San Jose, CA.

Ruiz-Casado M., Alfonseca E., Castells P. 2005. Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets. Actes de *AWIC*, 380-386.

Shi L., Mihalcea R. 2005. Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. Actes de *CICLing 2005*, Mexico.

Strapparava C., Valitutti A. 2004. WordNet-Affect: an Affective Extension of WordNet. Actes de *4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbonne, pp. 1083-1086.

Valitutti A., Strapparava C., Stock O. 2004. Developing Affective Lexical Resources. In *PsychNology Journal*, 2(1).

Ressources

BalkaNet – http://www.ceid.upatras.gr/Balkanet/

eXtended WordNet – http://xwn.hlt.utdallas.edu

FrameNet – http://framenet.icsi.berkeley.edu/

Global WordNet – http://www.globalwordnet.org

Mappings entre versions de WordNet – http://www.cs.unt.edu/~rada/downloads.html#wordnet et http://www.lsi.upc.es/~nlp/tools/mapping.html

OpenCyc – http://www.opencyc.org/

ResearchCyc – http://research.cyc.com/

SemCor Corpus – http://www.cs.unt.edu/~rada/downloads.html

SentiWordNet – http://sentiwordnet.isti.cnr.it/

Simple Wikipedia – http://simple.wikipedia.org

SUMO – http://www.ontologyportal.org/ - http://ontology.teknowledge.com/

VerbNet – http://verbs.colorado.edu/~mpalmer/projects/verbnet/downloads.html

Wikipedia en anglais – http://en.wikipedia.org

WordNet – http://wordnet.princeton.edu

WordNet Domains & WordNet-Affects - http://wndomains.itc.it/download.html

WordNet::Similarity – http://www.d.umn.edu/~tpederse/similarity.html

GraalWeb ou accéder à une bibliothèque décentralisée de grammaires locales

Matthieu Constant
IGM - Université de Marne-la-Vallée
5, bd Descartes
Champs-sur-Marne
77454 Marne-la-Vallée Cedex
mconstan@univ-mlv.fr

Résumé

Cet article présente un système en-ligne de partage de grammaires locales de descriptions linguistiques : la bibliothèque Graal. Celle-ci a pour caractéristique d'être décentralisée : chaque auteur gère ses propres grammaires sur son propre serveur. L'interface GraalWeb permet de visualiser, explorer et télécharger les grammaires disponibles dans cette bibliothèque à partir d'un index précalculé, de manière transparente.

Mots-clefs :

diffusion, grammaires locales, recherche d'information, ressources linguistiques

1 Introduction

Le partage et la diffusion de ressources informatiques est en plein essor depuis une dizaine d'années avec la démocratisation d'Internet. La naissance de Linux et la création de la licence GPL[1] a également généré un élan extraordinaire de diffusion de ressources libres. Dans la communauté TAL, la diffusion de ressources linguistiques est, depuis peu, une composante valorisée du fait de la prise de conscience collective du réel besoin de telles ressources (Romary, 2000). Les corpus et les lexiques sont les ressources les plus usuellement diffusées et partagées. De telles entreprises n'existent pas ou très peu pour les grammaires. Cela s'explique peut-être en partie par le foisonnement des formalismes et des formats. Pourtant, depuis quelques années, avec la lexicalisation des grammaires comme (Abeillé, 2002), le développement de grammaires de plus en plus précises rend nécessaires des collaborations dans des cadres formels unifiés. Ainsi, on assiste à la création

[1] GNU Public License

de "réseaux" dédiés à de telles tâches[2].

Nous nous intéressons spécifiquement aux grammaires locales (Gross, 1993; Gross, 1997), formalisme de description linguistique partagé par la communauté RELEX formée d'une trentaine d'équipes. De ce fait, elles rentrent particulièrement bien dans le cadre d'un projet de diffusion.

Dans cet article, nous décrivons un système de bibliothèque décentralisée de telles grammaires, Graal. Ce système, ouvert à tous, a pour ambition de
– proposer un support simple à des chercheurs isolés pour diffuser leurs grammaires locales,
– faire office, à terme, d'état-de-l'art dans le domaine des grammaires locales,
– permettre une utilisation intensive des grammaires locales dans des applications du TAL au moyen d'outils d'importation.

Cette bibliothèque est accessible en-ligne de manière transparente au moyen de l'applet GraalWeb (`http://igm.univ-mlv.fr/~mconstan/library/`). Elle est pour l'instant limitée aux grammaires au format Unitex (Paumier, 2003).

Dans un premier temps (section 2), nous présenterons assez brièvement les grammaires locales. Nous décrirons ensuite de manière détaillée le système Graal (section 3). Nous développerons enfin quelques aspects pratiques avec la description d'un outil donnant accès à ce système : l'applet GraalWeb, une vue en-ligne de Graal (section 4).

2 Grammaires locales

Les formalismes de grammaires foisonnent en TAL et apparaissent à différents niveaux d'analyse allant de la morphologie à la syntaxe. Les modèles à états finis tels que les expressions régulières, les automates ou les transducteurs ont montré une efficacité certaine, notamment pour l'analyse morphologique et l'analyse lexicale (Mohri, 1997; Karttunen, 2001). L'analyse syntaxique nécessite des formalismes un peu plus évolués, même si plusieurs études ont montré l'intérêt des transducteurs pour cette opération (Roche, 1993; Abney, 1996). Utilisés historiquement, les grammaires algébriques puis les réseaux récursifs de transitions (Woods, 1970) ont montré des limites pratiques et ont évolués vers des grammaires algébriques décorées de contraintes d'unification telles que LFG (Bresnan & Kaplan, 1982). La famille des grammaires d'arbres adjoints (Joshi, 1987) est également très largement utilisée dans la communauté, ainsi que les grammaires de contraintes telles que HPSG (Pollard & Sag, 1994) qui prennent de plus en plus d'ampleur.

Entre analyse lexicale et analyse syntaxique, il existe un niveau intermédiaire aux limites floues formé d'un ensemble de phénomènes locaux figés ou semi-figés. Les modèles à états finis y ont été appliqués avec un certain succès (Maurel, 1990). Le formalisme des grammaires locales (Gross, 1993; Gross, 1999), une extension de ces modèles, est apparu comme une évolution très intéressante du fait de sa simplicité et sa modularité.

[2]On citera, par exemple, le projet LinGO (grammaires HPSG), le réseau RELEX (lexiques et grammaires) ; projet PAPILLON (lexiques multilingues) ; projet lexsynt (lexiques syntaxiques).

2.1 Représentation

Les grammaires locales sont équivalentes à des réseaux récursifs de transitions. Elles comportent deux alphabets disjoints : un alphabet de symboles terminaux et un alphabet de symboles non-terminaux. A chaque symbole non-terminal, est associé un automate sur les deux alphabets. Il existe un symbole non-terminal particulier jouant le rôle d'axiome, soit le point d'entrée de la grammaire. Ces grammaire reconnaissent théoriquement des langages algébriques. Elles ont également une singularité pratique : l'utilisation de masques lexicaux complexes comme étiquettes des automates. Les masques lexicaux définissent des sous-ensembles d'items lexicaux eux-même définis dans des lexiques. Par exemple, l'étiquette *<N+Conc :ms>* correspond à l'ensemble des noms concrets au masculin singulier. Cette singularité ne change rien au niveau formel car un masque lexical peut être remplacé par une disjonction d'items lexicaux. Chaque automate d'une telle grammaire est représenté sous la forme d'un graphe orienté dont les étiquettes sont sur les sommets. Les symboles non-terminaux sont des appels à d'autres automates.

Le formalisme des grammaires locales est partagé par un réseau informel d'une trentaine d'équipes de recherche en informatique linguistique. Les plate-formes Intex (Silberztein, 1993) et Unitex (Paumier, 2003) offrent un cadre unifié de travail (formalisme et formats utilisés).

2.2 Analyse de texte et applications

L'intérêt majeur des grammaires locales est de représenter de manière simple et compacte des contraintes lexico-syntaxiques définissant des classes syntaxiques comme les déterminants nominaux (Silberztein, 2003), les complexes verbaux (Gross, 1999) et même des classes syntaxico-sémantiques comme les adverbes de dates (Maurel, 1990), les prépositions locatives (Constant, 2003). À un moindre niveau, les grammaires locales sont aussi utilisées pour l'analyse locale de surface basée sur des contraintes grammaticales ou graphiques : ex. chunking (Blanc *et al.*, 2007), reconnaissance d'entités nommées (Friburger & Maurel, 2004), etc. Elles servent aussi à l'analyse de textes spécialisés comme les bulletins boursiers (Nakamura, 2005).

L'intégration de grammaires locales dans des processus industriels est de plus en plus courante comme le montre le projet Outilex (Blanc & Constant, 2006) financé par le Ministère français de l'Industrie. Ce projet rassemblant une dizaine de partenaires dont la moitié d'industriels est basé sur la technologie des grammaires locales.

3 Graal, un système décentralisé de catalogue de grammaires locales

Les grammaires locales se développent de manière anarchique dans la communauté et il est difficile d'avoir une vue précise de l'ensemble des grammaires locales existantes. Pour palier à ce problème, nous proposons un système de partage de telles ressources : la bibliothèque Graal[3] qui consiste en

[3]Graal signifie "Grammar and automata library".

un ensemble de serveurs HTTP de grammaires locales comme le montre la figure 1. Ces serveurs jouent l'unique rôle de "dépôt". Un utilisateur ou une application souhaitant avoir accès à leur contenu passent par un serveur d'accès. Ce serveur comporte un index à partir duquel toutes les requêtes sont traitées. L'architecture décentralisée est ainsi transparente pour l'utilisateur.

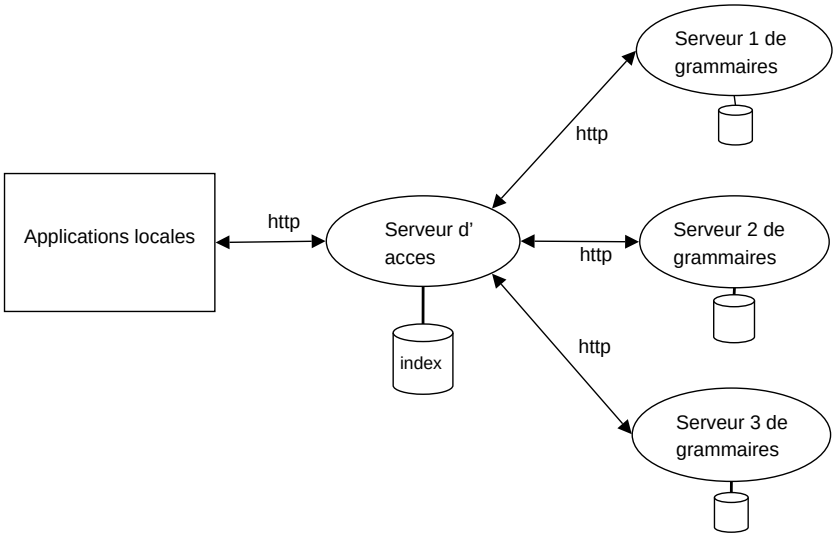


FIG. 1 – Architecture de Graal

3.1 Un ensemble de dépôts indépendants

Le système Graal est donc constitué d'un ensemble de dépôts (ou serveurs de grammaires locales) gérés indépendamment les uns des autres par leur propriétaires respectifs. Un dépôt est défini par une URL de base et un propriétaire. Par exemple, il peut être situé sur le propre site web d'un auteur de grammaires. Cet ensemble de dépôts est connu à l'avance par le système. Tout auteur souhaitant avoir son propre dépôt référencé doit donc en informer l'administrateur de Graal.

Chaque dépôt est composé d'un ensemble de paquetages de grammaires. Un paquetage de grammaires est une archive comprenant une collection d'automates (ou "graphes"), une licence et des documentations XML et HTML. La documentation n'est pas obligatoire mais elle est fortement conseillée car le document XML comprend des informations utiles telles que les auteurs, la langue, une description linguistique, les points d'entrées (les automates principaux), des exemples de séquences reconnues, etc. Notons qu'un automate d'un paquetage peut faire référence à des automates du même paquetage, mais aussi à des automates d'un autre paquetage pouvant être situé dans un autre dépôt[4]. Un paquetage est défini par un dépôt et un chemin relatif dans ce dépôt. Pour être connus du système, les chemins relatifs des paquetages d'un dépôt doivent être listés par son propriétaire dans un fichier défini à l'avance. Ainsi, un auteur est libre de rajouter ou supprimer des paquetages de Graal quand il ou elle le souhaite.

[4]Un système de référencement des automates a été mis en place à cet effet et a été rendu compatible avec Unitex.

3.2 Un serveur d'accès et un index

Cette architecture décentralisée est rendue transparente pour l'utilisateur grâce à un index situé sur le serveur d'accès. Cet index détient des informations précises sur la bibliothèque, notamment sur le contenu linguistique des différents dépôts ce qui permet aux utilisateurs de définir des requêtes spécifiques sur le contenu, lexical notamment (cf. section 4). Graal n'est donc pas seulement un répertoire organisé de liens vers des ressources, avec pour chacune d'elles des descriptions méta-linguistiques, comme la plupart des catalogues en-ligne. Dans son état actuel, l'index comprend les informations suivantes :

– la liste des paquets de grammaires (langue, dépôt, chemin),
– les termes utilisés dans certains champs-clés de la documentation des paquetages,
– les termes utilisés dans les automates,
– la dépendance entre les automates : quels automates appellent quels automates,
– les automates principaux (soit donnés dans la documentation ; soit calculés automatiquement par l'indexeur)

L'indexation des différents dépôts est lancée périodiquement (pour l'instant manuellement). Le référencement d'un paquetage n'est donc pas instantané après son placement dans un dépôt : il faut attendre la prochaine indexation. C'est d'ailleurs le cas avec les moteurs de recherche du type Google. Dans un futur proche, nous souhaitons mettre en place un système de veille afin de "réagir" au plus vite aux mises à jour de la bibliothèque. Par ailleurs, à chaque indexation, les paquetages sont téléchargés pour obtenir une sauvegarde de la bibliothèque.

Notre système ressemble un peu par son architecture à celui proposé dans (Romary, 2000). Cependant, la décentralisation de ce dernier est plus poussée : le serveur d'accès est limité à la gestion des flux de requêtes ; ce sont les différents serveurs de ressources (pour nous, grammaires locales) qui traitent les requêtes elles-mêmes. Nous n'avons pas souhaité reprendre ce système par souci de simplicité.

3.3 La question de la qualité

La décentralisation de notre système engendre un certain nombre de problèmes. En particulier, comment garantir la qualité du contenu de la bibliothèque ? Une trop grande liberté donnée aux auteurs ne risque-t-elle pas de conduire à la construction d'une bibliothèque au contenu linguistique médiocre ? En effet, un tel système offre moins de contrôle qu'un système centralisé. Par exemple, dans un système centralisé tel que celui du projet Papillon (Mangeot-Lerebours *et al.*, 2003) pour la construction de lexiques multilingues, l'analyse préalable des nouvelles entrées par un collège d'experts permet de garantir la qualité du lexique. Ce n'est pas le cas dans notre bibliothèque. Cependant, quelques solutions existent, comme :

– *un contrôle a priori :*
  Lors de la demande d'une personne d'ajouter son site à l'ensemble des dépôts, si le propriétaire n'apparaît pas fiable, l'administrateur est libre de ne pas l'ajouter.
– *un contrôle a posteriori :*
  Une commission d'évaluation de la bibliothèque pourrait être mise en place et rédigerait des recommandations pour chaque dépôt. En cas de non-respect des recommandations par le pro-

priétaire, son dépôt pourrait simplement être exclu de Graal.

4 GraalWeb, le "google des grammaires locales"

GraalWeb est une applet Java qui donne une vue en ligne de Graal au moyen d'un moteur de recherche et d'un explorateur. Elle permet aussi de télécharger les paquetages de grammaires disponibles dans la bibliothèque. Le moteur de recherche utilise des techniques classiques du domaine de la recherche d'informations. L'explorateur permet d'avoir une vue d'ensemble de la bibliothèque (l'ensemble des paquetages, les dépendances entre les grammaires) et une vue détaillée du contenu au moyen d'un visualisateur avancé d'automates.

Pour l'instant, les requêtes tournent en local, c'est-à-dire que l'index est chargé au niveau du client à chaque chargement de l'applet, ce qui nécessite un index de petite taille (i.e. avec un nombre limité d'informations). Dans l'avenir, avec l'augmentation de la taille de Graal, il sera nécessaire de faire évoluer la bibliothèque vers un système où les requêtes sont traitées sur le serveur d'accès (et non en local). Une telle évolution a été prévue dans l'implantation actuelle de GraalWeb.

4.1 Un moteur de recherche

GraalWeb comporte un moteur de recherche permettant de trouver les grammaires utilisant un certain nombre de termes soit dans leur lexique, soit dans leur documentation. En entrée, chaque requête est définie par un ensemble de mots-clés. Son traitement produit une liste d'automates triés selon leur degré de pertinence par rapport à la requête. Cet outil est basé sur des techniques de recherche d'informations classiques avec une représentation des documents et des requêtes au moyen de vecteurs de termes, et des mesures de similarité entre ces vecteurs (Baeza-Yates & Ribeiro-Neto, 1999). Dans l'état actuel de nos travaux, les vecteurs utilisés sont des vecteurs binaires de termes : l'élément associé à un terme est égal à 1 si ce terme apparaît dans le document ; il est égal à 0 dans le cas contraire.

Nous avons développé trois techniques pour la recherche de grammaires selon leur contenu lexical. La première technique consiste à considérer que les termes d'un automate sont ses symboles terminaux. Le degré de similarité entre un automate et une requête est le cosinus de l'angle de leurs vecteurs respectifs. La deuxième technique (indépendante de la requête) consiste à ne tenir compte que de la dépendance entre les différents automates de la bibliothèque et donner un degré d'importance à chaque automate en utilisant la technique du PageRank de Google (Page *et al.*, 1998) : plus un automate est appelé par des automates importants, plus il est important. Nous appelons ce calcul *GrammarRank* en hommage à son illustre inspirateur. La troisième technique consiste à combiner lexique et dépendance. Elle est basée sur le fait qu'un terme utilisé dans un automate est aussi utilisé indirectement par un automate qui l'appelle. Notre algorithme consiste simplement à propager les termes dans le graphe de dépendance inverse de la bibliothèque. Des expériences récentes relativement similaires ont été testées avec succès pour la recherche documentaire sur le Web comme dans (Qin *et al.*, 2005). Le score final d'un automate combine ses trois techniques auxquelles on affecte des coefficients : 0.1 pour la première technique, 0.05 pour le GrammarRank,

0.85 pour la technique hybride.

Lorsque l'on effectue une recherche sur la documentation des paquetages, on utilise une variante de la première technique. Si un paquetage est jugé pertinent, seuls les automates principaux sont listés.

4.2 Un explorateur de grammaires locales

GraalWeb comporte aussi un explorateur de grammaires locales permettant d'avoir une vue à la fois globale et détaillée du système Graal. Tout d'abord, il est possible d'avoir la liste des paquetages disponibles et, pour chacun d'eux, voir la documentation HTML associée (si disponible) pour avoir une idée de son contenu linguistique. La structure de dépendance entre les automates d'un paquetage est également visualisable sous la forme d'un arbre (qui se déploie au fur et à mesure de son exploration) comme pour les explorateurs de systèmes de fichiers. Les noeuds enfants de la racine de l'arbre sont les graphes principaux du paquetage. Chaque automate est également visualisable à l'aide d'un visualiseur de graphes implémenté à partir du code source d'Unitex. Il permet de voir chaque automate de manière détaillée. Un appel à un automate est considéré comme un lien hypertexte que l'on peut suivre pour le visualiser. Notre explorateur permet aussi de suivre la dépendance inverse de la bibliothèque, c'est-à-dire avoir la liste des automates parents de l'automate courant par un simple clic droit de souris, et de les visualiser en les sélectionnant.

5 Conclusion et perspectives

La bibliothèque Graal a l'ambition d'être un système de partage de grammaires locales dans la communauté TAL. Elle a la particularité d'être décentralisée ; plus précisément, chaque auteur dispose ses grammaires sur son propre site. Un indexeur se charge de centraliser les informations sur un serveur d'accès. L'applet GraalWeb permet aux utilisateurs d'avoir une vue en-ligne de cette bibliothèque de manière transparente.

La bibliothèque est pour l'instant limitée aux grammaires Unitex, mais nous projetons de l'étendre à d'autres formats. Par ailleurs, nous souhaitons améliorer nos fonctionnalités de recherche d'informations au moyen de techniques plus évoluées, comme, par exemple, la recherche des grammaires qui incluent une séquence donnée de mots (Constant, 2003). Enfin, il sera prochainement possible de réaliser une projection de tout ou d'une partie de Graal sur un système local intégrant des modules de traitements de textes alimentés par des grammaires locales.

À l'heure actuelle, il existe huit paquetages de grammaires tous fournis par des chercheurs de l'Institut Gaspard Monge. L'ensemble contient environ 1 700 automates de descriptions linguistiques. Nous espérons que ce petit ensemble de grammaires servira de pompe d'amorçage, en incitant les chercheurs du domaine à partager leurs grammaires au moyen de notre système et ainsi permettre de nouvelles avancées significatives dans le domaine.

Remerciements

Cette recherche a été en partie financée par le CNRS et le projet Outilex du Ministère de l'Industrie.

Références

ABEILLÉ A. (2002). *Une grammaire électronique du français*. Paris : CNRS Editions.

ABNEY S. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, **2**(4), 337–344.

BAEZA-YATES R. & RIBEIRO-NETO B. (1999). *Modern Information Retrival*. Addison-Wesley.

BLANC O. & CONSTANT M. (2006). Outilex, a linguistic platform for text processing. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, p. 73–76.

BLANC O., CONSTANT M. & WATRIN P. (2007). Segmentation en super-chunks. In *Actes de TALN 2007 (Traitement automatique des langues naturelles)*, Toulouse : ATALA.

BRESNAN J. & KAPLAN R. (1982). *Lexical-functional grammar : A formal system for grammatical representation*, In J. BRESNAN, Ed., *The Mental Representation of Grammatical Relations*, p. 173–281. The MIT Press : Cambridge, Mass.

CONSTANT M. (2003). *Grammaires locales pour l'analyse automatique de textes : Méthodes de construction et outils de gestion*. PhD thesis, Université de Marne la Vallée.

FRIBURGER N. & MAUREL D. (2004). Finite-state transducer cascades to extract named entities in texts. *Theoretical Computer Science*, **313**, 94–104.

GROSS M. (1993). Local grammars and their representation by finite automata. In M. HOEY, Ed., *Data, Description, Discourse, Papers on the English Language in honour of John McH Sinclair*, p. 26–38. Berlin/New York : Springer Verlag.

GROSS M. (1997). *The Construction of Local Grammars*, In E. ROCHE & Y. SCHABES, Eds., *Finite State Language Processing*, p. 329–352. The MIT Press : Cambridge, Mass.

GROSS M. (1999). Lemmatization of compound tense in english. *Linguisticae Investigationes*, **22**.

JOSHI A. K. (1987). *An introduction to tree adjoining grammars*, In MANASTER-RAMER, Ed., *Mathematics of Language*, p. 329–352. John Benjamins : Amsterdam.

KARTTUNEN L. (2001). Applications of finite-state transducers in natural language processing. In S. YU & A. PAUN, Eds., *Implementation and Application of Automata*, volume 2088 of *Lecture Notes in Computer Science*, p. 34–46. Heidelberg : Springer Verlag.

MANGEOT-LEREBOURS M., SÉRASSET G. & LAFOURCADE M. (2003). Construction collaborative d'une base lexicale multilingue - le projet papillon. *Traitement Automatique des Langues (TAL)*, **44**(2), 151 – 176.

MAUREL D. (1990). Description par automate des dates et des adverbes apparentés. *Mathématiques et Sciences Humaines*, **109**, 5–16.

MOHRI M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, **23**(2), 269–312.

NAKAMURA T. (2005). *Analysing texts in a specific domain with local grammars : The case of stock exchange market reports*, In Y. KAWAGUCHI, S. ZAIMA, T. TAKAGAKI, K. SHIBANO & M. USAMI, Eds., *Linguistic Informatics - State of the Art and the Future*, p. 76–98. Benjamins : Tokyo University of Foreign Studies, Amsterdam/Philadelphia.

PAGE L., BRIN S., MOTWANI R. & WINOGRAD T. (1998). *The PageRank Citation Ranking : Bringing Order to the Web*. Stanford Digital Technologies.

PAUMIER S. (2003). *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. PhD thesis, Université de Marne la Vallée.

POLLARD C. & SAG I. A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago : CSLI Publications.

QIN T., LIU T.-Y., ZHANG X.-D., CHEN Z. & MA W.-Y. (2005). A study of relevance propagation for web search. In *The 28th Annual International ACM SIGIR Conference*, New York, NY : ACM Press.

ROCHE E. (1993). *Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire*. PhD thesis, Université Paris 7.

ROMARY L. (2000). *Outils d'accès à des ressources linguistiques*, In J.-M. PIERREL, Ed., *Ingénierie des langues*. Série Informatique et systèmes d'information. Hermès Science : Paris.

SILBERZTEIN M. (2003). Finite-state description of the french determiner system. *Journal of French Language Studies*, **13**(2).

SILBERZTEIN M. D. (1993). *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*. paris : Masson.

WOODS W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, **13**(10), 591–606.