Génération (semi-) automatique de ressources lexicales réutilisables à grande échelle : Conversion des tables du LADL

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

Résumé

Cette communication présente un système réalisant la migration (semi-) automatique d'un Lexique-Grammaire de 10 600 verbes sous une forme réutilisable pour différentes applications TALN. Notre convertisseur exploite le Lexique-Grammaire des verbes du LADL (disponible publiquement) qui exprime les propriétés syntaxiques de ceux-ci sous formes de tables.

Le convertisseur est constitué de deux composants de base : Le premier composant construit, pour un verbe donné, une représentation intermédiaire, i.e. l'ensemble R de constructions légitimes décrites dans le formalisme PATR-II. Le second composant utilise R pour générer une ou des entrées lexicales dans le format d'une application TALN cible.

Le convertisseur est à l'heure actuelle réalisé sous forme de maquette. Il a permis de traduire les tables 1 et 4 de (Gross, 1975) en un lexique au format HPSG.

1 Introduction

L'un des principaux obstacles au développement de systèmes de TALN ayant une couverture du français suffisante est l'absence de lexiques informatiques de grande taille. La migration de ressources lexicales constitue un moyen efficace et peu coûteux de construire de tels lexiques. Cette article présente les résultats d'une expérience de portage des tables de verbes du Lexique-Grammaire, conçu et réalisé au LADL (Gross, 1975), dans un format adapté au TALN basé sur les grammaires d'unification lexicalisées. Malgré leur richesse (61 tables décrivent de manière fine le comportement syntaxique de plus de 10 000 verbes) et le fait qu'elles soient disponibles publiquement,[1] ces données ne sont pas, à l'heure actuelle, utilisées par la communauté informatique linguistique du fait de la difficulté de leur interprétation. Leur exploitation informatique se réduit, à notre connaissance, à une seule expérience (Alcouffe, Revellin Falcoz, & Zaysser, 1993). Notre expérience, réalisée à ce jour sous forme de maquette, permet de convertir les tables du Lexique-Grammaire en un lexique PATR-II qui sert de pivot pour construire des lexiques destinés à des systèmes de TALN pour le français, inspirés de modèles théoriques comme HPSG, LFG ou TAG.

Dans la section 2, nous décrivons la structure des tables du Lexique-Grammaire et leur interprétation. Nous présentons ensuite, dans la section 3, la construction du lexique intermédiaire. Puis, nous abordons en section 4 la conversion des entrées du lexique intermédiaire en entrées d'un lexique HPSG. Enfin, la section 5 présente les résultats obtenus et les perspectives futures.

2 Les tables

Le Lexique-Grammaire décrit les principaux aspects du comportement syntaxique des prédicats du français sous forme de tables à valeurs binaires (+/-). Cette description comprend les propriétés intrinsèques de ces prédicats, leurs constructions légitimes ainsi que les propriétés des constituants qui y apparaissent.

2.1 Description

Le Lexique-Grammaire décrit plus de 10 000 prédicats verbaux à l'aide de 61 tables. Chaque table correspond à une classe de verbes ; elle est identifiée par un numéro et une *construction de base*. Les constructions sont décrites sous forme d'expressions linéaires composées de symboles prédéfinis. Par exemple, la table 1 (figure 1) regroupe les verbes opérateurs U dont la construction de base est N_0 U Prép V^0 Ω dans laquelle U a un sujet N_0 et un complément propositionnel infinitif V^0 Ω ayant N_0 pour sujet et dont les compléments éventuels Ω ne sont pas détaillés ; le complément de U est introduit par une préposition Prép. Les 61 tables se divisent en quatre groupes : 19 concernent les verbes dont les arguments peuvent être des complétives (Gross, 1975) ; 17 décrivent les verbes transitifs (Boons, Guillet, & Leclère, 1976a) ; 9 sont relatives aux les verbes intransitifs (Boons, Guillet, & Leclère, 1976b) ; 16 sont consacrés à la description des verbes transitifs locatifs (Guillet & Leclère, 1992).

1. Les tables du Lexique-Grammaire sont accessibles publiquement à l'URL suivante :
http://www-ceril.univ-mlv.fr/LexiqueGrammaire/

Typologie des propriétés. Chaque table décrit une classe de verbes à l'aide d'un ensemble de propriétés caractéristiques de celle-ci. Les propriétés des verbes, décrites dans les colonnes de la table, sont de trois types. Les constructions, également appelées « propriétés transformationnelles » (Gross, 1975), représentent les principales variations du cadre syntaxique du verbe (ex. passivation, optionnalité des compléments, etc.) Les *spécifications de constituant* (également appelées « transformations partielles ») décrivent complètement un constituant d'une construction (ex. N_0 =: N_{hum} indique que le sujet de la construction peut être un nom humain). Enfin, les *spécification de trait*, comme Aux = avoir (le verbe se conjugue avec l'auxiliaire *avoir*), spécifient seulement l'un des trait d'un constituant. Les spécifications de constituant et de trait sont également appelées « propriétés distributionnelles ».

La figure 1 illustre ces trois types de propriétés: la colonne 1 est une spécification du constituant sujet N_0 de la construction de base ; la colonne 4 spécifie le trait *auxiliaire* du verbe principal U ; la colonne 7 (N_0 U) est une transformation qui indique que l'infinitive complément est optionnel.

Structure des tables. Les colonnes des tables sont organisées à l'aide de trois éléments de structure. Les *cartouches horizontales* indiquent quel constituant est concerné par les propriétés qu'elle domine. Par exemple, les deux premières colonnes de la table 1 sont dominées par une cartouche Sujet qui indique que les propriétés qu'elles contiennent spécifient le sujet de la construction de base. On peut noter que l'information apportée par les cartouches est déjà présente dans la formulation des propriétés qu'elles dominent. Par exemple, N_0 =: N_{hum} indique déjà qu'elle spécifie le sujet. En réalité, dans leur version imprimée, les entêtes des tables ne contiennent pas de préfixes (ex. dans (Gross, 1975), la colonne 1 de la table 1 a pour entête N_{hum}). Les préfixes ont été rajoutés dans la version électronique pour compenser partiellement la suppression de toutes les informations structurelles des tables.

Le second élément de structure indique les *dépendances* entre propriétés. Elles sont représentées typographiquement par l'enchâssement des colonnes dépendantes à l'intérieur de celles qui les « contrôlent ». Les trois dernières colonnes de la table 1 sont une instance de ce paradigme qui indique que les deux dernières propriétés concernent le complément d'objet direct de la construction N_0 U N_1.

Le troisième élément de structuration est la *disjonction* de colonnes. Les disjonctions sont utilisées pour indiquer qu'une ou plusieurs propriétés dépendent de plusieurs autres. Cette forme de structuration est illustrées par les colonnes 9 et 10 de la table 1. Comme l'indique (Gross, 1975), les disjonctions sont, du point de vue formel, des abréviations qui permettent de ne pas dupliquer les colonnes dépendantes. Shématiquement d'autres termes, les colonnes de la partie gauche de la figure 2 peuvent être remplacées par celles de droite.

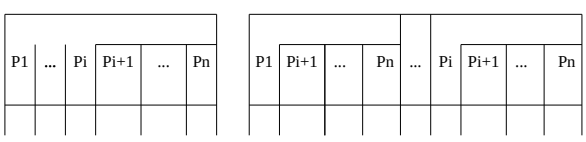

FIG. 2 – *Distribution d'une disjonction de contrôleurs*

2.2 Interprétation

Les tables ont, outre leur structure typographique, une structure sous-jacente qui spécifie l'interprétation des propriétés qu'elles contiennent. Cette spécification est définie conjointement par la typologie des propriétés et par la structure des tables.

Construction de base. On constate, par exemple sur la table 1, que la construction de base d'une table n'apparaît comme une propriété de celle-ci. Afin de rendre l'interprétation des tables plus régulière, nous avons rajouté à chacune d'elle sa construction de base. Cette construction est une propriété dont toutes les autres dépendent et doit donc être ajoutée comme un contrôleur de toutes les colonnes de la table. Par ailleurs, toutes les cases de cette colonne ont la valeur + puisque tous les verbes de la table ont par définition cette propriété comme construction de base.

Construction de référence. Toute propriété utilisée dans une table doit, pour être interprétée, être rattachée à une *construction de référence*. Pour les propriétés distributionnelles, la construction de référence est celle qui contient le constituant spécifié par cette propriété. Par exemple, la propriété N_1 =: N_{hum} (colonne 20 de la table 1) a pour construction de référence N_0 U Prép N_1 (colonne 19). Dans le cas des transformations, la construction de référence est la base à partir de laquelle la construction est dérivée. Par exemple, la construction de référence de la transformation N_1 = Ppv est la construction de la colonne 19. De manière plus formelle, la construction de référence d'une construction P_1 est la première construction P_2 qui domine P_1, modifiée par toutes les propriétés distributionnelles qui se trouvent entre P_2 et P_1 (i.e. dominées par la première et dominant la seconde).

Constituant de référence. Pour interpréter une propriété distributionnelle (spécification de constituant ou de trait), on doit également connaître le constituant de la construction de référence concerné par cette propriété. Cette information est également nécessaire dans le cas des pronominalisations. Ce constituant peut être identifié de deux façons : par l'intermédiaire de la première cartouche qui domine la propriété ou bien par héritage, à partir de la première spécification de constituant qui contrôle cette propriété. Signalons que dans la version imprimée des tables, le constituant concerné est donné par le préfixe de la colonne. Dans la version électronique, cette identification peut être relativement complexe. Par exemple, le fait que les deux dernières colonnes de la table 1 concerne le complément N_1 de la construction N_0 U N_1 peut seulement être inféré du fait qu'elles ne peuvent être des propriétés de U et que les pro-

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

FIG. 1 – *Début de la table 1 (Gross, 1975)*

priétés de $\mathtt{N}_0$ sont héritées de celle du sujet de la construction de base (i.e. colonnes 1 et 2)

Interprétation. Les diverses relations qui s'établissent entre les diverses propriétés qui composent une table sont à présent explicites : chaque transformation peut être rattachée à sa construction de référence et chaque propriété distributionnelle au constituant concerné de sa construction de référence. Ainsi, chaque table définit un ensemble de triplets pour chacun des verbes qu'elle contient :

$$(1)\ \langle T, \{F^0, \ldots, F^q\}, \{\langle C_0, \{F_0^0, \ldots, F_0^{q_0}\}\rangle, \ldots, \langle C_n, \{F_n^0, \ldots, F_n^{q_n}\}\rangle\}\rangle$$

où T décrit l'ensemble des constructions légitimes pour le verbe, y compris la construction de base, $\{F^0, \ldots, F^q\}$ est l'ensemble des spécifications de traits qui concernent des constituants de T, C_i ($1 \leq i \leq n$) est une spécification d'un constituant c de T et $\{F_n^0, \ldots, F_n^{q_n}\}$ l'ensemble des spécifications de trait qui concernent c. Le couple $\langle T, \{F^0, \ldots, F^q\}\rangle$ peut, à son tour, être vu comme un ensemble de couples :

$$(2)\ \{\langle CT_0, \{FT_0^0, \ldots, FT_0^{r_0}\}\rangle, \ldots, \langle CT_k, \{FT_k^0, \ldots, FT_k^{r_k}\}\rangle\}$$

où CT_j est la description du j^{e} constituant de T et où $\{FT_j^0, \ldots, FT_j^{r_j}\}$ est le sous ensemble de $\{F^0, \ldots, F^q\}$ des spécifications de trait qui concerne ce constituant. En d'autres termes, les triplets (1) peuvent se réécrire comme des ensembles de couples $\langle C, \{F^0, \ldots, F^s\}\rangle$. Chaque triplet correspond à une formule logique :

$$(3)\ \bigwedge_{c=c_1}^{c_k} \left(\bigvee_{i=0}^{q_n} C_c^i \wedge \left(\bigwedge_{t=t_0}^{t_l} \left(\bigvee_{j=0}^{s_m} F_t^j \right) \right) \right)$$

où $c_0 \ldots c_k$ sont les constituants de T, $C_c^0 \ldots C_c^{q_c}$ sont les propriétés qui décrivent les réalisations possibles du constituant c, $t_0 \ldots t_l$ sont les traits utilisés pour décrire c et $F_t^0 \ldots F_t^{s_t}$ sont les propriétés qui spécifient les valeurs possibles du trait t.

3 Lexique intermédiaire

La construction d'un ensemble d'entrées lexicales à partir d'une table consiste à : expliciter la structure sous-jacente de cette table ; associer aux propriétés des représentations formelles, en l'occurrence des ensembles de contraintes PATR-II ; compléter la spécification des constructions en exploitant la structure sous-jacente. Le lexique résultat associe à chaque entrée d'une table un ensemble de représentations de constructions complètement spécifiées (i.e. complètes relativement à l'information contenue dans la table), chacune correspondant à un triplet tel que (1).

D'un point de vue opératoire, cette construction est réalisée en plusieurs étapes. Chaque étape fait l'objet d'une passe de traitement indépendante. Le traitement en passes permet d'améliorer la robustesse du traducteur et de distinguer les différents niveaux et les différentes dimensions du traitement.

3.1 Questions méthodologiques

Un examen des tables que nous souhaitons traduire montre que celles-ci sont composées d'un sujet, d'un verbe et de compléments dont le nombre varie entre 0 et 3. Nous avons choisi d'utiliser des représentations canoniques (Günthner, 1988), c'est-à-dire, de considérer que toutes les constructions sont de la forme :

(4) $\mathtt{Sujet\ Verbe\ Compl}_1\ \mathtt{Compl}_2\ \mathtt{Compl}_3$

où `Verbe` est un segment verbal comprenant éventuellement un auxiliaire ou un pronom réfléchi et où les compléments $\mathtt{Compl}_1$, $\mathtt{Compl}_2$ et $\mathtt{Compl}_3$ peuvent être absents ; cependant, si $\mathtt{Compl}_i$ est absent, alors $\mathtt{Compl}_j$ pour tout $j > i$ doit l'être aussi. On peut ainsi représenter chaque construction par une structure à 5 blocs, chaque bloc décrivant les propriétés du constituant auquel il correspond. Le bloc est vide lorsque le constituant (complément) est absent.

Par exemple, la construction $\mathtt{N}_{hum}$ `V que P` a pour représentation la structure de traits donnée en figure 3.

Par ailleurs, nous avons choisi de ne pas faire d'hypothèse particulière sur la structure interne des différents constituants : nous souhaitons que le lexique intermédiaire soit

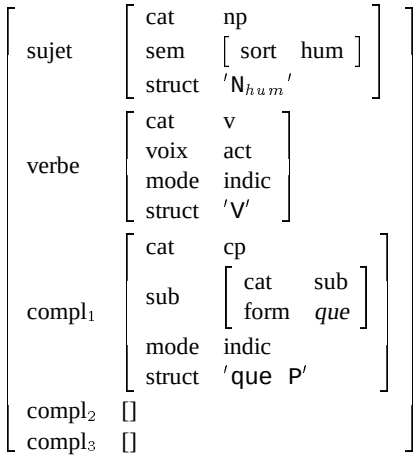

FIG. 3 – *Représentation de* N_{hum} V que P

une formalisation des tables qui reflète la position défendue par (Gross, 1975, page 46) ; d'autres parts des hypothèses théoriques fortes rendraient plus difficile la construction de lexiques cibles pour des systèmes de TALN basés sur d'autres théories linguistiques car ces hypothèses ne seraient vraisemblablement pas partagées par la plupart de ces théories. Nous faisons donc une description « à plat » des constituants. En particulier, nous considérons les prépositions, les conjonctions de subordination, les noms opérateurs (ex. *le fait* que P) et le pronom démonstratif (ce dans *ce* que P) comme des marqueurs rattachés directement à la racine du constituant.

Signalons que nous n'avons pas suivi (Gross, 1975, page 46) dans son analyse des propositions complétives comme étant des noms.

3.2 Implémentation

La première étape de la construction du lexique intermédiaire est la restauration de la structure typographique dans la version électronique des tables. Ceci est réalisé en séparant l'entête de chaque table puis en la garnissant manuellement de balises SGML. Toutes les étapes suivante de la conversion sont réalisées par programme.

Traitement de l'entête de la table. La seconde étape consiste à traduire, à l'aide d'un filtre PERL les entêtes balisées et les lignes des tables sous forme de termes PROLOG. On procède ensuite à la distribution des contrôleurs des disjonctions comme cela est schématisé en 2.1 (figure 2). Puis on détermine le type de chaque propriété en effectuant une analyse superficielle du contenu de la colonne. Par exemple, une colonne décrit une transformation partielle si elle commence par une suite de symbole correspondant à un préfixe (ex. N_0) suivie de =: ; elle décrit une pronominalisation si le préfixe est suivi de = Ppv. L'étape suivante détermine les constructions de référence des propriétés en se basant sur le typage qui vient d'être effectué et sur la structure de la table. On peut alors calculer pour chaque propriété une représentation PATR-II et déterminer les constituants concernés par les spécifications de constituant ou de trait.

Traitement des entrées de la table. Les étapes suivantes suivantes ont pour but de propager les propriétés des constituants des bases aux constituants des constructions qui en sont dérivées. Pour cela, les colonnes de la table sont réorganisées en fonction de la structure de celle-ci : la table devient un ensemble de couples tel que (1). On rajoute ensuite à chaque transformation les propriétés que les constituants qu'elle partage avec sa base ont dans celle-ci. Ces propriétés sont ensuite réanalysées afin de tenir compte des effets de la transformation. Par exemple, une transformation partielle N_2 =: V^1 Ω devient N_2 =: V^0 Ω lorsqu'elle est héritée par une transformation passive (ex. [passif *par*] dans la table 3).

4 Génération d'un lexique HPSG

4.1 Principe

L'entrée d'un verbe, à la sortie du premier composant, est une liste L de predicats PROLOG. Chaque prédicat décrit une construction possible du verbe (avec les spécifications nécessaires), et indique par '+' ou '-' si la réalisation de cette construction est attestée. Chaque prédicat PROLOG dans L distingue les traits fournis par la transformation T_i (ou la construction de base T_{base}), et l'ensemble des affectations (A_i) associées :

(5)
$$\begin{aligned} L \;=\; & [\ \text{constr\_base}(..., \text{cons}(\ T_{base}\ ), [\ A_{base}\ ]\ ),\\ & \text{transformation}(..., \text{cons}(\ T_1\ ), [\ A_1\ ]\ ),\\ & ..., \text{transformation}(..., \text{cons}(\ T_i\ ), [\ A_i\ ]\ ), ...\ ] \end{aligned}$$

Le modèle théorique HPSG s'appuie sur une très forte lexicalisation de la grammaire, ce qui explique la taille importante du lexique et la complexité des informations contenues dans chaque entrée. Celles-ci sont formulées au moyen de structures de traits typés (STTs). La syntaxe que nous avons adoptée pour la représentation finale des STTs (et donc des entrées lexicales) est celle définie dans le formalisme ALEP (Alshawi, Arnold, Backofen, J. Lindop, Netter, Pulman, Tsujii, & Uszkoreit, 1991), utilisé dans le cadre du projet LS-GRAM (Namer, Heyd, & Jacquey, 1996a) dont le but était la réalisation d'un analyseur à grande échelle du français dans un formalisme inspiré du modéle HPSG.

Le principe de la construction du lexique ALEP consiste à définir une entrée par transformation « positive » de L, et à en ignorer les transformations « négatives ». Ce principe implique qu'une transformation, quelle qu'elle soit, est une nouvelle entrée lexicale, et donc sous-entend l'absence, dans la grammaire, des règles lexicales.

L'application de ce principe revient à réaliser les étapes suivantes:

1. Chaque transformation positive (construction de base, et transformations autorisées) est convertie, moyennant un certain filtrage linguistique (cf. section 4.2) en une entrée lexicale, au moyen du même programme.

2. Pour chaque transformation positive, on ne garde que les ensembles de traits. Ceux-ci proviennent à la fois de T_i et des A_i « positifs ».

3. On regroupe les traits (après élimination des doublons et tri) par constituant (`sujet, comp1, comp2, ...`) et catégorie (`np, vp, cp, ...`); ce regroupement est indépendant de la provenance dans la transformation (C_i ou A_i) puisque les deux sources d'informations sont compatibles (cf. section 4.3.1).

4. Par ailleurs, on répertorie pour chaque constituant (autre que le verbe) l'ensemble des catégories syntaxiques autorisées. Ceci va permettre de synthétiser en un seul trait, et donc et de regrouper en une seule entrée, les différentes réalisations syntaxiques de chaque constituant (cf. sections 4.3.1 et 4.3.2).

5. Grâce à ces deux ensembles de traits, on enrichit un schéma de structure lexicale ALEP (cf. section 4.3.3, figure 7) satisfaisant un système de type préétabli, en renseignant par les descriptions linguistiques (DL) appropriées les traits identifiant les propriétés catégorielles, syntaxiques ou sémantiques du verbe.

6. La structure argumentale du verbe (i.e. en HPSG et ALEP la valeur du trait CONTENT) est donnée conjointement par le lemme du verbe et le numéro de la table. La DL partielle (DLP) qui en résulte est commune à la construction de base et aux transformations (cf. section 4.3.2). Elle est donc conservée lors de la construction de chaque entrée correspondant à L.

4.2 Problèmes linguistiques

Le lexique intermédiaire se fixe comme objectif de restituer, dans un format où la représentation des constructions, triées et ordonnées, est lisible et exploitable automatiquement, toutes les propriétés codées dans le Lexique-Grammaire. Seule une partie de ces propriétés est «directement» exploitable après filtrage, moyennant le réordonnement mentionné dans les points 3 et 4 de la section 4.1. En revanche, d'autres doivent subir des modifications pour en permettre le codage final, et enfin, certains traits ne sont pas réutilisables dans un lexique basé sur l'unification.
Les raisons qui empêchent l'exploitation de certaines descriptions linguistiques (DL), ou qui entraîne leur altération sont de deux sortes, toutes deux reliées à la nature du formalisme cible:

– Tout d'abord, certaines transformations (e.g. l'interrogative: `Où N0 V-il?` transformée de `N0 Vmvt Vinf W` dans la table 2) ne peuvent pas être modélisées en HPSG, leur codage n'étant pas prévu dans le cadre de ce formalisme.

– D'autre part, certaines transformations sont linguistiquement incorrectes, dans le cadre théorique considéré. C'est le cas de l'adjonction, pour certains verbes de la table 4 (`QuP V Nhum`, donc divalents), d'un argument supplémentaire («`auprès de Nhum`»): celui-ci est assimilable à un circonstant, et de ce fait n'est pas intégrable en HPSG dans la structure argumentale du verbe.

L'approche que nous avons suivie consiste à ignorer les traits correspondant aux types de cas énumérés ci-dessus.

```
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="nhum":sujet.sem.sort="hum":
   sujet.struct="[Nhum]":sujet.struct="[Nnr]"
```

FIG. 5 – *Tableau Associatif* **%propcat**

4.3 Détails du Programme

Le programme se déroule en deux parties. Tout d'abord une grammaire lex/yacc extrait de la structure intermédiaire les ensembles positifs de traits :

(6) `<chemin><opérateur><valeur>`

A partir de la représentation de l'exemple (5), la grammaire renvoie les traits des couples (T_i,A_i) signalés par '+'. Chaque ligne du fichier généré est le contenu d'une liste Prolog (i.e. la description d'un constituant) dont les éléments sont séparés par ':'. Une balise, en début de ligne, identifie la construction à laquelle la description appartient. La figure (4) illustre le format avant et après conversion, pour le verbe *assommer* de la table 4 du Lexique-Grammaire (`Qu P V N1`):

Ce fichier F constitue l'entrée d'un programme PERL qui distribue les données dans deux tableaux principaux, puis convertit les traits pertinents en DL partielles (DLPs) ALEP, et enfin renvoie l'entrée lexicale ALEP obtenue en regroupant les DLPs de façon appropriée.

4.3.1 Organisation des informations

La figure (4) montre que les traits dans F nécessitent une réorganisation pour pouvoir être exploités: l'information est parfois dupliquée (`'C.sujet.cat'="cp"`), sont parfois dupliqués, la description du même constituant (e.g. `C.sujet.cat="cp"`) peut être distribuée sur plusieurs lignes, certains traits peuvent apparemment se contredire

(7) a. `'C.sujet.struct'="['QuP']"`

b. `'C.sujet.struct'="[le,fait,que,'P']"`

Le programme trie les traits rendus uniques, puis les répertorie, constituant par constituant, dans le tableau associatif **%propcat**, représenté partiellement dans la figure(5) où les paires clé/valeur sont séparés par `=>`:

En parallèle, le tableau associatif **%syntagme** est construit, dont les clés identifient les constituants: `C.sujet.cat`, `C.comp`$_i$`.cat`, et dont les valeurs codent les réalisations syntaxiques possibles de chaque constituant: `np`, `vp`, `cp`...

(8)
```
compl1.cat => "np"
   sujet.cat => "cp":"np"
```

```
(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,'C.sujet.struct'=['Nhum']]),
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']])
(...)
```

Entrée : Lexique Intermédiaire

⇓

```
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'="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']"
(...)
```

Sortie: Fichier F

FIG. 4 – *Entrée et Sortie de la grammaire* LEX/YACC

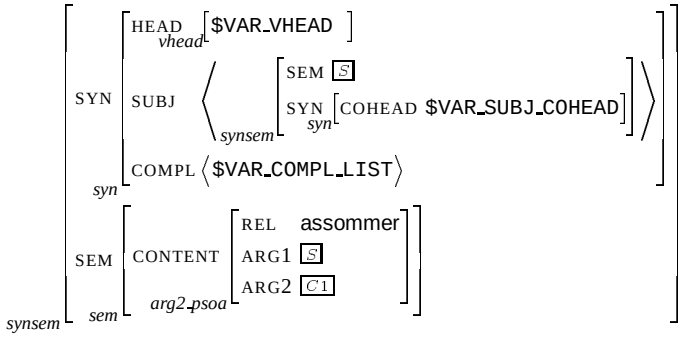

FIG. 6 – *Structure Générique*

4.3.2 Construction des DLP

Le principe de construction du signe lexical est récursif. Au niveau supérieur, la fonction **&construction** synthétise une structure générique correspondant au lemme du verbe, et au numéro de table. Celle-ci est une STT reflètant la structure des signes lexicaux de HPSG : le trait SEM|CONTENT est renseigné par le lemme et le numéro de table, alors que le trait SYN doit décrire les propriétés catégorielles (HEAD) et la complémentation (traits SUBJ et COMPL) du signe. L'interface syntaxe-sémantique est assurée par le partage des indices (S et Ci).

Les symboles `$VAR_...` sont des variables PERL dont la valeur, calculée par rapport à **%propcat** et **%syntagme**, va spécifier la structure gérique, pour HEAD, SUBJ et COMPL. Ainsi, la variable `$VAR_SUBJ_COHEAD` est calculée à partir du couple clé-valeur (`sujet.cat, "cp":"np"`) de **%syntagme**, et de la valeur des clés `sujet.cat="cp"` et `sujet.cat="np"` de **%propcat**. Le résultat est la STT valeur de l'attribut COHEAD, qui identifie les réalisations syntaxiques attesteées (ici : `np` et `cp`) du constituant concerné (ici: `sujet`), ainsi que leurs propriétés. Ce trait, apparaissant également sur chaque complément, est la base d'une approche (présentée dans (Namer, Schmidt, & Theofilidis, 1996c) et (Namer, Schmidt, & Theofilidis, 1996b)) permettant d'éviter l'utilisation de la disjonction dans la représentation de la complémentation [2].

4.3.3 Sortie du convertisseur

Le résultat de cette combinaison de constructeurs de DLPs est une entrée lexicale instanciant (6). Les variables non spécifiées sont remplacées automatiquement par des valeurs par défaut, comme l'illustre dans (7) la représentation partielle de l'entrée de *assommer* [3]:

5 Résultats et Perspectives

Nous avons jusqu'ici testé le convertisseur sur 2 des 61 tables. L'extension prévue aux autres tables va nécessiter, dans la première phase, une modification du parenthésage manuel des colonnes (cf. section 3). Dans la deuxième phase, il s'agira de coder la conversion d'éventuelles nouvelles propriétés de façon à les rendre conformes à la structure du lexique en cours de codage.

La maquette actuelle a produit les résultats suivants:

- Environ 700 verbes ont été traités,
- Pour chaque verbe, il faut multiplier en moyenne, le nombre d'entrée par 3, ceci correspondant au nombre de transformations autorisées. En d'autres termes, la sortie du convertisseur contient 2000 entrées lexicales, dont le format correspond à un système de type proche de HPSG qui a été défini et validé lors du projet

2. La disjonction est souvent génératrice d'ambiguïtés lexicales, et donc très coûteuse dans un système basé sur l'unification (PROLOG); ainsi l'approche par COHEAD permet de réduire les temps d'exécution.

3. Le symbole « ∨ » représente la disjonction booléenne de deux valeurs qui dans **%propcat** sont associées au même attribut (ex. `sujet.sem.sort` dans Figure(5)).

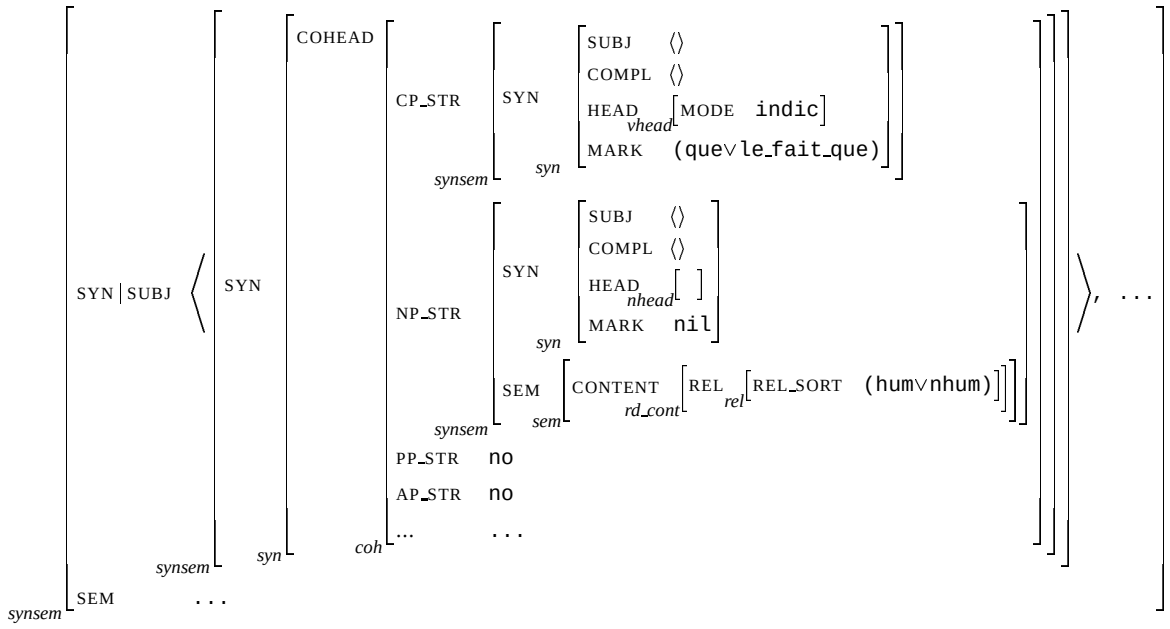

FIG. 7 – *Structure du sujet dans l'entrée de assommer*

LRE LS-GRAM (création de ressources linguistiques à grande échelle, (Namer et al., 1996a)).

– L'extension de la maquette à l'ensemble du Lexique-Grammaire des verbes du français engendrera vraisemblablement un lexique d'environ 30 000 entrées. La réutilisabilité de ce lexique est garanti par l'existence de la grammaire LS-GRAM pour l'analyse du français.

Le perspectives d'extension de ce convertisseur peuvent prendre plusieurs directions:

Il est possible de l'adapter à d'autres langues (italien, allemand, espagnol ...) pour lesquelles des recherches ont été menées en leur temps pour la réalisation d'un Lexique-Grammaire sur le modèle de celui réalisé pour le français. Bien entendu, cette hypothèse suppose que les tables soient publiquement accessibles, et disponibles sous format électronique.

Par ailleurs, HPSG n'est pas le seul modèle pour lequel il serait interessant de générer automatiquement un lexique: cette démarche concerne également la réalisation d'analyseurs inspirés d'autres théories syntaxiques basés sur la lexicalisation des grammaires, comme LFG (plusieurs implémentations existent dans le domaine public, mais aucune, à notre connaissance, ne dispose actuellement de très gros lexiques), ou TAG (l'analyseur FTAG du français (Abeille, Candito, Daille, & Robichaud, 1996) dispose désormais d'une large couverture grammaticale': l'extension du lexique permettrait de valider celle-ci sur un large corpus).

Références

Abeille, A., Candito, M.-H., Daille, B., & Robichaud, B. (1996). FTAG - Un analyseur syntaxique de phrases françaises. In *Actes de la conférence ILN'96*. Nantes.

Alcouffe, P., Revellin Falcoz, B., & Zaysser, L. (1993). Azote: Des tables du LADL au format GENELEX. In *Actes du colloque Informatiques et Langues Naturelles*, IRIN, Université de Nantes, Nantes.

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

Boons, J.-P., Guillet, A., & Leclère, C. (1976a). *La structure des phrases simples : Constructions intransitives*, Vol. 8 de *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.

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

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

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

Namer, F., Heyd, S., & Jacquey, E. (1996a). E-LS-GRAM Lingware Development Documentation - Final Report. Deliverable e-d7-frb, LRE-61029, CEC.

Namer, F., Schmidt, P., & Theofilidis, A. (1996b). Adaptation d'une théorie syntaxique au génie linguistique: le projet LS-GRAM. In *Actes de la conférence ILN'96*. Nantes.

Namer, F., Schmidt, P., & Theofilidis, A. (1996c). Lexicalism and Efficiency: Experiments with the ALEP System. In *Actes de la conférence TALN/HPSG'96*. Marseille.