Automatic Construction and Validation of French Large Lexical Resources: Reuse of Verb Theoretical Linguistic Descriptions

Nabil Hathout* **& Fiammetta Namer***

*CNRS-INaLF
Chateau du Montet
Rue du Doyen Roubault
F-54500 Vandœuvre-lès-Nancy
hathout@inalf.cnrs.fr

*Université Nancy 2
UFR Sciences du langage
BP. 3397
F-54015 Nancy cedex
namer@clsh.u-nancy.fr

Abstract

We address in this paper some problems related to the reuse for NLP of LADL's Lexicon-Grammar (LG). This major source of French verbs lexical knowledge has been publicly available on the Internet for several years. However, it has not been used by the NLP community, mainly because of its format: ASCII files each of them containing a table with binary values (+/−). The interpretation of these tables is non trivial because large parts of the linguistic informations they contain are neither explicit nor represented in a uniform manner. The paper presents 3 aspects of the research: (1) The translation of LG into a PATR-II Intermediate Lexicon (IL). The aim of this translation is to normalize and to represent explicitly the lexical properties encoded in LG tables. IL representations are independent of any particular linguistic theory. (2) IL is used to generate lexicons for NLP applications based on unification grammars. We have build an HPSG lexicon used within the ALEP system to parse French, and a TAG lexicon used for French text generation. These lexicons are dual of one another since, for a each entry, the first represents the properties that hold while the later represents the ones that do not hold. The generation of these lexicons raises interesting questions regarding the lexicon organization in these theories. (3) The evaluation of LG coverage on a corpus. This evaluation uses a French shallow parser able to recognize quite precisely the constituents that the verbs take as arguments. The lexical descriptions of the verbs can then be saturated in order to recognize the phrases headed by these verbs.

1. Introduction

One of the main obstacle to the development of wide coverage NLP systems is the absence of large computational lexicons. Migration of existing lexical resources constitutes an efficient and cheap way to build such large lexicons. This paper addresses some problems related to the reuse for NLP and to the evaluation of LADL's Lexicon-Grammar(LG) (Gross, 1975). This major source of French lexical knowledge is composed of 61 tables describing the syntactic behavior of 4,961 verbs (10,716 entries) w.r.t. a set of uniform of syntactic properties. LG has been publicly available on the Internet for several years [1]. However, it has not been used by the NLP community, mainly because of its format. The interpretation of these tables is non trivial because large parts of the linguistic informations they contain are neither explicit nor represented in a uniform manner. Section 2 discusses this question in more detail.

The research presented in this paper has 3 aims. The first is to make explicit the informations coded in LG tables and to give them a formal representation, namely PATR-II lexical entries. PATR-II is a well known formalism easy to use and flexible enough to enable us to represent LG informations straightforwardly. The conversion of LG into a theoretical-independent PATR-II Intermediate Lexicon (IL) implements the interpretation of LG syntactic properties, as described in LG underlying theoretical researches (Gross, 1975; Boons, Guillet, & Leclère, 1976a, 1976b; Guillet & Leclère, 1992). Section 3 is dedicated to IL and to its construction. Our second aim is to translate IL into lexicons for target NLP applications. IL plays a role of pivot between LG and these lexicons. We have build a HPSG lexicon used within the ALEP system to parse French (Alshawi, Arnold, Backofen, Carter, Lindop, Netter, Pulman, Tsujii, & Uszkoreit, 1991), and a TAG lexicon used for French text generation (Meunier, 1997). These lexicons are dual of one another since the first describes the properties that hold while the later describes the ones that do not hold. As will be seen in section 4, the generation of these lexicons raises some interesting questions regarding the lexicon organization in these theories. The third aim of the research we present is the evaluation of LG coverage on a large corpus (section 5). Such an evaluation is necessary because LG has been constructed by hand, mainly by introspection. The evaluation uses a French shallow parser able to recognize quite precisely the constituents that the verbs take as arguments (namely DPs, PPs and APs). The parser uses IL verbs lexical descriptions in order to recognize finite propositions, non finite verb pharses, gerunds and adjectival phrases headed by past participles.

[1] http://www-lli.univ-paris13.fr/LexiqueGrammaire/

2. Tables

LG describes the main aspects of the syntactic behavior of French predicates in a tabular format with binary values (+/−). This description includes the intrinsic properties of these predicates, their licensed constructions and the alternative realizations of these constructions constituents.

Sujet						Adjectif				Comp. direct								
N0 =: Nhum	N0 =: Nnr	N0 =: le fait Qu P	N0 =: V1 W		V "concret"	N0 V	-a =: -ant	-a =: -able	-a =: -eux	-a =: -(at)eur	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	N1 se V de ce Qu P	N1 se V auprès de N3 de ce Qu P	N1 est Vpp de ce Qu P	[passif par]	[passif de]	N0 V N1 contre Nhum
+	+	+	+	abasourdir	-	+	+	-	-	-	+	-	-	-	-	+	+	-	-
+	+	+	+	abattre	+	+	-	-	-	-	+	+	-	+	-	+	+	-	-

Figure 1: Beginning of table **4** (Gross, 1975)

2.1. Description

LG describes 4,961 verb predicates by means of 61 tables. Each table corresponds to one verb class identified by a number and defined by a *base construction*. LG constructions are given as linear expressions made of predefined symbols. For instance, table **4** (figure 1) gathers the verbs `V` whose base construction is `Qu P V` $\mathtt{N}_1$ where `V` has a propositional subject `Qu P` and a nominal complement $\mathtt{N}_1$; the subject is a finite proposition either indicative or subjunctive, introduced by a subordinating conjunction `que`.

Properties Typology. Each table describes a verb class by means of a set of properties characteristic of this class. The properties of the verbs, represented in the columns of the table, are of three types: *constructions*, also called "transformational properties" (Gross, 1975), represent the main variations of the syntactic frame of the verb (e.g. passivization, optionality of some complements, etc.); *constituent specifications* (also called "partial transformations") completely specify a construction constituent (e.g. $\mathtt{N}_0\mathtt{=:}\mathtt{N}_{hum}$; the subject is a human noun); *feature specifications*, such as `Aux=avoir` (the verb is conjugated with the *avoir* auxiliary), only specify a feature of a constituent. Constituent and feature specifications are also called "distributional properties".

Figure 1 illustrates these three kinds of properties. Column 1 specify the subject constituent $\mathtt{N}_0$ of the base construction. Columns 8 to 11 specify the adjectivation suffixes of `V`. Adjectivation suffixes are regarded as features since they neither specify syntactic configuration for `V` nor its arguments. Column 15 ($\mathtt{N}_1$ `se V de ce Qu P`) is a transformation which describes a pronominal alternative of the base construction.

Table Structure. The columns are organized by means of three structural elements. First, horizontal *cartouches* indicate the part of the construction concerned by the properties it dominates. For instance, the first four columns of figure 1 are dominated by `Sujet` which indicates that they specify the construction subject. Second, *dependences* indicate that some properties depend on another one. They are represented graphically by embedding the dependent columns into the one which controls them. Columns 3 and 4 of table **4** are an example of such structure which refines the specification of the subject when it is "not restricted". Third, *disjunctions* of columns indicate that some properties depend on several other ones. They are used as abbreviations in order to avoid duplicating the dependent properties and they only appear as controllers.

2.2. Interpretation

In addition to their explicit typographic structure, LG tables have also an underlying structure which specifies their properties interpretation. This interpretation is jointly defined by the properties typology and by the table structure.

Base Construction. As can be seen in figure 1, the base construction of a table is not a property of that table. We added it in order to make the table interpretation more regular. A base construction being a transformation which all other properties depend on, it is added as a controller of all the table columns. All lines of the added column are + since this property is the base construction of all the table verbs, and therefore is licensed for them.

Reference Construction. In order to be interpreted, any property used in a table need to be attached to a *reference construction*. The reference construction of a distributional property is the construction which contains the constituent specified by the property. As for transformations, their reference construction is the construction which they are derived from.

Reference Constituent. The interpretation of distributional properties also requires to know which of the reference construction constituents is concerned by the property. This information is also needed for pronominalization. In the electronic version of the tables, the relevant constituent may be identified from the property prefix.

3. Intermediate Lexicon

The translation of a LG table into a set of lexical entries consists in: (1) making explicit its underlying structure; (2) associating formal representations with its properties, namely PATR-II constraint systems; (3) completing the constructions representations by exploiting the underlying structure. The resulting lexicon associates a set of fully specified construction representations with each table entry. From an operational point of view, IL construction is made in successive passes in order to have a more robust translator and to

distinguish the different levels and dimensions of the translation. We already have processed a third of LG, that is 18 tables, describing 2,589 verbs (3,485 entries).

3.1. Representation

An examination of the tables constructions shows that they all are composed of a subject, a verb and complements whose number ranges between 0 and 4. We choose to use *canonical representations* (Günthner, 1988) for these constructions, namely to consider that all of them are of the following form:

(1) $\texttt{Sujet Verbe Compl}_1\ \texttt{Compl}_2\ \texttt{Compl}_3\ \texttt{Compl}_4$

where `Verbe` may be a verbal segment possibly containing an auxiliary or a reflexive pronoun and where $\texttt{Compl}_1$, ..., $\texttt{Compl}_4$ may be empty; however, if $\texttt{Compl}_i$ is empty, then $\texttt{Compl}_j$ is empty as well for all $j > i$. Each construction may then be represented as a structure with 6 parts, each part describing one constituent.

We also choose not to make any hypothesis on the internal structure of the constituents; first, we want IL to be a formalization of the tables which reflects the views of their authors on that matter; second, strong theoretical hypotheses would make the construction of target lexicons for NLP systems based on other linguistic theories more difficult because these hypotheses are likely not to be shared by these theories.

3.2. Implementation

The first stage in IL construction consists in restoring the explicit typographical structure in the tables electronic version: there headings are separated and then manually SGML marked. The reminder of the construction is fully automated.

Headings processing. The second stage translates the marked headings and the tables lines into PROLOG terms by means of a PERL filter. Then the dependences with disjunctions are splited. The following pass determines the type of each table property by shallowly parsing of the columns headings. The fifth stage assigns identifiers to constructions and determines the properties reference constructions. The next pass actually parse the properties and computes their PATR-II representation.

Lines processing. The following stages perform the inheritance by the derived transformations of their constituents properties. First, the table columns are reorganized according to the underlying structure. The table becomes a set of 3-uplets $\langle T, F, C\rangle$ where T is a construction, F a set of feature specifications of T constituents, and C a set of constituent specifications attached to T. The feature specifications constraints are merged into the constructions and constituents representations. Then the properties of the inherited constituents are added to the transformations representation. The last stage merges the inherited representations into the target ones and takes into account the effects of the transformation.

4. Computational Lexicons

IL is the input of a translation system that generates in parallel two files of lexical resources to be reused in NLP applications for French that are respectively based on HPSG and TAG linguistic theories[2]. The system has been applied to the IL representations of the four following LG tables : **4** (divalent direct transitive verbs with a sentential subject), **36DT** (trivalent direct transitive verbs with a dative/source secondary object), **38L** (tetravalent direct transitive verbs with both source and goal locative objects), and **1** (divalent indirect transitive verbs with a prepositional infinitive clause object).

4.1. Comparing Targets

The target lexica are based upon two distinct theories. Both theories are recent, and include, among others, the concepts of "unification grammar" and "lexicalism". Such concepts involve for a lexical entry to describe not only the word itself, but also and above all its internal structure and its maximal projection. Moreover, a word description obeys a feature theory, called "type system" (henceforth TS) in HPSG, and "family" in TAG.

On the other hand, these targets have conceptual differences, some of which playing a crucial role in the migration algorithm.

Target Applications. The HPSG lexicon has been used in a parser written in ALEP (Alshawi et al., 1991), whereas the TAG lexicon is written in the G-TAG formalism (Danlos, 1995; Danlos & Meunier, 1996; Danlos, 1998) for the FLAUBERT generation system (Meunier, 1997). However, this difference is not very relevant, given the expected grammars reversibility in both theories.

Interpretation. As it will be shown , an HPSG lexical entry (see § 4.2) reflects one of the possible constructions for a word, whereas a TAG entry (see § 4.3) describes all the illegal constructions of a given word.

Underlying Architecture. The HPSG features and their values are defined by means of a near-to-standard TS[3]. Within the limits of the constraints expressed by this TS, the representation of any construction, any feature sharing or calculation is easily realizable, and any potential change in the TS would have only minimal effects on the lexical entry description. The TAG notion of family is different (Candito, 1996): it is a symbol, that identifies for its predicate members all the potential constructions and transformations, in terms of a set of parametric trees labeled with the appropriate features. Thus, the definition or modification of a family entails much heavier constraints in the lexical entries interpretation, than what happens with the HPSG TS notion.

[2] We do not introduce here these theories. For that purpose, the reader may refer to (Pollard & Sag, 1988), (Pollard & Sag, 1994) and (Joshi, 1987).

[3] It results from the combination of the (Pollard & Sag, 1994), chap.9 TS, and the semantic extensions defined in (Badia, 1998).

4.2. The HPSG Lexicon

An HPSG lexical entry is a Typed Features Structured linguistic description in which only positive properties are expressed[4]. In other words, representing the fact that a verb admits no passive construction is simply done by *not* defining the verb passive construction lexical entry.

The basic algorithm for the IL conversion into HPSG lexical entries has been exposed in (Hathout & Namer, 1997). After a brief sketch of it (§ 4.2.1), we present the inheritance mechanism that has been added to it: it retrieves features that belong to a reference constituent and are inherited (but not repeated in the IL) in the transformation the conversion of which is in progress (§ 4.2.2).

The generation of an HPSG lexicon rests on two main principles:

1. the resulting lexicon is completely static (no lexical rule is assumed); in other words, there is a one-to-one correspondence between an IL formula (see § 3) and an HPSG lexical entry,
2. generalized disjunctions are turned down, replaced by the application-oriented co-description approach developed in (Rieder, Schmidt, & Theofilidis, 1994).

4.2.1. Basic Algorithm

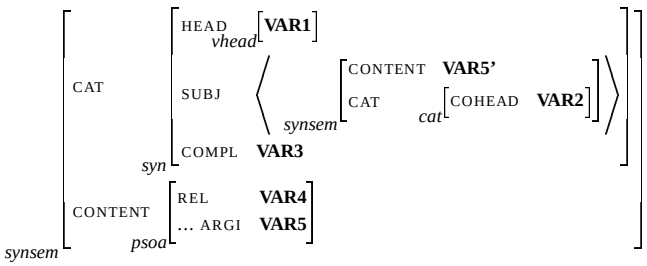

Figure 2: HPSG lexical entry: general structure

As shown by figure 2, both basic construction and transformations are instances of a generic structure, which reflects the involved TS and where assigning the variable slots (**VAR1**, ...) requires to reorganize the features of the corresponding IL construction:

- The verb CONTENT is directly computed according to the table basic construction.
- Syntax-to-semantic interface (ie. the constituents CONTENT index values) depends on the transformation effects on constituents.
- For the grammatical function of each constituent (eg. SUBJ) the list of the potential syntactic realizations is itemized (eg. the "np:vp" list means that the given grammatical function can be realized either as an NP or as a VP), and such is the appropriate list of properties for each (grammatical function, syntactic realization) pair in order to instantiate the COHEAD attribute (eg. the **VAR2** slot in figure 2).

4.2.2. Retrieving Inherited Features

In addition to the basic algorithm, transformations inherit semantic features from their reference constructions, following a mechanism illustrated by Table 1, and distributed in 4 phases:

(A) : F-FAMILY **for V=acheter**

	Construction	Reference	Features
(1.A.1)	cons/0	*/*	...
	cons/0	*/*	verb = "acheter" ...
(1.A.14)	cons/13	*/*	compl2.sem.sort = nhum
	…	…	…
	t1/6	cons/0	sujet <= base.sujet ...
	t1/6	cons/0	compl1.sem.sort = partie_corps, compl1.sem.ref = sujet ...
	…	…	…

(B) : The `[Ppv2 = lui]` **Transformation**

	Construction	Reference	Features
(1.B.1)	t2/14	cons/13	sujet <= base.sujet ...
(1.B.2)	t2/14	cons/13	cat=pro, compl2 <= base.compl2, compl2.form= lui ...
	t2/2	*/*	sujet.sem.sort=hum ...

Table 1: Use of F-FAMILY to synthesize a transformation

1. For the synthesis of the V verb ith transformation (eg. the `[Ppv2=lui]` transformation fore "acheter" (buy), in Table 1.B), the F-FAMILY table is built (eg. 1.A), and stores the already synthesized constructions.
2. Each IL property to be synthesized is specified by two symbols (see 1.B): the first one identifies the current property description (eg. **t2/14**) and the second one, the reference construction the property depends on (eg. **cons/13**).
3. Recovering an inherited property for a function (eg. compl2 in (1.B.2), vs subj in (1.B.1)) requires the reference construction to be retrieved in the F-FAMILY table, provided that the functions are compatible. For instance, the compl2 reference constituent in (1.B.2) is itself a compl2 (compl2 <= base.compl2) and this item is the actual constituent described in the reference construction **cons/13** in (1.A.14). Therefore, compl2 in the (B) transformation inherits the corresponding semantic features (sem.sort = nhum). Conversely, the subj reference constituent (subj <= base.subj) in (1.B.1) does not appear in the **cons/13** reference construction in (1.A.14): the subject synthesis in (B) does not involve inherited features to be recovered from the (A) F-FAMILY.
4. Step 3 is repeated as long as examined constructions depend themselves upon reference ones.

4.3. The TAG Lexicon

TAG lexicon design follows quite opposite techniques: as it has been said in 4.1, the definition of the TAG lexical entry of a V verb is in some way "negative", because it is mainly made up of a set of (conjunction of) negative features, each of them identifying a syntactic construction valid for the V TAG family, but which is illegal for V.

```
"ENTREE5" =        f-N0VN1p2N2;
                   [xxx] = /acheter/, [yyy2] = /à/;
                   [T_passive=−,T_no_N0=−,T_no_N2=−],
                   ... [T_N2loc=−];
                   N0:role= agent;
                   N1:role= thème;
                   N2:role= origine;;
```

Figure 3: TAG entry for "acheter"

[4] We mean here complex, structural properties, of course, i.e. atomic or binary negative values are allowed.

For instance, consider figure 3. The verb “acheter” (buy) is a member of the TAG family “f-N0VN1p2N2”. This sequence of formal symbols means that the family gathers verbs which share the “subject V direct_object prep_object” construction: the family identifier and interpretation apparently bring TAG and LG much closer than what happens with HPSG (here, there is a one-to-one correspondence with LG table **36DT**), and the migration algorithm design seems to be much simpler. We will show the limits of such an assumption.

According to figure 3, attributes [xxx] and [yyy2] respectively indicate the verb and the preposition forms. The “[T_N2loc=–]” feature says that the “p2N2” constituent cannot be realized as a locative complement. It also implies that the property is potentially valid for the family. Similarly, the “[T_passive=–, T_no_N0=–, T_no_N2=–]” indicates that the conjunction of the 3 properties : “passivization, agent omission and N2 omission” is forbidden for the verb (and allowed for the family).

Other features that characterize a TAG lexical entry, are the constituents semantic role and the verb morpho-syntactic properties (such as the conjugation auxiliary). The constituents syntactic realizations are also properties that must appear on the lexical entry: we will see that they entail problems in the family definition.

4.3.1. General Strategy

Observing figure 3, it results that the following decisions and tasks have to be done to migrate IL into TAG:

- One LG table corresponds to one TAG family (this choice is put in question in § 4.3.2);
- Table `transfo`, illustrated by Table 2 puts in correspondence each LG table with its valid transformations.

Table	Construction	Initial Value
36DT	[N0hum,V,N1,à,N2hum]	0
	…	…
38L	[N0,V,N1,de,Nsource,Loc,Ndest]	0
	[N0,V,N1,Loc,Ndest]	0
	…	…

Table 2: Valid constructions for each LG tables

  As the start, each transformation has the binary ‘0’ value;
- Table `t-to-f` (see Table 3) associates an LG transformation with a TAG negative (conjunction of) feature(s);

LG Illicit Construction	Negative TAG Features
[N0,V,N1,Loc,N]	[T_N2loc=–]
[N1,est,Vpp]	[T_passive=–,T_no_N0=–,T_no_N2=–]
[N1,V]	[T_ergative=–]
[N1,est,Vpp,W]	[T_passive=–,T_no_N0=–]

Table 3: Sample of LG to TAG correspondence

- Given the TAG definition of the lexicon, there should be a one-to-one correspondence between a LG table line and a lexical entry.
- The F-FAMILY table definition (see § 4.2.2) is the common module of the HPSG and TAG translators. To generate the TAG entry of a V verb, the translator exploits the maximal V F-FAMILY table, i.e. the one that contains all the formulas that have to do with V;
- The F-FAMILY is used to collect the positive (i.e. allowed) transformations for V, its lemma, the preposition forms (if any), the semantic roles and other specific features for V.
- The positive constructions for V are matched against Table 2. The corresponding ‘0’ values are switched to ‘1’. The TAG negative features corresponding to the remaining ‘0’-valued transformations (Table 3) are included in the lexical entry definition.

4.3.2. Problem : The LG Table **4**

In appearance, the TAG migration is the result of a short simple algorithm, because the main syntactic descriptions, sharing, constraints, etc. are the matter of the family definition. However, there is a situation for which this simple algorithm has to be put in question, namely when a constituent has several syntactic (and semantic) realizations, as it happens for the conversion of the LG table **4**.

```
"ENTREE12" =        f-S0VN1;
                    [xxx] = /abattre/;
                    ...
                    S0:cat= np,cp,vp;
                    N1:cat= np;
                    S0:mood= ind;
                    S0:compl= que;
```

Figure 4: Disjunction in the TAG entry for “abattre”

The (flat) structure of a TAG lexical entry is not compatible with multiple syntactic realizations, because cross-dependencies between syntactic realizations and other phenomena cannot be expressed: for instance, figure 4 shows that the subject of “abattre” may be either a (tensed or infinitive) clause, or an NP: the subject mood and complementizer features values concern only the tensed clause realization, and this constraint is not represented in figure 4.

It is clear that the one-to-one correspondence between LG tables and TAG families are not sufficient to solve this problem. A possible solution is to define a set of sub-families for “f-S0VN1”: each sub-family gathers verbs that share the same syntactic realizations, for each constituent. The maximal expected partition of “f-S0VN1” is illustrated in figure 5, and according to this figure, the family of “abattre” would become “f-S0VN1-a”.

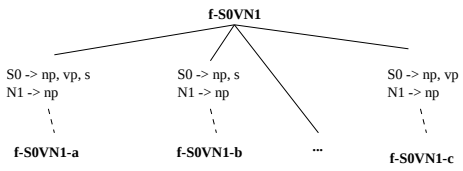

Figure 5: Partition of f-S0VN1

4.4. Validation and Perspectives

The conversion has provided a thousand TAG entries, and around four thousand HPSG entries. The HPSG lexicon has

been validated in the ALEP platform (Heyd, Jacquey, & Namer, 1996). The validation of the TAG lexicon is foreseen to be performed in the framework of the FLAUBERT generation system.

Future extensions, within the TAG formalism, will have to do, above all, with families/tables harmonizations. They are expected to improve the current TAG linguistic resources, for both parsing and generation.

5. Evaluation

LG is first and foremost a set of theoretical syntactic description of French verbs. It has been constructed mainly by introspection and is intended to reflect its authors intuitions and generalizations on key syntactic phenomena. The third part of the paper is dedicated to the evaluation of the "usefulness" of LG (in fact, IL) for NLP.

5.1. Shallow Parsing

Why a shallow parser? LG being a syntactic lexicon, its evaluation consists in determining its contribution to standard syntactic processing such as parsing, generation, etc. Parsing is particularly suited for this evaluation because it exploits IL straightforwardly. Furthermore, IL is quite large (2589 verbs; 3485 entries). This forbids an evaluation by hand. On the other hand, its coverage must be evaluated on a quite large corpus. Therefore a fast syntactic parser have to be used. Ideally, the evaluation should be made with a parser able to take advantage of all the dimensions of IL descriptions: structural, semantic and morphological (eg. able to distinguish human/non human/location/measure NPs, locative/non locative prepositions, etc.). However, we cannot develop such a parser because we lack a large general syntactic and semantic lexicon (IL is precisely a step toward building such a lexicon) and because of the complexity of such a development. Not to mention the system complexity and its effects on efficiency. Since we cannot afford the ideal solution, IL evaluation uses a French robust shallow parser, implemented in PROLOG (Sicstus 3.5) with reasonable efficiency: around 50 words per second on a Pentium 100 PC running Linux. We are aware of the obvious limits of this option, the major one being the lack of precision: only IL structural informations are taken into account.

Overview. The shallow parser we developed for IL evaluation takes POS tagged and lemmatized texts as input. So, corpora have first to be pre-processed:

1. segmentation of the text in sentences and words;
2. identification of the compounds;
3. removal of the SGML tags;
4. POS tagging using Brill tagger trained for French at CNRS-INaLF by Josette Lecomte and Patrick Paroubek (Lecomte & Paroubek, 1994);
5. robust lemmatization using TLFnome, a lexicon derived from "Trésor de la Langue Française" word list. The lemmatizer computes lemmata for unknown words by means of flexion rules learned from TLFnome.

The shallow parser processes texts in several passes. Parsing a text consists in delimiting sentences constituents and in bracketing them (all constituents are compact). The parser also associates with each constituent the structural informations computed during its parsing, namely its head and the list of its arguments.

What does "coverage evaluation" mean? IL coverage can be characterized in two ways. First, "quantitatively" by determining the proportion of verb occurrences that are described in IL. Second, "qualitatively" by determining its lexical descriptions adequacy, that is the proportion of verb occurrences that have an entry which properly describes the construction they occurs into.

5.2. Verb Arguments

The shallow parser segments sentences into constituents. It is robust in the sense that it skips the items that do not belong to recognizable constituents. The basic constituents it recognize are DPs, APs, PPs and verb strips (VP0s, that is a verb with its possible auxiliaries, clitic pronouns, negative adverbs and surrounding adverbs); see figure 6.

In order to evaluate IL adequacy, the parser must recognize with a fair precision the verbs arguments. These arguments are DPs, PPs, non finite verb phrases and finite clauses. The treatment of infinitives and finite clauses is presented below (§ 5.3.2 and § 5.3.3). The main problem in the delimitation of non propositional constituents is prepositional attachment to DPs and APs: verbs must not take as argument PPs that are in fact attached to one of their arguments.

Similar methods are used for attachments to DPs and APs, both being based on learning licensed attachment configurations from the corpus (Bourigault, 1994). Endogenous learning is used for attachment to DPs: determiners are divided in 4 classes ($\emptyset$, *le*, *un*, *others*); prepositions are divided in 4 classes as well (*de*, *à*, *sur*, *others*); the parser uses the $\langle N^0$ head, Prep, Det$\rangle$ configuration to determine whether the following Prep headed PP is always/never attached to the N^0 headed DP or if it must look in the corpus for identical configurations but with PPs having different complements. Endogenous learning gives good results, especially with technical texts. PPs attachment to APs uses the same method, however, the $\langle A^0$, Prep$\rangle$ configurations have been learned once for all from 50 Frantext scientific texts because the learning configurations are very strongly constrained. Notice that past participles used as adjectives are treated differently (by means of IL; see § 5.3.1).

5.3. Implementation

The first stage of IL implementation into the shallow parser is to translate it into a PROLOG external database. IL entries are translated as PROLOG terms with 6 arguments that have the format of the representations manipulated by the parser. The translator is a PROLOG program that extracts the structural informations from IL PATR-II constraint systems and

```
[IP [DP Un/dtn [NP lac/sbc NP] DP] a/acj souvent/adv occupé/vpar [DP les/dtn [NP
ombilics/sbc NP] DP] IP] que/sub$ [DP l'/dtn [NP action/sbc [PP de/prep [DP la/dtn
[NP glace/sbc NP] DP] PP] NP] DP] [VP0 a/acj surcreusés/vpar VP0] ./ponct
```

Figure 6: Segmented sentence

	Infinitive	Present Participle	Past Participle	Finite VP	Finite IP	Total	Rem.
Present	7,075	2,435	16,904	13,036	13,879 (+ 13,036)	53,329	VP & IP
	13.26%	4.56%	31.69%	24.44%	26.02%	100%	VP & IP
	54.82%	58.21%	67.96%	45.17%	32.47%	62.95%	VP & IP
Missing	5,830	1,748	7,971	15,820	15,820	31,369	IP = VP
	18.58%	5.57%	25.41%	50.43%	50.43%	100%	IP = VP
	45.17%	41.78%	32.04%	54.82%	67.52%	37.03%	IP = VP
Total	12,905	4,183	24,875	28,856	13,879 (+ 28,856)	84,698	VP & IP
	15.23%	4.93%	29.36%	34.06%	16.38%	100%	VP
	100%	100%	100%	100%	100%	100%	VP
Parsed	4,877	1,919	16,904	3,924	13,879	41,503	VP + IP
	11.75%	4.62%	40.72%	9.45%	33.44%	100%	VP + IP
	68.93%	78.80%	100%	31.10%	51.56%	77.82%	VP + IP
Fail	2,198	516	0	9,112	(13,036)	11,826	VP
	18.58%	4.36%	0%	77.05%	–	100%	VP
	31.06%	21.19%	0%	69.89%	48.43%	22.17%	VP
Total	7,075	2,435	16,904	13,036	13,879	53,329	VP + IP
	13.26%	4.56%	31.69%	24.44%	26.02%	100%	VP + IP
	100%	100%	100%	100%	100%	100%	VP + IP

Table 4: Detailed results for corpus 1

translates them as representations of the parser; IL 3,485 entries yield 13,570 PROLOG constructions.

5.3.1. Past Participles used as Adjectives

The treatment of past participles used as adjectives differs from the handling of the other verb phrases in three respects. First, because all part participles complements are optional, it does not enforce the lexicon completeness hypothesis: the parsing of such phrases succeeds even if the verb does not have a lexical entry that matches its actual complements list. Therefore, the parsing of these APs never fails (see the third column of the second part of table 4). Second, it implements a lexical rule that transforms the subject of active clauses into an additional agent complement (ie. a *par* headed PP). Third, the recognition of these phrases is performed at the same time as the other APs and not in the pass that parses VPs. Notice that part participles occurring in compound times VPs are treated as normal verbs (see § 5.3.2).

5.3.2. Non Finite Verb Phrases

Non finite verb phrases differ from finite clauses because they do not have subjects. However, the parser implements strictly the lexicon completeness hypothesis for both of them. The treatment of non finite verb phrases is composed of 4 stages. First, the verb strip is (re)parsed in order to identify the possible clitic arguments. Second, the parser delimits a segment immediately following the verb strip in which it has to find the verbs complements. This segment ends either by a punctuation or by an element marking the limit of the current clause (relative pronoun, subordinating conjunction, verb strip). Third, the parser looks for a lexical entry that can be completely saturated by the clitic pronouns and the constituents of the selected segment. The complements order in the lexical entry are is ignored; the complements taken from the segment can be disconnected (we allow items that are not arguments of the verb to intervene between the verb complements). The last stage computes the VP representation.

Non finite VPs are non finite verb and present participle phrases. They are parsed in a separate pass. In order to allow non finite verb VPs to be complements of operator verbs (eg. *commencer à*, *menacer de*, etc. described in LG table **1**), this pass treats sentences from right to left.

5.3.3. Finite Clauses

Finite clause are parsed in a third, separate pass. They are delimited on the left by the beginning of the sentence, a punctuation, a relative pronoun or a subordinating conjunction. The parser imposes strong constraints on the location of subjects (except for subject-verb inversions, non clitic subjects must immediately precede the verb strip). As a result, subject attachment has a good precision, but the parser fails to find one in the third of the cases where the finite VP is recognizable (see the fourth column of the second part of table 4). The parser performs a fourth pass in order to recover these VPs.

5.4. Results

The shallow parse have been used to evaluate LG on two corpora: one of 983,315 words composed of 4 scientific books from Frantext (in the domains of geomorphology, biology and chemistry) and one of 300,450 words composed of articles from the 1987 editions of "Le Monde" newspaper. The results obtained show that LG quantitative coverage is similar for both corpora (around 70%). This evaluation is quite reliable since it only depends on the tagger which has a precision of 97%. On the other hand, LG qualitative coverage varies with the corpus nature, partly because the tagger have been trained on Frantext texts and therefore makes less errors for these texts.

Present	Missing	Total
16,126	10,629	26,755
60.27%	39.72%	100%
Parsed	Fail	Total
11,446	4,680	16,126
70.97%	29.02%	100%

Table 5: Brief results for corpus 2

Notice that the 77.82% and 70.97% rates given in table 4 and 5 includes past participle phrases since IL is used for there treatment. However, this treatment never fails which somehow perverts the rates. They remain however decent even when past participles are not counted: 67.53% for corpus 1 and 50.95% for corpus 2.

6. Perspectives

This first evaluation of LG gives a good idea of what this lexicon can bring to NLP. We plan to carry out a more precise evaluation as soon as IL reaches it final size and when the shallow parser we are using will become more stable. In particular, we must evaluate the parser performance in order to determine the error rate for IL evaluation.

On the other hand, we are working on several other aspects of LG use in NLP, especially on:

- The coupling of IL and VERBACTION, a lexicon of action names extracted from TLFnome in order to enhance complexes NPs parsing;
- The adaptation of the method presented here to the "Trésor de la Langue Française" constructions and *entre-crochets*. TLF constructions are similar to LG one on many aspects. They are less strongly structured (in particular, there are no explicit verb classes) but their arguments are described in more detail.
- The enrichment of IL with new constructions learned from corpora. For each verb of the corpus that either is missing from IL or such that its phrase cannot be parsed, we collect IL constructions that can be used to parse this phrase. Then statistical and linguistic filters are applied so select the constructions that can be used reliably for these verbs.

References

Alshawi, H., Arnold, D., Backofen, R., Carter, D., Lindop, J., Netter, K., Pulman, S., Tsujii, J., & Uszkoreit, H. (1991). Eurotra ET6/1: Rule Formalism and Virtual Machine Design Study (Final Report). Tech. rep., Luxembourg, CEC.

Badia, T. (1998). Predicate-Argument Structures. In P. Schmidt & F. Van Eynde (Eds.), *Linguistic Specifications for Typed Feature Structures Formalisms*, Vol. 10 of *Studies in machine translation and natural language processing*. Luxemburg: European Commission.

Boons, J.-P., Guillet, A., & Leclère, C. (1976a). *La structure des phrases simples : Constructions intransitives*, Vol. 8 of *Langue et Culture*. Genève: Librairie Droz.

Boons, J.-P., Guillet, A., & Leclère, C. (1976b). La structure des phrases simples : Classes de constructions transitives. Rapport de recherches du LADL 6, Université Paris 7, Paris.

Bourigault, D. (1994). *LEXTER, un Logiciel d'EXtraction de TERminologie. Application à l'acquisition de connaissances à partir de textes*. Thèse de doctorat, École des Hautes Études en Sciences Sociales, Paris.

Candito, M.-H. (1996). A principle-based hierarchical representation of LTAGs. In *Proceedings of the International Conference on Computational Linguistics COLING'96*, Copenhagen, Danemark.

Danlos, L. (1995). Présentation de G-TAG, un formalisme pour la génération de textes. In *Actes de TALN'95*, GDR–PRC Communication Homme-Machine. Pôle Langage Naturel, Marseille.

Danlos, L. (1998). *G-TAG: a Formalism for Text Generation inspired from Tree-adjoining Grammar: TAG issues*. Stanford, US: CSLI.

Danlos, L., & Meunier, F. (1996). G-TAG: Présentation et applications industrielles. In *Informatique et Langue Naturelle, ILN'96, Nantes*.

Günthner, F. (1988). Features and Values 1988. Tech. rep. SNS-Bericht 88–40, Tübingen University, Tübingen.

Gross, M. (1975). *Méthodes en syntaxe : Régime des constructions complétives*, Vol. 1365 of *Actualités scientifiques et industrielles*. Paris: Hermann.

Guillet, A., & Leclère, C. (1992). *La structure des phrases simples : Constructions transitives locatives*, Vol. 26 of *Langue et Culture*. Genève: Librairie Droz.

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

Heyd, S., Jacquey, E., & Namer, F. (1996). E-LS-GRAM Lingware Development Documentation. Core Grammar and Extentions for FrenchLRE 61029. Tech. rep., TALANA, Nancy.

Joshi, A. (1987). The relevance of tree adjoining grammar to generation. In G. Kempen (Ed.), *Natural Language Genaration: New Results in Artificial Intelligence, Psychology and Linguistics*. Martinus Nijhoff.

Lecomte, J., & Paroubek, P. (1994). Premiers essais de l'assignateur de catégories grammaticales d'E. Brill sur des textes français. Tech. rep., INaLF, Nancy.

Meunier, F. (1997). *Implémentation d'un formalisme de génération inspiré de TAG*. Thèse de doctorat, Université Paris 7, Paris.

Pollard, C., & Sag, I. (1988). *An Information Based Approach to Syntax and Semantics, Volume 1*. CSLI Lecture Notes. Chicago: University of Chicago Press.

Pollard, C., & Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. CSLI Lecture Notes. Chicago: Chicago University Press.

Rieder, S., Schmidt, P., & Theofilidis, A. (1994). German LS-GRAM. Lingware Development Documentation. LRE 61029. Tech. rep., IAI, Saarbrücken.