(Semi-)Automatic Generation of the ALEP Analysis Lexicon

Nabil Hathout
CNRS – INaLF
Château du Montet
Rue du Doyen Roubault
F-54506 Vandœuvre-lès-Nancy
France
hathout@inalf.cnrs-nancy.fr

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

1 Introduction

This paper presents the results of an experiment of lexical resources migration, aimed at providing unification and lexicalization based syntactic parsers of French, with lexicons. The resources we use are the Lexion-Grammar tables of LADL Gross (1975), publicly available at the following URL : `http://www-ceril.univ-mlv.fr/LexiqueGrammaire/`. 61 tables concern verbs and verb phrases. They describe the syntactic behaviour of more than 10,000 of items.

The LADL tables have hardly been exploited by automatic tools because their interpretation is quite difficult. The aim of our experiment is to convert the LADL verb tables into intermediate PATR-II representations, which, in turn, are translated into verb lexicons for NLP systems based on linguistic models like HPSG or LFG.

The reminder of the paper is organised as follows : section 2 describes the structure and interpretation of the verb tables. Section 3 presents our approach, the motives of the choices we have made and the translation of the tables into an intermediate PATR-II lexicon. Section 4 describes the building of ALEP lexicon entries from the intermediate lexicon. Section 5 discusses the results of this experiment and gives a short conclusion.

2 Verb tables

Lexion-Grammar represents the main aspects of the behaviour of French predicates (verbs, verb phrases, predicative nominals and adjectives, etc.) in a tabular format with binary values (+/-). These aspects include the intrinsic properties of the predicates as well as their licenced constructions and these constructions constituents properties.

2.1 Description

The Lexion-Grammar describes more than 10,000 verbal predicates by means of 61 tables. Each table corresponds to one verb class identified by a number and defined by a *base construction*. The constructions are given as linear expressions made of predefined symbols. For instance, table 1 (figure 1) gathers the operator verbs U whose base construction is$\mathtt{N}_0$ U Prép $\mathtt{V}^0$ $\Omega$where U has a nominal subject $\mathtt{N}_0$ and a complement; this complement is an infinitive verb phrase$\mathtt{V}^0$ Ω whose subject is understood as being $\mathtt{N}_0$ and whose complements Ω are not detailled; the complement of U is marked by a preposition Prép. The 61 tables divide into four groups : 19 describe verbs with completive arguments Gross (1975), 17 transitive verbs Boons et al. (1976a), 9 intransitive verbs Boons et al. (1976b) and 16 transitive locative verbs Guillet and Leclère (1992).

A 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 two types : the "transformational" ones, as they are called by Gross 1975, represent the main variations of the syntactic frame of the verb (e.g. passivation, optionality of some complements, etc.) ; the "distributional" ones, specify the constituents which occur in the base construction and its transformations. Distributional properties can be of two sorts : *constituent specifications*, called "partial transformations" in Gross (1975), (e.g.N_0 =: N_{hum} ; the subject is a human noun) completely specify a construction constituent while feature specifications, such asAux = avoir (the verb is conjugated with the *avoir* auxiliary), only specify a feature of a constituent.

Figure 1 illustrates this three kinds of properties : column 1 is a distributional property specifying the subject constituent N_0 of the base construction ; column 4 specifies the auxiliary feature of the main verb U ; column 7 (N_0 U) is a transformation which describes the optionality of the infinitival complement.

Sujet			Aux =: avoir	Aux =: être	N0 est Upp W	N0 U	Compléments indirects																		
N0 =: Nhum	N0 =: Nnc																					N0 U dans N1	N0 U N1	N1 =: Nhum	N1 =: N-hum
								N1 =: que P	N1 =: que Psubj	Infinitives								N0 U Prép N1	N1 =: Nhum	N1 =: N-hum	N1 = Ppv				
										Tp = Tc	Tc =: passé	Tc =: présent	Tc =: futur	Vc =: devoir	Vc =: pouvoir	Vc =: savoir	V-inf0 W = Ppv								
+	-	acheter	+	-	-	-	de	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	+	+	+
+	+	aller	-	-	-	-	0	-	-	-	-	-	-	+	+	+	-	-	-	-	-	-	-	-	-

Figure 1 – Beginning of table 1 Gross (1975)

The columns are organised relatively to a three structural elements. The first one is a horizontal *cartouche* which designates the part of the construction concerned by the properties it dominates. For instance, the first two columns of figure 1 are dominated by a cartouche Sujet which indicates that they specify the construction subject. Notice that the information brought by the cartouches is redundant with the formulation of some properties. For instance, the property N_0 =: N_{hum}already indicates that it specifies the subject. Actually, in the printed version of the tables, the properties are formulated without prefixes (e.g. in Gross (1975) column 1 is only headed by N_{hum}). The prefixes (N_0 =: ; N_1 =:) have been added in the electronic version in order to partially compensate the loss of information due to the complete removal of their structural informations.

The second structural element indicates that some properties depend on another one. The *dependence* between properties is graphically represented by embedding the dependent columns into the one which controls them. The last three columns of table 1 are an example of such structure which indicates that the last two are distributional properties specifying the direct object of the transformation N_0 U N_1.

The third form of structuration is the *disjunction* of columns. Disjunctions only appear as controllers and are used to indicate that some properties depend on several other ones. This kind of structure is illustrated by the columns 9 and 10 of table 1. Gross 1975 uses disjunctions as abbreviations in order avoid duplicating the dependent properties. In other words, the columns in the right part of figure 2 can be substituted for the ones in the left part.

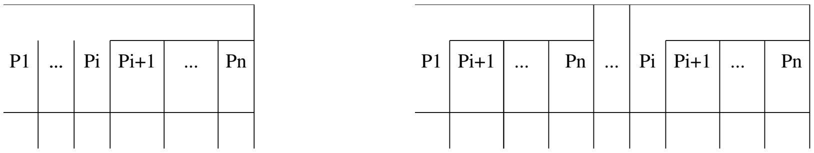

Figure 2 – Distribution of a disjunction of controllers

2.2 Interpretation

In addition to their explicit typographic structure, the tables have also an underlying structure which specifies their interpretation and the one of the properties they contain. This interpretation is jointly defined by the properties types (distributional *vs.* transformational) and by the explicit structure of the table.

As can be seen in figure 1, the base construction of a table is not a property of that table. In order to interpret the tables in a more regular way, they have been completed with their base construction. A base construction is a transformation which all other properties of the table depend on. It should then be added ahead of the table as a controller of all the tables columns. All lines of this columns are + since, by definition, all the verbs of the table have this property as their base construction.

The tables interpretation can be described by means of a set of principles. Let us first introduce the notion of *reference construction.* Any property used in a table must be interpreted w.r.t. some construction which plays the role of reference for this property. The reference construction of a distributional property is the construction which contains the constituent specified by the property. For instance, property $\mathtt{N}_1$ =: $\mathtt{N}_{hum}$ (column 20 of table 1) has the transformation $\mathtt{N}_0$ U Prp $\mathtt{N}_1$ (column 19) as reference construction. More generally, the reference construction of a partial transformation is the first transformation which it depends on. Since feature specifications do not completely specify constituents, their reference construction must take into account all constituent specifications which contol it. As for transformations, their reference construction is the base construction which they are derived from. For instance, the reference construction of $\mathtt{N}_1$ = Ppv in column 22 is the construction in column 19. Formally, the reference construction of a transformation P_1 is the first transformation P_2 which controls P_1 and which is modified by all the distributional properties controlled by P_2 and controlling P_1.

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 pronominalisation. The specified constituent is either given by the first cartouche dominating the property or inherited from the first constituent specification which controls it. Notice that, in the electronic version, the relevant constituent may be identified from the property prefix. In the printed version, this determination may be quite difficult. For instance, the fact that the last two columns of table 1 specify the complement $\mathtt{N}_1$ of the transformation in column 23 can only be deduced from the fact that they cannot concern the verb U and that the properties of the subject $\mathtt{N}_0$ are inherited from the base construction of the table, namely the properties given in columns 1 and 2.

The relations that hold between the properties that compose a table are now explicit : each transformation can be linked with its base and each distributional property linked with the relevant constituent of its reference construction. The columns of a table can be reorganised as a set of triplets :

(1) $\langle T, \{F^0, \dots, F^q\}, \{\langle C_o, \{F_0^0, \dots, F_0^{q_0}\}\rangle, \dots, \langle C_n, \{F_n^0, \dots, F_n^{q_n}\}\rangle, \}\rangle$

where T ranges over the set of the table transformations (including the base construction), $\{F^0, \dots, F^q\}$ is the set of the feature specifications linked with T, C_i $(1 \leq i \leq n)$ is a constituent specification linked with T and $\{F_i^0, \dots, F_i^{q_i}\}$ is the set of the feature specifications linked with C_i. The sets of distributional properties $\{F_k^0, \ldots, F_k^{q_i}\}$ and $\{C_o, \ldots, C_n\}$ can be interpreted as logical formulas :

$$(2)\ \bigvee_{c=c_1}^{c_k} \left(\bigwedge_{i=0}^{q_n} D_c^i \right)$$

where $c_1, \ldots, c_k$ are the reference construction constituents and $\{D_c^0, \ldots, D_c^{q_c}\}$ the subset of the properties which concern constituent c.

3 Intermediate lexicon

The construction of a set of lexical entries from a table consists in : making explicit its underlying structure ; associating formal representations with its properties, namely PATR-II constraint systems ; completing the constructions representations by exploiting the underlying structure. The intermediate lexicon associates a set of fully specified construction representations with each entry (i.e. each verb of the table). From an operational point of view, the construction of the intermediate lexicon is made in successive passes in order to have a more robust translator and to distinguish the different levels and dimensions of the translation.

3.1 Representation choices

An examination of the constructions used as transformational properties shows that they all are composed of a subject, a verb and complements whose number ranges between 0 and 3. We choose to use *canonical representations* Günthner (1988) for these constructions, namely to consider that all of them are of the following form :

(3) `Sujet Verbe` $\texttt{Compl}_1$ $\texttt{Compl}_2$ $\texttt{Compl}_3$

where `Verbe` may be a verbal segment possibly containing an auxiliary or a reflexive pronoun and where $\texttt{Compl}_1$, $\texttt{Compl}_2$ and $\texttt{Compl}_3$ 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 by a 5 parts structure, each part describing one constituent.

We also choose not to make any hypothesis on the internal structure of the constituents ; first, we want the intermediate lexicon to be a formalisation 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 be shared by these theories. In particular, we consider the prepositions, the subordinations, the operator nouns (e.g. *`le fait`* `que P`) and the demonstrative (*`ce`* `que P`) as markers directly attached to the constituent root.

3.2 Implementation

The first stage in the construction of the intermediate lexicon consists in restoring the explicite structure in the tables. We have separated there headings and SGML marked them. This processing have been done manually. The following passes are fully automated.

The second stage translates the SGML marked headings and the tables lines into PROLOG terms by means of a PERL filter. Then the dependences with disjunctions are splited in series of dependences with single controllers (cf. figure 2). The following pass determines the type of each table property by shallowly parsing of the columns headings. For instance, a property is a constituent specification if the heading starts with a prefix made of a position descriptor (ex. $\texttt{N}_0$) followed by `=:`. The fifth stage assigns identification numbers to transformational properties ; it also determines the number of the base construction of each transformation and that of the reference construction of distributional properties.

Then, the actual parsing of the headings takes place. At this stage, the values (`+`/`-`) of the individual entries are taken into account : the processing stage is repeated for each entry of the table. For each property, the parser determines which controller it depends on, namely, the type (construction or constituent) of the controller, its number if it is a construction or the number of the construction it is part of if it is a constituent. It determines which constituent of the reference construction is specified, if necessary. The parser also computes the PATR-II representation of the properties.

The following stages perform the inheritance by the derived transformations of their constituents properties. First, the table columns are reorganised according to the underlying structure. The table becomes a set of couples as described in (1). 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 in order to take into account the effects of the transformation. For instance, `N`$_2$ `=: V`^{1} Ω becomes `N`$_2$ `=: V`^{0} Ω when inherited by a passive construction.

4 Generating the ALEP lexicon

4.1 Principle

In the output of the first component, the entry of a verb is a list L of Prolog predicates. Each predicate describes, by means a set of features, a possible verb construction (and the requiered specifications), and indicates by `+` or `-` whether the construction is realizable. In L, the features are distributed according to their source : the (T$_i$) transformation (or the T$_{base}$ *base construction*) or the (F$_i$) corresponding sets of features :

(4) `L = [base_constr(...,cons(T`$_{base}$`),[F`$_{base}$`]),...,transformation(...,cons(T`$_i$`),[F`$_i$`]),...]`

A new element in the ALEP lexicon is created when the Prolog predicate is labelled with a `+` sign : `-` signed sets of features are ignored. Both the *base construction* and the transformations are converted as ALEP lexical entries. This excludes lexical rules from the grammar, since LRs are in charge of lexical transformation and thus become useless. Section 5 discusses the consequences the multiplication of entries entails for an ALEP grammar.

Applying this principle amounts to realize the following steps :

1. A single program converts (after linguistic filtering, see section 4.2) the `+` signed predicates corresponding to both the *base construction*, and the various transformations.
2. For each `+` signed transformation, features are extracted jointly >from T$_i$ and F$_i$.
3. These features are sorted (doubles are suppressed), and are grouped together according to the category (`np`, `vp`, `cp`, ...) and the (grammatical) function (`sujet`, `compl1`, `compl2`, ...).
4. Besides, the data describing a relation between a (same) constituent and (several) syntactic categories are factorized as a single feature. This new distribution is used to synthesize within a single ALEP linguistic description (LD), and thus into a single entry, the different syntactic realizations of each constituent (cf. sections 4.3.1 et 4.3.2).
5. With these two sets of features, a lexical ALEP structure schema (cf. section 4.3.3, figure 8) is progressively specified by substituting, with the appropriate LDs, the positions in the schema corresponding to the verb HEAD feature, the content of the SUBJ list, and the content of the COMPLS list.
6. The verb arg-structure (i.e. the CONTENT value) is provided jointly by the verb lemma and the table number. The resulting partial LD (PLD) holds for the *base construction* and the transformations (cf. section 4.3.2). Therefore, it is reused for the construction of each entry built from L.

4.2 Linguistic Problems

The intermediate lexicon aims at returning all the properties encoded in the Lexion-Grammar, in a format where the constructions representation is automatically readable and exploitable. But among these properties :

1. Some are not reusable in a unification-based lexicon,
2. others have to undergo linguistic adaptations to become compatible with the output format,
3. the other ones are "directly" encodable after filtering, according to the sorting mentioned in the points 3 and 4 of section 4.1.

The reasons explaining points 1 and 2 are twofold, both being related to the target formalism :

— Transformations such as interrogation : `Où N0 V-il ?` (*base construction* = `N0 Vmvt Vinf W`, table 2) are not foreseen in the target underlying theory.

— Some transformations are linguistically wrong in the frame of the target formalism : so, adjoining an additional argument (`auprès de Nhum`) for some verbs in table 4 (i.e. divalent) is in contradiction with the verb arg-structure.

In both cases we have chosen not to represent the corresponding features in the output lexicon.

4.3 Detailed Program

The program is made up with 2 parts. First a LEX/YACC grammar extracts the sets of `+` signed features (<path><operator><value>) from the intermediate lexicon, as shown in figure 4 which illustrates the migration of *assommer (stun)* (table 4). The result is a file (F) :

```
col(0,c(-1),+,construction,cons(['C.sujet.cat'=cp,'C.sujet.sub.cat'=sub, 'C.sujet.sub.form'=que,'C.sujet.struct'=['QuP']]
['C.compl1.cat'=np,'C.compl1.struct'=['N1']],['C.compl2'=nil],['C.compl3'=nil])),
[col(2,a,+,sujet,['C.sujet.cat'=np,'C.sujet.sem.sort'=hum, '.sujet.struct'=['Nhum']]),
col(3,a,+,sujet,['C.sujet.cat'=np,'C.sujet.sem.sort'= *, 'C.sujet.struct'=['Nnr']]),
col(4,a,+,sujet,['C.sujet.cat'=cp,'C.sujet.nom_op.cat'=np,'C.sujet.nom_op.type'=oper,'C.sujet.nom_op.form'='le   fait',
'C.sujet.sub.cat'=sub,
'C.sujet.sub.form'=que,'C.sujet.mode'=indic,'C.sujet.struct'=[le,fait,que,'P']])
col(12,a,+,compl1,['C.compl1.cat'=np,'C.compl1.sem.sort'=hum,'C.compl1.struct'=['Nhum']]),
col(13,a,+,compl1,['C.compl1.cat'=np,'C.compl1.sem.sort'=nhum,'C.compl1.struct'=['N-hum']]),
col(14,a,-,compl1,['C.compl1.cat'=cp,'C.compl1.nom_op.cat'=np,'C.compl1.nom_op.type'=oper,
'C.compl1.nom_op.form'='le fait','C.compl1.sub.cat'=sub,
'C.compl1.sub.form'=que,'C.compl1.mode'=indic,'C.compl1.struct'=[le,fait,que,'P']])])
```

Input : Intermediate lexicon

⇓

```
cons :'C.sujet.cat'="cp" :'C.sujet.sub.cat'="sub" :'C.sujet.sub.form'="que" : 'C.sujet.struct'="['QuP']"
cons :'C.compl1.cat'="np" :'C.compl1.struct'="['N1']" - cons :'C.compl2'="nil" - cons :'C.compl3'="nil"
cons :'C.sujet.cat'="np" :'C.sujet.sem.sort'="hum" :'C.sujet.struct'="['Nhum']"
cons :'C.sujet.cat'="np" :'C.sujet.sem.sort'="*" :'C.sujet.struct'="['Nnr']"
cons :'C.sujet.cat'="cp" :'C.sujet.nom_op.cat'="np" :'C.sujet.nom_op.type'="oper" :
'C.sujet.nom_op.form'="le fait" : 'C.sujet.sub.cat'="sub" :'C.sujet.sub.form'="que" :
'C.sujet.mode'="indic" :'C.sujet.struct'="[le,fait,que,'P']"
cons :'C.sujet.cat'="cp" :'C.sujet.sub.cat'="sub" :'C.sujet.sub.form'="que" :'C.sujet.struct'="['Qu P']"
cons :'C.compl1.cat'="np" :'C.compl1.sem.sort'="hum" :'C.compl1.struct'="['Nhum']"
cons :'C.compl1.cat'="np" :'C.compl1.sem.sort'="nhum" :'C.compl1.struct'="['N-hum']"
```

Output : file F

FIGURE 3 – Input and output of the LEX/YACC grammar

F is then given as input to a PERL program which (1) distributes the data over two main arrays, (2) translates the relevant features in ALEP PLDs, (3) return the ALEP lexical entry obtained by grouping together the PLDs.

4.3.1 Data Organisation

As shown in figure (3), information in F may be duplicated. Moreover, features describing a given constituent (e.g. `C.sujet.cat=np`) appear to be discontinuous.

The program reduces duplications and sorts the data in order to classify them according to grammatical functions : the result is the associative array **%propcat**, partially illustrated for F by figure (5) (key/val pairs are separated by `'=>'`) :

(5)

```
compl1.cat="np" =>   compl1.cat="np" :compl1.sem.sort="hum" :
                     compl1.sem.sort="nhum" :compl1.struct="[N-hum]" :
                     compl1.struct="[N1]" :compl1.struct="[Nhum]"
compl2="nil" =>      compl2="nil"
compl3="nil" =>      compl3="nil"
sujet.cat="cp" =>    sujet.cat="cp" :sujet.mode="indic" :sujet.nom_op.cat="np" :
                     sujet.nom_op.form="le fait" :sujet.nom_op.type="oper" :
                     sujet.struct="[QuP]" :sujet.struct="[le,fait,que,P]" :
                     sujet.sub.cat="sub" :sujet.sub.form="que"
sujet.cat="np" =>    sujet.cat="np" :sujet.sem.sort="*" :sujet.sem.sort="hum" :
                     sujet.struct="[Nhum]" :sujet.struct="[Nnr]"
```

At the same time, the array **%syntagme** performs the association between a key, whose value is a constituent identifier : `C.sujet.cat, C.comp`$_i$`.cat`, and the set of values which encode the possible syntactic realisations for the key :

(6)

```
compl1.cat => "np"          sujet.cat => "cp" :"np"
```

4.3.2 Partial Linguistic Descriptions

Building the ALEP lexical entry is a recursive process. At top-level, the function **&construction** synthetizes a generic structure by means of the verb lemma and the table number [1] :

(7)

$$ {}_{\text{synsem}}\left[\begin{array}{ll} \text{syn} & {}_{\text{syn}}\left[\begin{array}{ll} \text{head} & {}_{\text{vhead}}\left[\text{\$VAR\_VHEAD}\right] \\ \text{subj} & \left\langle {}_{\text{synsem}}\left[\begin{array}{ll}\text{sem} & \boxed{S}\\ \text{syn} & {}_{\text{syn}}\left[\begin{array}{ll}\text{cohead} & \text{\$VAR\_SUBJ\_COHEAD}\end{array}\right]\end{array}\right]\right\rangle \\ \text{compl} & \left\langle \text{\$VAR\_COMPL\_LIST}\right\rangle\end{array}\right] \\ \text{sem} & {}_{\text{sem}}\left[\begin{array}{ll}\text{content} & {}_{\text{arg2\_psoa}}\left[\begin{array}{ll}\text{rel} & \textsf{assommer}\\ \text{arg1} & \boxed{S}\\ \text{arg2} & \boxed{C1}\end{array}\right]\end{array}\right]\end{array}\right] $$

Symbols `$VAR_VHEAD, $VAR_SUBJ_COHEAD, $VAR_COMPL_LIST` are PERL variables whose value is computed according to **%propcat** and **%syntagme**. So, `$VAR_SUBJ_COHEAD` is instanciated from the **%syntagme** key/value pair `(sujet.cat/"cp":"np")` and from the values of the keys `sujet.cat="cp"` and `sujet.cat="np"` in **%propcat**.

4.3.3 Output

The result of this PLD constructors combination is a lexical entry specifying (7). Default values are automatically assigned to those variables which have been left unspecified according to the input data. Figure (8) provides a schematized and partial representation of the output for *assommer*.

[1] The entries are produced in the ALEP format. The figures are given in AVM notation only for sake of readability.

(8) $${}_{\text{synsem}}\!\left[\begin{array}{l}
\text{syn}\,|\,\text{subj}\left\langle {}_{\text{synsem}}\!\left[\text{syn}\ {}_{\text{syn}}\!\left[\text{cohead}\ {}_{\text{coh}}\!\left[\begin{array}{l}
\text{cp\_str}\ {}_{\text{synsem}}\!\left[\text{syn}\ {}_{\text{syn}}\!\left[\begin{array}{l}\text{subj}\ \langle\rangle\\ \text{compl}\ \langle\rangle\\ \text{head}\ {}_{\text{vhead}}\!\left[\text{mode indic}\right]\\ \text{mark}\ (\text{que} \vee \text{le\_fait\_que})\end{array}\right]\right]\\
\text{np\_str}\ {}_{\text{synsem}}\!\left[\begin{array}{l}\text{syn}\ {}_{\text{syn}}\!\left[\begin{array}{l}\text{subj}\ \langle\rangle\\ \text{compl}\ \langle\rangle\\ \text{head}\ {}_{\text{nhead}}\!\left[\ \right]\\ \text{mark nil}\end{array}\right]\\ \text{sem}\ {}_{\text{sem}}\!\left[\text{content}\ {}_{\text{rd\_cont}}\!\left[\text{rel}\ {}_{\text{rel}}\!\left[\text{rel\_sort}\ (\text{hum} \vee *)\right]\right]\right]\end{array}\right]\\
\text{pp\_str no}\\
\text{ap\_str no}\\
\dots
\end{array}\right]\right]\right]\right\rangle, \dots\\
\text{sem} \dots
\end{array}\right]$$

5 Results and Consequences for ALEP

So far, we have used our program to migrate lexica from 2 of the 61 tables. Extending the converter, in order to cover all the Lexion-Grammar, essentially amounts to (a) provide the columns tagging for the new tables, and (b) improve the PLD constructor by interpreting the corresponding new lexical properties (if any).

Our current model has produced the following results : (i) Some 700 verbs have been processed, (ii) The migration of each of them entails in the average the generation of 3 ALEP entries, according to the number of + signed transformations. In other words, the output is a lexicon made up with 2000 elements in the ALEP format, whose syntax obeys the type system defined for the LS-GRAM French Grammar (cf. Heyd et al. (1996)), (iii) Extending the model to the whole Lexion-Grammar will probably produce an ALEP lexicon with circa 30 000 entries.

This leads to two remarks : (1) We have a device which allows the ALEP grammar developper to have at his disposal a lexicon covering all the syntactic descriptions for French verbs, i.e. a crucial data for large-scale NLP applications on real texts, which is the ambition of ALEP-written linguistic resources ; (2) On the other hand, migrating the 61 tables will generate a huge lexicon, because of the number of entries, and because of the large amount of information carried by each entry. Moreover, it is likely that a great number of lexical entries will reflect "natural" lexical ambiguities (i.e. homographic forms with different reading). In addition, one have to take into account ambiguities generated by the static representation of transformations.

Now, ALEP has at the moment two characteristics which are incompatible with these two points : (1) such as any non deterministic Prolog-based formalisms, it is allergic to lexical ambiguities ; (2) it does not include so far a sophisticated and efficient DBMS.

In many fields, ALEP cannot compete, so far : it is gluttonous in terms of computing resources, it does not (yet) run on PCs, it is rather complex to teach and to use. Therefore, I think that the winning card ALEP still owns, is its reputation of a system for 'Large-Scale Grammar Development' - which is connected to large lexica : that's why ALEP must be improved in order to include a high-performance system for the lexical DBM in order to exploit lexica with the size of the ones we are able to build automatically.

Références

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

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

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

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

Günthner, F. (1988). Features and values 1988. Technical Report SNS-Bericht 88-40, Tübingen University, Tübingen.

Heyd, S., Jacquey, É., and Namer, F. (1996). E-ls-gram lingware development documentation. core grammar and extentions for French. lre 61029. Technical report, TALANA, Nancy.