Extracting subcategorisation information
from Maurice Gross' grammar lexicon

Claire Gardent[♢], **Bruno Guillaume**[♠], **Guy Perrier**[♡], **Ingrid Falk**[♣]

[♢]CNRS/LORIA
[♠]INRIA/LORIA
[♡]University Nancy 2/LORIA
[♣]CNRS/ATILF
Nancy, France
FirstName.LastName@loria.fr

Abstract

Maurice Gross' grammar lexicon contains rich and exhaustive information about the morphosyntactic and semantic properties of French syntactic functors (verbs, adjectives, nouns). Yet its use within natural language processing systems is hampered both by its non standard encoding and by a structure that is partly implicit and partly underspecified. In this paper, we present a method for translating this information into a format more amenable for use by NLP systems, we discuss the results obtained so far, we compare our approach with related work and we identify the possible further uses that can be made of the reformatted information.

1. Introduction

As has been shown repeatedly, detailed subcategorisation information (that is, information about the complements of natural language predicative items such as verbs, deverbal nouns and predicative adjectives) is an essential component in enhancing the linguistic coverage and the accuracy of NLP systems. Thus for instance, (Briscoe and Carroll, 1993) shows that half of parse failures on unseen data test results from inaccurate subcategorisation information in the ANLT dictionary; (Carroll and Fang, 2004) demonstrates that for a given domain, using an HPSG (Head Driven Phrase Structure Grammar) enriched with detailed subcategorisation information improves the parse success rate by 15%; and (Han et al., 2000) show that good subcategorisation information is a key factor in achieving good quality machine translation.

For French, there exists to date no reference lexicon[1] that would contain detailed extensive subcategorisation information i.e., that would associate with each predicative item the set of its subcategorisation frames.

However Maurice Gross' grammar lexicon provides a systematic description of the syntactic properties of the syntactic functors of French namely, verbs, predicative nouns and adjectives. As (Hathout and Namer, 1998; Gardent et al., 2005) pointed out, the subcategorisation information contained in this lexicon is both detailed and extensive.

It is detailed in that each predicate usage is associated with a description of its possible subcategorisation frames while each frame is associated with a fine-grained description of the morphosyntactic properties of its arguments, of their optionality and of their grammatical properties e.g., with respect to cliticisation (see (Gardent et al., 2005) for more details).

It is extensive in that it covers verbs, adjectives and nouns. Thus roughly 5 000 verbs have been described over a total of 10 716 verb usages (Gross, 1975; Boons et al., 1976a; Boons et al., 1976b). Further, 25 000 verbal locutions are also described as well as 20 000 locutions using "être" (to be) or "avoir" (to have) (Gross, 1989).

The grammar lexicon has been digitised by the Laboratoire d'Automatique Documentaire et Linguistique (LADL) and is now partially available under an LGPL-LR licence[2].

In short, the grammar lexicon and its digitised analogue, the LADL tables, provide an interesting basis for building a subcategorisation lexicon for French. In this paper, we first argue that in order to be usable by NLP systems, the information contained in the grammar lexicon need to be pre-processed. We then propose a method for doing so and describe the data structures and algorithms used by this method. Finally, we present some first results and discuss perspectives for further research.

2. The ladl tables

2.1. Content

The LADL tables consist of a set of tables where each table groups together (usages of) predicative items that share some definitional properties. In particular, all predicative items in a given table share one (sometimes two) basic subcategorisation frames. For each predicative item

---

We would like to thank Eric Laporte and the Institut d'électronique et d'informatique Gaspard-Monge for making some of the LADL tables available to us in eletronic format. We also thank the Contrat Plan Etat Région : Ingénierie des Langues, du Document et de l'Information Scientifique, Technique et Culturelle for partially funding the research presented in this paper.

[1] Lionel Clement is currently working on a version of LEFFF that includes detailed subcategorisation information. This updated version is not yet publicly available however.

[2] cf. `http://infolingu.univ-mlv.fr/DonneesLinguistiques/Lexiques-Grammaires/Presentation.html`

present in the table, the columns of the table further specify the subcategorisation properties of that item. Typically, the table columns will provide information of the following type :

- detailed information about the verb and about the possible realisations of its arguments (e.g., whether a given argument can be realised as a noun, as an infinitival or as a finite sentence; if an argument can be realised as a nominal, whether it may includes a preposition and of what type; etc.)
- information reflecting syntactic properties of the verb or of the arguments (e.g., whether an argument can be cliticised)
- information about alternative subcategorisation frames
- information about semi-regular redistributions true or not of a specific subcategorisation frame (e.g., whether a transitive syntactic frame admits the passive)

Figure 2.1. shows the first two rows of table 8, a table which describes verbs taking a sentential or nominal complement introduced by the preposition "de" (e.g., *Jean se repent de sa conduite/Jean regrets his behaviour*). The first 6 columns describe properties of the subject; columns 6 and 7 describe properties of the verb; columns 8, 9, 10, 29 and 30 describe alternative subcategorisation frames; column 28 describe the possibility of having an impersonal passive; and columns 11 to 27 describe the possible realisations and syntactic properties of the *de*-argument.

2.2. Structure

As should be clear from the previous section, the columns of a LADL tables can describe different types of information. To automatically compute the subcategorisation frames of a given item in a given table, it is therefore necessary to know which column provides which type of information. But this is not sufficient. Indeed the columns of the table are also structured along three dimensions namely, disjunction, conjunction and dependency. Thus for instance in table 8:

- columns 13 and 14 are *dependent* on columns 11 and 12 (the possibility of an infinitival or interrogative sentential complement is dependent on the possibility of having a sentential complement in indicative or subjunctive mode)
- colums 16 and 17 represent *disjunctive* information on atomic feature values (the infinitival complement is compatible with either a future marking adverbial, a past marking adverbial, both or none).
- column 2 encodes *disjunctive* information on argument realisation (the subject is "non restreint/unrestricted" that is, it can be either a non human nominal, an infinitival or a finite sentence).
- columns 6 and 7 encode *conjunctive* information about the verb (the verb is associated with a given lemma and possibly with the reflexive particle "se" or "s"')

3. From the LADL tables to an NLP subcategorisation lexicon

Although the LADL tables are rich in content, their current format makes them difficult to use in NLP application. The reasons for this are threefold.

First, the format itself is non standard. In NLP applications, subcategorisation information is standardly gathered within a syntactic lexicon which associates with each predicative item the set of its possible subcategorisation frames. Further, subcategorisation frames are usually represented by a set of feature structures where each element in the set encode the linguistic properties either of the verb or one of the argument occuring in the frame being described. To be easily usable by NLP applications, it is highly desirable that such a syntactic lexicon be derived from the LADL tables.

Second, the structure of the table is either implicit in the headings or altogether absent. For instance in Table 8, the dependency between columns 16 to 21 and column 15 is not marked in any way while the atomic disjunction described by columns 18 to 20 need to be automatically recovered from the fact that the columns are adjacent and share the same feature (Vc) in their heading.

Third, the headings are non standard and need to be translated in feature structure specification that are more in line with current practice in syntactic annotation.

In what follows, we present a method for extracting from the LADL tables, an NLP oriented syntactic lexicon. In essence this method aims at making the table structure explicit and at translating the headings into standard practice feature structure notation. Specifically, it consists in the following three steps:

1. For each table, a **SynLex-graph** is (manually) produced which represents our interpretation of the table. This graph makes the table structure explicit and translates the headings into path equations.
2. A **graph traversal algorithm** is specified such that, given a SynLex-graph and a table, it produces for each entry in that table the set of subcategorisation frames associated by the table with that entry. The resulting lexicon is called a **LADL-lexicon** and closely reflects the content of the LADL table. Some of the information it contains is not currently used by most NLP tools in particular, parsers and surface realisers.
3. A **simplification algorithm** is specified such that given a LADL-lexicon, it produces an **NLP-lexicon**. The NLP lexicon is a simplified version of the LADL-lexicon where only features relevant for parsing/generating are preserved and which only partially reflects the content of the LAD-table. It is with this lexicon that NLP is expected to proceed.

We now describe each of these components in turn.

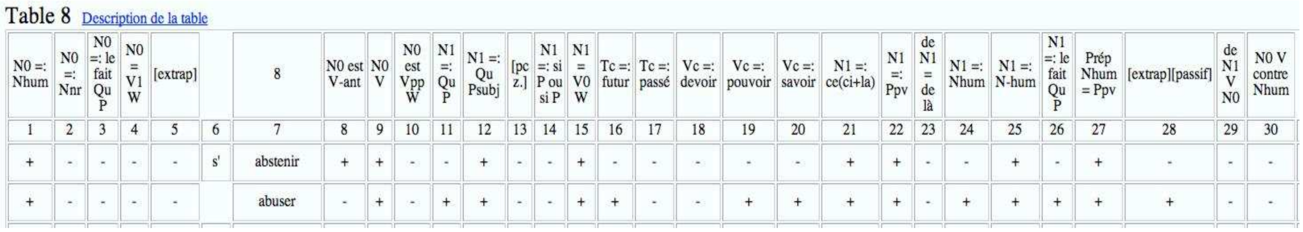

Table 8 Description de la table

N0 =: Nhum	N0 =: Nnr	N0 =: le fait Qu P	N0 = V1 W	[extrap]		8	N0 est V-ant	N0 V	N0 est Vpp W	N1 =: Qu P	N1 =: Qu Psubj	[pc z.]	N1 =: si P ou si P	N1 = V0 W	Tc =: futur	Tc =: passé	Vc =: devoir	Vc =: pouvoir	Vc =: savoir	N1 =: ce(ci+la)	N1 =: Ppv	de N1 = de là	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	Prép Nhum = Ppv	[extrap][passif]	de N1 V N0	N0 V contre Nhum
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
+	-	-	-	-	s'	abstenir	+	+	-	-	+	-	-	+	-	-	-	-	-	+	+	-	-	+	-	+	-	-	-
+	-	-	-	-		abuser	-	+	-	+	+	-	-	+	+	-	-	+	+	+	+	-	+	+	+	+	+	-	-

3.1. The SynLex graphs

The intuition. The SynLex graph makes the table structure explicit and translates the table headings into path equations. Fig. 1 gives an example subgraph of the graph associated with table 8.

To make the tables structure explicit, we use an and/or graph so that and-nodes can be used to represent conjunction of information, or-nodes to represent disjunction of information and node-accessibility to represent dependency between informational units.

The graph nodes are labelled with path equations which map the column headings into feature structures.

In a SynLex graph, the different types of linguistic information are then encoded both by the content and the structure of the graph. Thus:

- Different realisations of the same argument is encoded as a disjunction on the non terminal node representing this argument (e.g., "a0" in Fig. 1).

- Atomic disjunctions on feature values are encoded as a disjunction of leaf nodes with identical feature (e.g., node [2] and [3] in Fig. 1).

- Alternative subcategorisation frames are encoded as a disjonction of root nodes (e.g., nodes [9] "NO V" and True "Base Frame" in Fig. 1).

- Redistribution information linked to specific subcategorisation frames is encoded as an "lf" named, direct dependent of the relevant frame node

The syntax. A SynLex graph is an and/or acyclic graph containing three kinds of nodes namely, *or-nodes, and-nodes* and *frame-nodes*.

Or-nodes indicate an alternative. They contain no information and are represented by ellipses.

And-nodes indicate conjunctive information. They are represented by rectangle boxes divided in two parts: a top part containing a condition and a bottom part containing any number (including zero) of path equations. A condition can be:

- `[`c`]` which is true if the column number c is `+` or is not empty (in case of column for lemma, preposition or particle) or

- `[!`c`]` which is true if the column number c is `-`

A path equation is written *arg.feat* = *value* where *arg* can be `v`, `a0`, `a1`, `a2` or `lf`. When processing the graph, the feature value pair *feat* = *value* is added to the feature structure denoting the value of *arg*. A feature value *value* is either a disjunction of atomic values or the symbol $\$c$ where for the line l and the column c in the table, $\$c$ denotes the value of the cell $[l, c]$ in the table.

Finally, **frame**-nodes describe the different syntactical frames that are associated with a table. They are represented by shaded rectangle boxes with no in-edges. Like and-nodes, frame-nodes have a top- and a bottom-part. However, in frame-nodes the condition can be the constant `True` (this is the case in particular for the base syntactical frames which are common to all verbs of the table). Furthermore, the bottom part contains, instead of a set of feature equations, a set of features that describes the skeleton feature structure needed for the initialization of the algorithm.

In graph drawings, everything that is between quote ("like this") are comments (and are ignored by the algorithm).

3.2. Processing the SynLex-graphs

The basic idea is to compute for the graph associated with a table a reduced graph for each line of the table. Then, each path in the reduced graph gives an entry in the output LADL lexicon.

3.2.1. Reduced graph

Given a table and a line l, we compute the reduced graph from the full graph as follows:

- for each **and**-node or **frame**-node where the condition is false, remove the node and its in-edges and out-edges;

- for each **or**-node without out-edge, remove the node and its in-edges;

- replace the symbol $\$c$ by the content of the cell $[l, c]$.

In the remaining nodes, the conditions are necessarily true and are discarded (together with their comments).

The coherence between the column in the tables are reflected by the fact that every reduced graph obtained is connected. Thus if some column depends on some other (like columne 3 depends on column 2 in table 8) then in the corresponding full graph, each **and**-node controlled by condition `[3]` appears in a subgraph of some **and**-node controlled by condition `[2]`. An incoherent line in table 8 (e.g., column 2 with - and column 3 with +) would give a disconnected reduced graph.

When processing a table, if such disconnected reduced graph happens, this means either that the full graph is not correct for the table considered or that there is an error in the table.

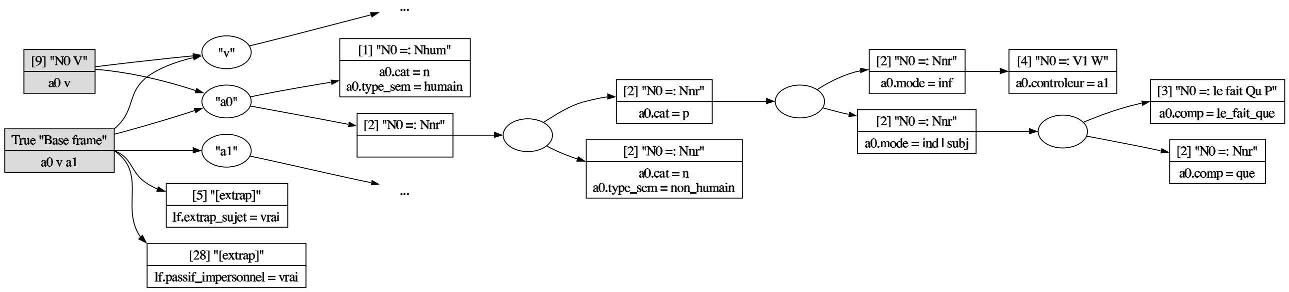

Figure 1: A (simplified) sub-graph of the graph specified for Table 8

Fig. 2 and 3 show the reduced graphs for the verbs *abuser* and *bénéficier* in Table 8. For readibility reason, theses graphs only partially reflect the content the LADL entries for these two verbs namely that shown in the following table.

	1	2	3	4	5	9	28
	N0=: Nhum	N0=: Nnr	N0=: le fait Qu P	N0=: V1W	[extrap]	N0 V	[extrap] [passif]
abuser	+	-	-	-	-	+	+
bénéficier	-	+	+	-	+	-	+

3.2.2. Computing the LADL-lexicon

The procedure for computing the LADL-lexicon consists in collecting for each path through the reduced graph, the path equations associated with this path. More specifically:

- for each **frame**-node F, a skeleton feature structure of the form [(f$_i$ = nil)$^{+}$, LF = nil] is created with nil the empty feature structure, f$_i$ the features given by the bottom part of F and LF, the feature describing the lexical features of that frame (passivisation, reflexivisation, etc.)

- the subgraph rooted in F is processed where for each possible switching in that subgraph (a switching consists in a choice for each **or**-node of one of its out-edges), all path equations in the remaining **and**-nodes are added to the skeleton feature structure.

This procedure applied to our two examples verbs gives the following lexicon entries. To be precise, in the complete lexicon, these entries should be duplicated for each possible feature structures of arguments `v` and `a1`.

```
--abuser--
a0[cat=n, type_sem=humain] v[..]
      lf[passif_impersonnel=vrai]
a0[cat=n, type_sem=humain] v[..] a1[..]
      lf[passif_impersonnel=vrai]

--bénéficier--
a0[cat=n, type_sem=non_humain] v[..] a1[..]
      lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=inf] v[..] a1[..]
      lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=le_fait_que] v[..] a1[..]
      lf[extrap_sujet=vrai, passif_impersonnel=vrai]
a0[cat=p, mode=ind|subj, comp=que] v[..] a1[..]
      lf[extrap_sujet=vrai, passif_impersonnel=vrai]
```

3.2.3. Computing the NLP-lexicon

Some of the information contained in the LADL tables is very fine-grained and not currently used by parsers and generators. For instance, in table 8, columns 16 to 20 express constraints on the temporal and verbal properties of the infinitival complement. Columns 16 and 17 indicates whether the infinitival can be combined with a futur and a past tense adverbial respectively while columns 18 to 20 indicates whether the infinitival verb can be *devoir, pouvoir* or *savoir*.

To obtain a syntactic lexicon that is more in line with what is commonly used in NLP, we produce from the LADL lexicon a simpler syntactic lexicon (the NLP lexicon). This lexicon contains only a subset of the features used by the LADL lexicon. This lexicon is produced by first specifying the set of features $\mathcal{F}_{nlp}$ that will occur in the NLP lexicon. This set is a subset of the set of features used to produce the LADL lexicon ($\mathcal{F}_{nlp} \subset \mathcal{F}_{ladl}$). Next, the NLP lexicon is created from the LADL lexicon by copying over for each entry in the lexicon, only these path equations whose features are in $\mathcal{F}_{nlp}$.

4. Related work

In (Hathout and Namer, 1997; Hathout and Namer, 1998), an earlier proposal is presented which shows how to extract three types of syntactic lexicons from the LADL table namely, an intermediate theory neutral lexicon, a Tree Adjoining Grammar lexicon and a Head-Driven Phrase Structure Grammar lexicon.

The main difference with our approach is that this earlier proposal does not work from an explicit representation of the table structure and content such as the SynLex graphs. Instead, the implicit structure of the table is made explicit in the table file itself and the table headings are semi-automatically translated into path equations. Given the complexity of the LADL tables and the difficulty there is in evaluating the LADL derived lexicons, this is a non trivial difference. Indeed, the SynLex graphs provide a declarative specification of the table content and structure which makes checking the correctness of the table interpretation easier and which facilitates debugging (if a subcategorisation frame in the generated lexicon seems ill-formed it is easy to trace back the source of the error within the SynLex graph). Because it admits of a procedural interpretation whose effect is to produce the subcategorisation lexicon, the SynLex graph further supports rapid and consistent modifications of the generated lexi-

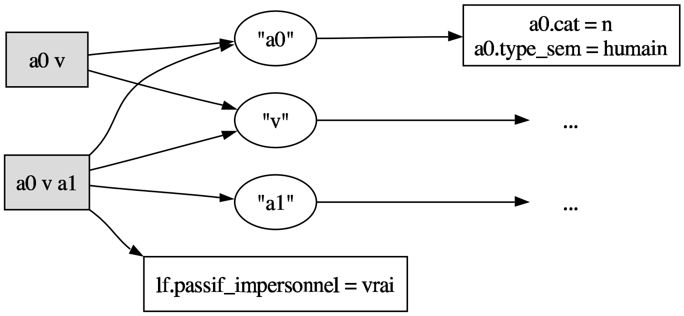

Figure 2: reduced graph for the verb *abuser*

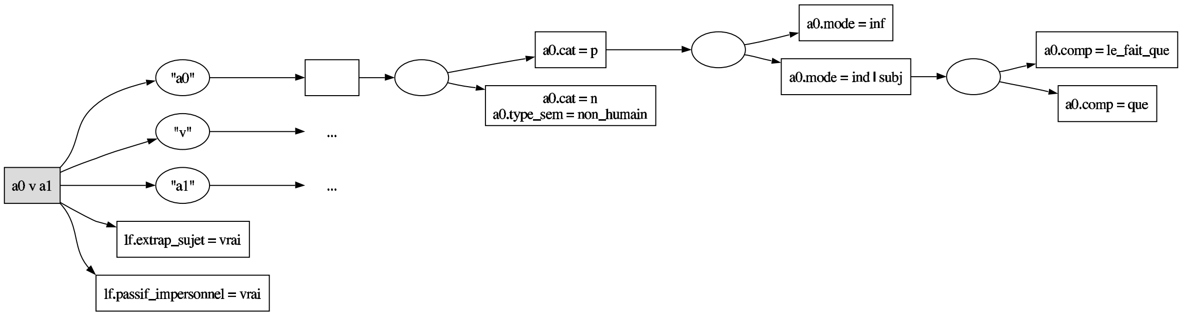

Figure 3: reduced graph for the verb *bénéficier*

cons. We used this feature for instance, to add information about grammatical functions and thematic role to the generated lexicon so that in each subcategorisation frame and for each argument occurring in that frame, its grammatical function and its thematical role is automatically generated from the upgraded SynLex graph. More generally, the SynLex graph allow us to *tailor* (add or remove) the information that is given by the LADL tables to the requirements placed on the extracted lexicon.

In sum, by providing this intermediate representation, the present approach makes it easier to understand the content of the tables, to verify the correctness of the proposed interpretation and to debug and modify the generated lexicons.

Another difference concerns the processing of the table headings. While in our approach the headings are translated manually into a set of path equations, in (Hathout and Namer, 1997; Hathout and Namer, 1998) these are translated semi-automatically. Although a semi-automatic translation is appealing, we chose not to use one for two reasons. First, certain headings translate in our approch to subgraphs rather than path equations. This is the case for instance for “NO = Nnr” which indicates a unrestricted subject and which translates in our approach to a disjunctive subgraph representing the subject either as a non human noun phrase, an infinitival or a finite sentence. Second, certain headings seem to be context sensitive in that they do not vehiculate the same information from table to table. For instance, the heading “N1 = QuPsubj” indicates a sentential complement in subjonctive mode in table 10 whilst in table 8, the same heading need to be interpreted as indicating a sentential complement in subjunctive mode and introduced by the preposition “de” (because by default, N1 in this table is introduced by the preposition “de”).

Finally the two approaches differ in their coverage and the availability of the generated lexicon. While (Hathout and Namer, 1997; Hathout and Namer, 1998) present results for four tables, we have already processed 12 and intend to process the remaining available tables. Our results are furthermore available on the web.

5. Results and perspectives

The three step method described above has been applied to the available subset of the first set of tables compiled by Maurice Gross which describes what Harris called *operators* namely, verbs that take a (finite or infinitival) sentential complement. For these verbs, Maurice Gross specified 19 tables covering roughly 2 500 verbs. Through the LGPL-LR licence, we obtained 12 of these tables thereby covering 1 936 verbs and 2 019 verb usages.

For each of these tables, we produced a SynLex-graphs and computed the corresponding LADL- and NLP-lexicons. These are now available at `http://www.loria.fr/~gardent/ladl/content/resultats.php`.

In sum, we have specified a procedure for extracting a subcategorisation lexicon from a LADL table and developed the tools to implement it. We have furthermore applied this procedure to 12 of the available tables.

Much remains to be done however.

First, the remaining tables need to be processed. As the tables processed so far all belong to the same subgroup (the group of tables directly supervised by Maurice Gross), we expect tables from other subgroups to exhibit differences either in content or in structure – it will be interesting to see whether the approach proposed straightforwardly extend to these or whether changes are required.

Second, the data need to be "normalised". To be widely usable, a resource must conform to general linguistic and computational usage. Linguistically, feature names and categories should be used which "make sense" to the widest possible audience. To this end, we intend to make use of the catalogues proposed by MulText, EAGLES and more recently by the Lexical Markup Framework ISO (TC37/SC4) standard. The latter in particular, provides a high level model for representing data in lexical resources and thus guarantees a maximum of interoperability with multilingual computer applications. Computationally, it is important to use a language which supports efficient and generalised processing. XML is in this respect a natural candidate as it is a *de facto* standard supporting information structuring, structure checking and querying.

Third, the quality of the results need to be evaluated. Typically there is a large number of subcategorisation frames associated with each entry in the lexicon and some of them looks rather implausible. To weed out the dictionnary, several complementary methods should be used in combination.

A parser can be used on large corpora to detect cases where parsing either fails (undergeneration; a subcategorisation frame might me missing) or on the opposite, where ambiguity is extremely high (overgeneration: there might be to many subcategorisation frames assigned to the verb involved).

Lexicon comparison is another possibility where the NLP lexicon could be compared with other existing syntactic lexicons such as LeFFF (Clément et al., 2004). In a first step, both lexicon must be reduced to a maximum common denominator. Next, their respective coverage and subcategorisation information can be compared. Cases of complete overlap (same set of subcategorisation frames for a given item) suggest that the information is correct, fully disjunctive cases might indicate either missing or conflicting information; partial overlaps will require more sophisticated treatment (either manual or statistic based).

A third possibility is to generate with the LADL lexicon and compare the output with the results obtained from the lexicon fusion and the parsing process. For instance, if the frame used by a generated sentence is marked as revealing either missing or conflicting information, the generated sentence might give complementary information as to the validity of the frame (if the sentence is acceptable then the frame is valid).

Another direction for future research concerns the creation of a more structured lexicon. As (Levin, 1993) shows, verbs can be organised in syntax based classes thereby revealing semantic commonalities among verbs of a common class. Maurice Gross's grammar lexicon evidently points in the same direction as it groups togethers verb usages with similar syntactic properties. We are currently using the subcategorisation frames defined by our approach to create for each LADL table a set of verb classes which reflects the finer grained syntactic information contained by the table. In the longer run, the expectation is that a VerbNet for French can be created similar to that developed for English by (Kipper et al., 2000).

6. References

Boons, J.-P., A. Guillet, and C. Leclère, 1976a. *La structure des phrases simples en français. I : Constructions intransitives*. Droz, Gen ève.

Boons, J.-P., A. Guillet, and C. Leclère, 1976b. La structure des phrases simples en français. ii : Classes de constructions transitives. Technical report, Univ. Paris 7.

Briscoe, E. and J. Carroll, 1993. Generalised probabilistic lr parsing for unification-based grammars. *Computational Linguistics*.

Carroll, J. and A. Fang, 2004. The automatic acquisition of verb subcategorisations and their impact on the performance of an hpsg parser. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP)*. Sanya City, China.

Clément, L., B. Sagot, and B. Lang, 2004. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC'04*. Lisbonne.

Gardent, C., B. Guillaume, G. Perrier, and I. Falk, 2005. Maurice gross' grammar lexicon and natural language processing. In *Proceedings of the 2nd Language and Technology Conference*. Poznan, Poland.

Gross, G., 1989. *Les constructions converses du français*. Droz, Gen ève.

Gross, M., 1975. *Méthodes en syntaxe*. Hermann.

Han, C., B. Lavoie, M. Palmer, O. Rambow, R. Kittredge, T. Korelsky, and N. Kim, 2000. Handling structural divergences and recovering dropped arguments in a korean/english machine translation system. In *Proceedings of the Association for Machine Translation in the Americas*. Berlin/New York: Springer Verlag.

Hathout, N. and F. Namer, 1997. Génération (semi)-automatique de ressources lexicales réutilisables à grande échelle. In *Actes des 1ères JST FRANCIL, AUPELF-UREF*. Avignon.

Hathout, N. and F. Namer, 1998. Automatic construction and validation of french large lexical resources: Reuse of verb theoretical linguistic descriptions. In *First International Conference on Language Resources and Evaluation, Granada, Spain*.

Kipper, K., H. Trang Dang, and M. Palmer, 2000. Class based construction of a verb lexicon. In *Proceedings of AAAI-2000 Seventeenth National Conference on Artificial Intelligence*. Austin TX.

Levin, B., 1993. *English verb classes and alternations: a preliminary investigation*. Chicago University Press.