Intégrer les tables du Lexique-Grammaire à un analyseur syntaxique robuste à grande échelle

Benoît Sagot[1] Elsa Tolone[2]
(1) Alpage, INRIA / Paris 7, 30 rue du Ch. des rentiers, 75013 Paris, France
(2) IGM, Université Paris-Est, 77454 Marne-la-Vallée Cedex, France
benoit.sagot@inria.fr, elsa.tolone@univ-paris-est.fr

Résumé. Dans cet article, nous montrons comment nous avons converti les tables du Lexique-Grammaire en un format TAL, celui du lexique Le*fff*, permettant ainsi son intégration dans l'analyseur syntaxique FRMG. Nous présentons les fondements linguistiques de ce processus de conversion et le lexique obtenu. Nous validons le lexique obtenu en évaluant l'analyseur syntaxique FRMG sur le corpus de référence de la campagne EASy selon qu'il utilise les entrées verbales du Le*fff* ou celles des tables des verbes du Lexique-Grammaire ainsi converties.

Abstract. In this paper, we describe how we converted the lexicon-grammar tables into an NLP format, that of the Le*fff* lexicon, which allowed us to integrate it into the FRMG parser. We decribe the linguistic basis of this conversion process, and the resulting lexicon. We validate the resulting lexicon by evaluating the FRMG parser on the EASy reference corpus depending on the set of verbal entries it relies on, namely those of the Le*fff* or those of the converted lexicon-grammar verb tables.

Mots-clés : Lexiques syntaxiques, Lexique-Grammaire, analyse syntaxique.

Keywords: Syntactic lexica, Lexicon-Grammar, parsing.

1 Introduction

Les tables du Lexique-Grammaire constituent aujourd'hui une des principales sources d'informations lexicales syntaxiques pour le français. Leur développement a été initié dès les années 1970 par Maurice Gross, au sein du LADL puis de l'IGM (Université Paris-Est) (Gross, 1975; Boons *et al.*, 1976; Guillet & Leclère, 1992). Ces informations se présentent sous la forme de *tables*. Chaque table regroupe les éléments d'une catégorie donnée partageant certaines *propriétés définitoires*, qui relèvent généralement de la sous-catégorisation. Ces éléments forment une *classe*. Une table se présente sous forme de matrice : en lignes, les éléments lexicaux de la classe correspondante ; en colonnes, les propriétés qui ne sont pas forcément respectées par tous les éléments de la classe ; à la croisée d'une ligne et d'une colonne le signe $+$ ou $-$ selon que l'entrée lexicale décrite par la ligne accepte ou non la propriété décrite par la colonne.

Les tables actuelles souffrent de diverses formes d'incohérence et d'incomplétude. En particulier, les propriétés définitoires ne sont pas représentées dans les tables[1]. Pour y remédier, des

[1]Ceci a également motivé les travaux de (Gardent *et al.*, 2005). Nous renvoyons à (Constant & Tolone, 2008) pour une comparaison entre la version texte des tables, utilisée ici, et le travail présenté par (Gardent *et al.*, 2005).

tables des classes sont en développement à l'IGM pour chaque catégorie, qui associent à chaque classe l'ensemble de ses propriétés définitoires (Paumier, 2003). C'est en particulier le cas des verbes simples, pour lesquels une table des classes est développée actuellement, en complément des 61 tables correspondantes.

Les résultats préliminaires de ce travail de fond nous ont permis d'envisager l'utilisation des tables du Lexique-Grammaire des verbes simples dans un analyseur syntaxique à grande échelle, l'analyseur FRMG (Thomasset & de la Clergerie, 2005). Mais une telle intégration suppose d'interpréter et de convertir les concepts linguistiques et les données lexicales des tables du Lexique-Grammaire dans le formalisme lexical utilisé par FRMG, celui du lexique syntaxique Le*fff* (Sagot *et al.*, 2006). C'est ce travail que nous décrivons dans cet article. Le résultat en est un lexique au format Le*fff* encodant aussi bien que possible les données lexicales des tables. Nous validons notre processus de conversion par l'évaluation préliminaire des performances de FRMG lorsqu'il est couplé à ce lexique.

L'article est organisé comme suit. La section 2 décrit le lexique *lglex* construit grâce à la table des classes des verbes à partir des différentes tables. La section 3 décrit le format du lexique syntaxique Le*fff*. La section 4 présente les fondements linguistiques et la méthodologie pratique de conversion de *lglex* au format Le*fff* et le lexique obtenu. La section 5 compare les performances de l'analyseur FRMG selon qu'il utilise les entrées verbales du Le*fff* ou de *lglex*.

2 Le lexique verbal *lglex*

Une table des classes regroupe en colonnes l'ensemble de toutes les propriétés syntaxiques répertoriées pour la catégorie concernée, et liste en lignes l'ensemble des classes définies pour cette même catégorie. À l'intersection d'une ligne et d'une colonne, le symbole $+$ (resp. $-$) indique que la propriété correspondante est vérifiée (resp. non vérifiée) par tous les éléments de la classe (c'est-à-dire par toutes les entrées de la table correspondante). Le symbole o indique que la propriété est explicitement codée dans la table concernée, car elle est vérifiée par certaines de ses entrées mais pas toutes. Enfin, le symbole $?$ indique une cellule non encore renseignée.

Le développement de la table des classes des verbes est bien avancé (Constant & Tolone, 2008). Toutes les propriétés définitoires des classes de verbes y ont été codées, par des $+$ et des $-$, rendant ainsi explicites les informations sous-entendues dans les tables elles-mêmes. Toutes les cellules correspondant aux propriétés codées dans les tables sont codées o. Seules les propriétés encore non étudiées pour certaines classes de verbes peuvent faire l'objet d'un codage par $?$.

Grâce à ces travaux de mise en cohérence et d'explicitation des propriétés syntaxiques des verbes dans les tables du Lexique-Grammaire, il a été possible de construire une version structurée des tables du Lexique-Grammaire, disponible en un format textuel et en un format XML, et appelé lexique *lglex* (Constant & Tolone, 2008)[2]. La sous-partie de ce lexique qui reproduit les entrées verbales des tables librement distribuées est elle-même librement distribuée[3].

La construction de *lglex* repose sur l'outil *LGExtract*, qui prend en entrée les tables d'une catégorie donnée, la table des classes de cette catégorie, et un fichier de configuration. Ce fichier

[2]Certains des problèmes ainsi résolus ont d'ailleurs été détectés grâce à l'exploitation de *lglex* au cours du travail décrit ici.

[3]Distribution partielle sous license LGPL-LR à l'adresse `http://infolingu.univ-mlv.fr`, Données Linguistiques > Lexique-Grammaire > Visualisation

définit comment chaque propriété (issue de la table des classes ou, dans le cas de propriétés qui y sont codées *o*, de la table correspondante) contribue à la construction de l'entrée *lglex*.

Dans le lexique *lglex*, les informations syntaxiques sont représentées de façon partiellement formalisée. Dans sa version textuelle, une entrée de *lglex* se présente comme suit :
– l'entrée commence par un identifiant indiquant sa catégorie, la table dont il provient et le numéro de l'entrée dans cette table ;
– la section *lexical-info* indique les informations lexicales liées à l'entrée : lemme, mais également auxiliaire(s) et prépositions associées à certains arguments ; les prépositions possibles à la place de l'indication *Prép* (resp. *Loc*) sont dans la partie *prépositions* (resp. *locs*) ;
– la section *args* décrit les distributions des différents arguments (sujet et compléments, répartis en sous-sections *const* dont la position est repérée par l'élément *pos*) ; une distribution donnée (élément *comp*) indique sa catégorie grammaticale (*comp* pour une complétive, *NP* pour un syntagme nominal, etc.), son introducteur (élément *introd-prep* ou *introd-loc*), des traits sémantiques (*hum*, *nothum*, et d'autres), des traits complémentaires (*mood* dans le cas d'une complétive, etc), et l'intitulé complet des colonnes ayant contribué à définir cette distribution, qui sont toutes de la forme argument = : réalisation ; cette section est donc bien formalisée ;
– la section *all-constructions* liste différentes constructions dans lesquelles l'entrée peut prendre part ; ces constructions sont réparties en constructions absolues (élément *absolute*), qui sont nommées de façon complète (p.ex. N0 V N1), et constructions relatives (élément *relative*) qui sont des noms de redistributions (p.ex. [passif par]) ; les constructions codées + dans la table des classes sont précédées de la mention *true::*, les autres sont précédées de la mention *o::* ; cette section est donc bien moins bien formalisée que la précédente, et les informations qu'elle contient ne sont pas directement exploitables informatiquement.
– la section *exemple* illustre l'entrée.

À titre d'exemple, voici au format *lglex* l'entrée du verbe *ruisseler* dans la table 35L.

ID=V_35L_242
lexical-info=[locs=(loc=[id="1",list=()],loc=[id="2",list=()]),cat="verb",verb=[lemma="ruisseler"],
aux-list=(),prepositions=()]
args=(
const=[dist=(comp=[cat="NP",source="true",introd-prep=(),origine=(orig="Loc N1 =: de N1 source"),
introd-loc=(prep="de")]),pos="1"],
const=[dist=(comp=[cat="NP",introd-prep=(),origine=(orig="Loc N2 =: vers N2 destination",
orig="Loc N2 =: dans N2 destination"),introd-loc=(prep="vers",prep="dans"),destination="true"]),pos="2"],
const=[pos="0",dist=(comp=[cat="NP",introd-prep=(),nothum="true",origine=(orig="N0 =: N-hum"),
introd-loc=()])])
all-constructions=[absolute=(construction="o::N0 V Loc N1 source Loc N2 destination",construction="o::N0 V",
construction="o::N0 être V-ant",construction="true::N0 V Loc N1"),
relative=(construction="Ppv =: y",construction="Ppv =: en",construction="[extrap]")]
example=[example="L'eau ruisselle de la gouttière sur les passants"]

3 Le Le*fff* et le format Alexina

Le Le*fff* (Lexique des formes fléchies du français) est un lexique syntaxique à large couverture pour le français librement distribué (Sagot *et al.*, 2006; Sagot & Danlos, 2007)[4].

[4]Distribution sous license LGPL-LR à l'adresse `http://gforge.inria.fr/projects/alexina/`

Il repose sur l'architecture Alexina d'acquisition et de modélisation de lexiques morphologiques et syntaxiques. Un lexique *intensionnel* associe à chaque entrée un cadre de sous-catégorisation canonique et liste les redistributions possibles à partir de ce cadre. La *compilation* du lexique intensionnel en lexique extensionnel construit alors différentes entrées pour chaque forme fléchie du lemme et chaque redistribution possible. Soit l'entrée intensionnelle simplifiée suivante :

clarifier$_1$ *Lemma;v;*<*arg*$_0$*:Suj:cln|scompl|sinf|sn,arg*$_1$*:Obj:(cla|scompl|sn)*>*;*
%ppp_employé_comme_adj,%actif,%se_moyen_impersonnel,
%passif_impersonnel,%passif

Elle décrit une entrée du lemme verbal *clarifier*, qui est transitive directe (deux arguments réalisés canoniquement par les fonctions syntaxiques Suj et Obj décrites entre les chevrons), et qui admet les redistributions fonctionnelles (préfixées par le symbole %) *participe passé employé comme adjectif*, *actif* (la distribution par défaut), *se moyen impersonnel* (*il s'est clarifié de nombreuses choses à cette réunion*), *passif impersonnel* (*il a été clarifié par Pierre que Luc était le coupable*), et *passif*. À titre d'exemple, l'entrée extensionnelle pour la forme fléchie *clarifiés* et la redistribution *passif* a la forme (simplifiée) suivante :

clarifiés v [pred='clarifier$_1$*<arg*$_1$*:Suj:cln|scompl|sn,arg*$_0$*:Obl2:(par-sn)>',*
@passive,@pers,@Kmp]; %passif

Les fonctions syntaxiques sont définies dans le Le*fff* par des critères proches de ceux de DICOVALENCE (Van den Eynde & Mertens, 2006). L'inventaire des fonctions syntaxiques est le suivant : Suj (sujet), Obj (objet direct), Objà (objet indirect introduit canoniquement par la préposition *à*), Objde (objet indirect introduit canoniquement par la préposition *de*), Loc (locatif), Dloc (délocatif), Att (attribut), Obl ou Obl2 (autres arguments obliques). Les critères définitoires de ces fonctions sont décrits dans (Sagot & Danlos, 2007).

Chaque fonction syntaxique peut être réalisée par trois types de réalisations : *pronoms clitiques*, *syntagme direct* (syntagme nominal (sn), adjectival (sa), infinitif (sinf), phrastique fini (scompl), interrogative indirecte (qcompl)) et *syntagme prépositionnel* (syntagme direct précédé d'une préposition, comme de-sn, à-sinf ou pour-sa[5]. Enfin, une fonction dont la réalisation est facultative voit sa liste de réalisations possibles mise entre parenthèses.

Des informations syntaxiques complémentaires (contrôle, mode des complétives, etc.) sont notées par des *macros* (@CtrlSujObj, @ComplSubj, etc.) dont l'interprétation formalisée dépend du contexte d'utilisation. Une modélisation de ces macros en LFG est fournie avec le Le*fff*.

4 Conversion des tables de verbes simples au format Le*fff*

4.1 Découpage en entrées : la construction de base et ses variantes

Chaque entrée de *lglex* est associée à des constructions dont on peut distinguer plusieurs types :

1. la (ou les) construction(s) dites « de base », c'est-à-dire définitoires de la classe d'origine de l'entrée ; elles sont identifiées dans le format *lglex* par la mention true:: ;
2. les constructions « de base étendues », obtenues par adjonction d'arguments à la construction de base[6] ; nous faisons l'hypothèse (vérifiée en pratique) que ces constructions sont

[5] à-scompl et de-scompl représentent les réalisations en *à/de ce que P*.

[6] La situation est en réalité plus complexe. En effet, certains de ces arguments supplémentaires ne prennent part à aucune des constructions, et ne sont connues que par leur description dans la section *args* de l'entrée. Nous les

toutes des intermédiaires entre la construction de base et une construction dite « de base maximalement étendue » ou CBME ; par exemple, l'entrée de *rassembler* dans la table 32PL (*Max a rassemblé ses articles (dans un ouvrage)*) a pour construction de base la construction transitive simple N0 V N1, mais la possibilité d'ajouter un complément en Loc N2 conduit à une CBME de la forme N0 V N1 Loc N2 ;

3. les constructions qui sont des variantes de la construction de base, obtenues par effacement d'un ou de plusieurs arguments ou par changement de type de réalisation (Qu P devenant V^{i}inf W, par exemple) ;
4. les constructions qui sont en réalité des redistributions (constructions relatives telles que [passif de], constructions absolues de type N_1 est Vpp de ce Qu P) ;
5. les constructions dont il semble qu'elles auraient dû conduire à des entrées distinctes, dites « entrées secondaires », telles que les constructions neutres ou les transformations de type N_1 se V de ce Qu P (cf. *Luc se félicite d'avoir réussi à séduire Léa* vs. *Max félicite Luc qu'il ait réussi à séduire Léa*).

De par leur origine, les informations présentes dans la section *args* d'une entrée *lglex* participent à la définition de constructions qui sont des variantes de la construction de base, étendue ou non (cas 2 et 3). Pour identifier les entrées au format Le*ffff* à créer à partir d'une entrée *lglex*, il faut donc identifier, parmi les constructions listées dans la section *all-constructions*, la construction de base maximalement étendue (CBME), ainsi que les constructions qui relèvent du cas 5.

Nous avons développé une méthode permettant d'*aligner* deux constructions, c'est-à-dire de construire des correspondances entre arguments, malgré leurs différences de surface[7] et leur possible effacement. Cette méthode nous permet d'identifier et d'aligner la CBME et ses variantes, rassemblées pour former une entrée unique du lexique final, dite entrée *canonique*[8]. Pour chaque variante, nous construisons une séquence d'opérations élémentaires permettant de la reconstituer à partir de la CBME. Ces opérations permettront de construire, pour chaque argument, la liste de ses réalisations possibles, et de déterminer si elles sont effaçables.

Parmi les autres constructions (qui ne sont donc pas des variantes de la CBME), celles qui correspondent à des redistributions standard (*[passif par]*, *[extrap]*...) induisent l'ajout à l'entrée finale de la redistribution correspondante. Les autres induisent la création d'entrées distinctes dites *secondaires*, qu'elles relèvent du cas 5 ou qu'elles correspondent à des redistributions non encore répertoriées par le format Alexina.

Reprenons l'exemple ci-dessus. L'entrée indique les constructions suivantes :

all-constructions=[absolute=(construction="o::N0 V Loc N1 source Loc N2 destination",construction="o::N0 V",
construction="o::N0 être V-ant",construction="true::N0 V Loc N1"),
relative=(construction="Ppv =: y",construction="Ppv =: en",construction="[extrap]")]

La construction de base est donc N_0 V Loc N_1 (cas 1). La construction N_0 V Loc N_1 source Loc N_2 destination en étant l'unique extension, il s'agit de la CBME (cas 2). Les indications sémantiques de type source et destination étant ignorées, la construction de base devient une variante de la CBME, par effacement de l'argument 2. La construction N_0 V est également identifiée comme une variante de la CBME, par double effacement (cas 3). La construction

prenons en compte correctement.

[7] Par exemple, Qu P vs. N1, ou encore à N1 vs. Prép N1 si l'on sait par ailleurs, grâce à la section *lexical-info* ou à la distribution concernée, que la Prép peut être *à*.

[8] À l'exception de certaines constructions relatives, telles que Ppv = : y, pour lesquelles la description de la variante correspondante est obtenue directement.

N_0 être V-ant est laissée de côté pour l'instant. Du côté des constructions relatives, on trouve la construction Ppv = : y (resp. Ppv = : en) qui permettra d'ajouter *y* (resp. *en*) à la liste des réalisations de l'argument locatif (resp. délocatif) (cas 3). Enfin, la construction [extrap] induira l'adjonction d'une redistribution impersonnelle (%actif_impersonnel) à l'entrée (cas 4).

4.2 Construction des cadres de sous-catégorisation

Une fois répertoriées les entrées à produire, les cadres de sous-catégorisation sont construits. Pour cela, on calcule d'abord le cadre correspondant à la construction maximale de chaque entrée (la CBME pour l'entrée canonique, ou l'unique construction des entrées secondaires).

La fonction syntaxique de chaque argument est déterminée au moyen d'heuristiques, ainsi que sa réalisation canonique. Les fonctions syntaxiques sont obtenues de la façon suivante. Tout d'abord, le premier argument reçoit toujours la fonction Suj[9]. Le premier argument post-verbal, s'il est direct, se voit attribuer la fonction Obj, sauf pour les entrées de la table 32NM[10]. Ensuite, un argument introduit par *à* (resp. *de*) reçoit la fonction syntaxique Objà (resp. Objde), sauf si un indice complémentaire vient contredire ce choix (par exemple, pour un argument N_1 introduit par *à*, la propriété à N_1 = Ppv = : le lui confèrera fonction syntaxique Obj, ex. : *Il apprend à conduire / Il l'apprend*). Les arguments introduits par *Loc* ont la fonction syntaxique Loc, sauf ceux de la forme Loc N_i source ou vérifiant Loc N_i = : de N_i source, qui ont la fonction syntaxique Dloc. Enfin, les autres arguments sont considérés comme des Att s'ils sont directs, et comme des Obl s'ils sont introduits par une préposition (Obl2 si un Obl existe déjà).

Les réalisations de ces fonctions sont construites en deux temps. Tout d'abord, le type de syntagme (nominal, infinitif, phrastique,...) est déterminé. Par exemple, la réalisation scompl correspond aux distributions de catégorie *comp* et *ceComp* et aux arguments dans les constructions en Qu P. Ceci étant fait, il reste à lister les introducteurs possibles[11]. Nous ne rentrerons pas dans les détails, mais l'ensemble des prépositions et autres introducteurs (p.ex. *et*) sont pris en compte. Le résulat de ces heuristiques est le cadre de sous-catégorisation au format L*efff* pour la construction maximale de l'entrée. Ainsi, la CBME de l'exemple précédent devient

<Suj:sn|cln,Dloc:sn,Loc:sn>.

L'entrée canonique doit alors être complétée pour inclure les variantes de la CBME. Pour cela, nous ajoutons tout d'abord les réalisations issues de constructions telles que Loc N1 = Ppv = : y. Nous répercutons ensuite pour chaque variante de la CBME la séquence d'opérations élémentaires permettant de la dériver de la CBME : tout effacement d'un argument rend l'argument facultatif ; tout changement de réalisation induit une nouvelle réalisation possible de la fonction syntaxique concernée[12]. Le résultat de ce processus est un cadre de sous-catégorisation avec

[9] Les constructions impersonnelles sont toutes obtenues sous forme de redistributions, puisque la table 38I n'est pas prise en compte dans le lexique *lglex*.

[10] Elle rassemble les verbes à pseudo-objet de type *peser* (*Le sac pèse 10 kg*) ou *sentir* (*Cette pièce sent la fumée*).

[11] De plus, la réalisation cln (clitique nominatif) est ajoutée systématiquement à la fonction syntaxique Suj.

[12] Les dépendances entre ces opérations (par exemple, tel argument est effaçable seulement si tel autre l'est aussi) sont *perdues* : tout effacement est considéré comme possible sans condition, alors même qu'il peut ne venir que d'une seule construction. De même, toute réalisation d'un argument autorisée par une construction devient possible quelles que soient les réalisations des autres arguments. Il s'agit d'une approximation des données linguistiques présentes dans les tables. Cette approximation, qui n'a pas de conséquences importantes pour l'utilisation du lexique produit, a le mérite de permettre de diminuer au maximum le nombre d'entrées. De plus, elle est linguistiquement acceptable, puisqu'elle est par exemple également adoptée par le modèle de la valence mis en œuvre par le lexique DICOVALENCE (Van den Eynde & Mertens, 2006).

alternatives et arguments optionnels. Le cadre construit pour l'entrée canonique de l'exemple précédent est alors

<Suj:cln|sn,Dloc:(de-sn|en),Loc:(vers-sn|dans-sn|y)>.

4.3 Identification des redistributions admissibles

Le cadre de sous-catégorisation de base construit à la section précédente est le cadre dit *profond*, car il relève de la syntaxe profonde. Néanmoins, la table 38I des verbes impersonnels intrinsèques et autonomes ayant été écartée du lexique *lglex*, ce cadre profond correspond toujours à un cadre de surface, celui de la (re)distribution active (%actif). Toutes les entrées se voient donc attribuer au moins cette (re)distribution.

Les autres redistributions admissibles sont identifiées parmi les constructions répertoriées dans la section *all-constructions* de l'entrée *lglex*. Ainsi, [passif par] et [passif de] correspondent aux redistributions %passif et %passif_de, alors que [extrap] et [extrap][passif] correspondent aux redistributions %actif_impersonnel et %passif_impersonnel[13]. Comme étudié par (Danlos & Sagot, 2008), les redistributions pronominales (de type *se* moyennes et *se* neutres) étant mal codées dans le Lexique-Grammaire, nous ne les avons pas prises en compte en tant que redistributions. Le résultat de cette extraction pour l'exemple précédent est ainsi

<Suj:cln|sn,Dloc:(de-sn|en),Loc:(vers-sn|dans-sn|y)>;;%actif.

Un exemple plus riche, tel que celui de l'entrée de *bénéficier* dans la table 8 (*Max bénéficie de ce que Luc est parti*), devient quant à lui[14]

<Suj:cln|scompl|sinf|sn,Objde:de-scompl|de-sinf|de-sn|en>;;
%actif,%passif_impersonnel,%actif_impersonnel.

4.4 Prise en compte des informations complémentaires

D'autres types d'informations sont alors ajoutées, pour former l'entrée finale. Tout d'abord, le prédicat sémantique correspondant à l'entrée, qui est le plus souvent *Lemma* (c'est-à-dire identique au lemme verbal), peut être complété (par exemple en *se Lemma*, *ne pas Lemma*, etc) en fonction d'informations présentes dans la section *lexical-info* de l'entrée *lglex* (*ppvse="true"*, *neg="true"*, etc). Le lemme verbal lui-même, en-tête de l'entrée, est extrait du champ *lemma*, et complété par le numéro de la table dont est issue l'entrée, ainsi que du numéro d'entrée dans cette table. Ainsi, même lorsqu'une entrée *lglex* donne lieu à plusieurs entrées au format Le*fff*, ces entrées partagent un même identifiant. Par ailleurs, la classe de fréquence de l'entrée est récupérée dans le DELA, et est traduite en un poids de la façon suivante : les entrées *z1* reçoivent un poids standard de 100, les entrées *z2* (resp. *z3*) reçoivent un poids de 70 (resp. 50).

Les informations syntaxiques complémentaires les plus intéressantes sont représentées sous forme de macros. Il s'agit des informations suivantes :

– auxiliaire de conjugaison, récupéré dans le champ *aux-list*, qui induit l'ajout de la macro *@avoir* ou de la macro *@être* ;

[13] Rappelons que la table des classes explicite toutes les propriétés vérifiées par toutes les entrées d'une table. Ainsi, le fait que toute entrée à complément direct ne faisant pas partie de la table 32NM (c'est-à-dire toute entrée recevant une fonction syntaxique Obj) soit passivable est codé dans la table des classes par un + à l'intersection des lignes concernées et de la colonne [extrap].

[14] On note que la validité des redistributions impersonnelles est discutable.

– caractère (essentiellement) pronominal (*ppvse="true"*, macro *@pron*) ;
– caractère obligatoirement négatif (*neg="true"*, macro *@neg*) ;
– mode des complétives possibles (champ *mood* de la distribution correspondante), indiquée sous la forme de macros de type $@fM$, où f vaut respectivement *SComp*, *Comp*, *AComp* et *DeComp* pour les complétives de fonction syntaxique Suj, Obj, Objà et Objde, et où M est *Ind* ou *Subj* (ex. : *@CompInd*) ;
– informations de contrôle, extraites des champs *contr* des distributions ou du nombre i dans les arguments de type $\mathsf{V}^i\mathsf{inf}$ au sein des constructions, et qui sont représentées par des macros de la forme $@Ctrl f_1 f_2$, où f_1 est la fonction syntaxique du contrôleur et f_2 celle du contrôlé (exemple : *@CtrlSujObj*).

Enfin, la table de conjugaison à associer à l'entrée est récupérée dans le Le*fff*, permettant ainsi la compilation du lexique obtenu avec la description morphologique du français du Le*fff*.

Pour l'entrée de *ruisseler* dans la table 35L, que nous suivons depuis le début de cet article, le résultat final est alors le suivant :

ruisseler$^{35L}_{242}$ *v-er:std 100;Lemma;v;<Suj:cln|sn,Dloc:(de-sn|en),Loc:(vers-sn|dans-sn|y)>;cat=v;%actif*

Pour l'entrée de *bénéficier* dans la table 8, mentionnée ci-dessus, le résultat est :

bénéficier$^{8}_{10}$ *v-er:std 100;Lemma;v;<Suj:cln|scompl|sinf|sn,Objde:de-scompl|de-sinf|de-sn|en>;*
cat=v,@DeCompInd,@CtrlSujObjde;%actif,%passif_impersonnel,%actif_impersonnel_compl

4.5 Résultats

Le lexique verbal obtenu contient 16 903 entrées pour 5 694 lemmes verbaux différents, c'est-à-dire un nombre moyen d'entrées par lemme de 2,96. À titre de comparaison, le Le*fff* contient seulement 7072 entrées verbales pour 6818 lemmes verbaux distincts, soit un nombre moyen d'entrées par lemme de 1,04. Le lexique issu de *lglex*, quoique décrivant moins de lemmes verbaux, est donc beaucoup plus couvrant en termes de constructions syntaxiques et donc beaucoup plus ambigu. Les lemmes verbaux les plus ambigus dans le Le*fff* sont *tenir* et *(re)faire* (6 entrées) alors que dans le lexique extrait de *lglex* il s'agit des lemmes *tenir* (53 entrées), *jouer* (44 entrées) et *prendre* (35 entrées). Au niveau extensionnel, le Le*fff* contient 361 268 entrées, alors que le lexique extrait de *lglex* en contient 763 555.

L'obtention de ce lexique à partir de *lglex* par le processus décrit ici est réalisée par un script *perl* de moins de 1 000 lignes. La conversion proprement dite, c'est-à-dire l'exécution du script sur l'ensemble de *lglex*, prend moins d'une minute. Si donc une nouvelle version des tables et de la table des classes sont publiées, la construction du lexique correspondant au format Le*fff* est quasiment instantanée, et ne nécessite aucun nouveau développement.

5 Intégration dans FRMG : évaluation et discussion

La motivation de ce travail est de permettre aux données linguistiques codées dans les tables du Lexique-Grammaire de servir de base de données lexicales pour un analyseur syntaxique automatique du français. Parmi les analyseurs qui prennent en entrée un lexique au format Le*fff*, nous avons choisi l'analyseur FRMG (Thomasset & de la Clergerie, 2005). Il s'appuie sur une Grammaire d'Adjonction d'Arbres (TAG) compacte du français générée à partir d'une métagrammaire, et sur le Le*fff*. La compilation et l'exécution de l'analyseur se déroule dans

le cadre du système DYALOG (de la Clergerie, 2005). Il utilise comme entrée le résultat de la chaîne de traitement présyntaxique SXPipe (Sagot & Boullier, 2008), qui convertit un texte brut en suite de treillis de formes connues[15].

L'intégration du lexique au format Le*fff* extrait de *lglex* dans l'analyseur FRMG est immédiate : le *lexeur* de l'analyseur fait appel à une base de données lexicales construite à partir du Le*fff*. Il suffit de remplacer les entrées verbales du Le*fff* par le lexique construit à partir de *lglex*, de conserver les autres entrées du Le*fff*, de construire la base de données lexicales correspondantes, et de spécifier à FRMG d'utiliser cette dernière. Le résultat est une variante de l'analyseur FRMG, que nous noterons $\text{FRMG}_{\text{lglex}}$, par opposition à la variante standard notée $\text{FRMG}_{\text{Le}fff}$.

Nous avons évalué $\text{FRMG}_{\text{Le}fff}$ et $\text{FRMG}_{\text{lglex}}$ en analysant la partie annotée manuellement du corpus EASy (Paroubek *et al.*, 2005), soit 4 306 phrases de styles variés (journalistique, médical, oral, questions, littéraire...), et en utilisant la métrique d'évaluation EASy (les chunks et les « relations » EASy, c'est-à-dire approximativement les dépendances entre mots pleins). Les résultats détaillés de cette évaluation seront publiés par ailleurs. Globalement, on observe que les résultats sont pour l'instant légèrement meilleurs pour $\text{FRMG}_{\text{Le}fff}$. Par exemple, la f-mesure globale sur les chunks est de 84,4 % pour $\text{FRMG}_{\text{Le}fff}$ contre 82,2 % pour $\text{FRMG}_{\text{lglex}}$, et celle sur les relations est de 59,9 % pour $\text{FRMG}_{\text{Le}fff}$ contre 56,1 % pour $\text{FRMG}_{\text{lglex}}$. Nous ne pensons pas que ce résultat remette en question la pertinence de l'utilisation des tables du Lexique-Grammaire en analyse syntaxique, pour plusieurs raisons : d'une part, le processus de conversion décrit ici et son implémentation en sont encore à une version préliminaire ; d'autre part, le développement du Le*fff* a été réalisé en parallèle avec celui de FRMG et les évaluations EASy ; enfin, le lexique issu des tables a dû être complété par certaines entrées manquantes (auxiliaires et semi-auxiliaires, verbes impersonnels, etc...), mais certaines autres manquent peut-être encore. Du reste, nous avons déjà identifié des problèmes systématiques dans les données de départ (telle que la passivabilité incomplètement codée dans la table des classes) dont l'impact sur les résultats est très important. Une fois ces problèmes résolus, nous espérons que $\text{FRMG}_{\text{lglex}}$ aura des résultats significativement meilleurs que $\text{FRMG}_{\text{Le}fff}$.

Il est intéressant de constater que, dès à présent, la couverture des deux analyseurs est assez complémentaire. Par exemple, sur le sous-corpus `general_lemonde`, 177 phrases sont entièrement reconnues par les deux analyseurs, 85 seulement par $\text{FRMG}_{\text{Le}fff}$, 76 seulement par $\text{FRMG}_{\text{lglex}}$, et 111 par aucun des deux. Cette complémentarité est prometteuse. En particulier, il sera intéressant d'étudier les différences entre les erreurs faites par chacun de ces deux analyseurs, y compris au moyen de techniques automatiques telles que celles décrites dans (Sagot & de La Clergerie, 2008). Ceci pourrait permettre d'améliorer les différentes ressources, voire de détecter automatiquement des erreurs dans les lexiques. Dans le cas de *lglex*, nous nous attendons à ce que la plupart de ces erreurs proviennent du processus de conversion, mais certaines proviendront peut-être d'erreurs dans les tables du Lexique-Grammaire, et permettront donc d'améliorer ces dernières.

6 Conclusion et perspectives

Nous avons développé une méthodologie et un outil permettant de convertir la version textuelle des tables du Lexique-Grammaire en un lexique TAL utilisant le formalisme lexical du Le*fff*, et

[15] SXPipe comprend entre autres des modules de segmentation en phrases, de tokenization et correction orthographique, de détection d'entités nommées, et d'identification non déterministe des mots composés.

donc utilisable par l'analyseur syntaxique FRMG. La pertinence du lexique obtenu a été validée par son utilisation pour l'analyse syntaxique automatique du corpus d'évaluation EASy.

Ce premier travail nous a permis d'identifier un certain nombre de problèmes dans les données de départ (tables et table des classes), mais nous a conduit également à un certain nombre de simplifications et d'approximations dans le processus de conversion. Il y a donc la place pour des améliorations importantes, qui pourraient permettre à terme de construire un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire dont l'utilisation améliorera la qualité des outils et des ressources existants, par exemple par fusion avec d'autres ressources lexicales ou par intégration dans un analyseur syntaxique opérationnel.

Références

BOONS J.-P., GUILLET A. & LECLÈRE C. (1976). *La structure des phrases simples en français, Constructions intransitives*. Genève : Droz.

CONSTANT M. & TOLONE E. (2008). A generic tool to generate a lexicon for nlp from lexicon-grammar tables. In *Actes du 27ème Colloque Lexique et Grammaire*, L'Aquila, Italie.

DANLOS L. & SAGOT B. (2008). Constructions pronominales dans dicovalence et le lexique-grammaire – intégration dans le Le*fff*. In *Actes du 27ème Colloque Lexique et Grammaire*, L'Aquila, Italie.

DE LA CLERGERIE E. (2005). DyALog : a tabular logic programming based environment for NLP. In *Proceedings of 2nd International Workshop on Constraint Solving and Language Processing (CSLP'05)*, Barcelona, Spain.

GARDENT C., GUILLAUME B., PERRIER G. & FALK I. (2005). Maurice Gross' grammar lexicon and natural language processing. In *Proc. of the 2nd LTC*, Poznań, Poland.

GROSS M. (1975). *Méthodes en syntaxe*. Paris, France : Hermann.

GUILLET A. & LECLÈRE C. (1992). *La structure des phrases simples en français : Les constructions transitives locatives*. Genève : Droz.

PAROUBEK P., POUILLOT L.-G., ROBBA I. & VILNAT A. (2005). EASy : campagne d'évaluation des analyseurs syntaxiques. In *Proceedings of the EASy workshop of TALN 2005*, Dourdan, France.

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

SAGOT B. & BOULLIER P. (2008). SXPipe 2 : architecture pour le traitement présyntaxique de corpus bruts. *Traitement Automatique des Langues (T.A.L.)*, **49**(2). À paraître.

SAGOT B., CLÉMENT L., DE LA CLERGERIE E. & BOULLIER P. (2006). The Le*fff* 2 syntactic lexicon for French : architecture, acquisition, use. In *Proc. of LREC'06*.

SAGOT B. & DANLOS L. (2007). Améliorer un lexique syntaxique à l'aide des tables du lexique-grammaire – Constructions impersonnelles. *Cahiers du Cental*.

SAGOT B. & DE LA CLERGERIE É. (2008). Fouille d'erreurs sur les sorties d'analyseurs syntaxiques. *Traitement Automatique des Langues (T.A.L.)*, **49**(1).

THOMASSET F. & DE LA CLERGERIE E. (2005). Comment obtenir plus des méta-grammaires. In *Proceedings of TALN'05*, Dourdan, France : ATALA.

VAN DEN EYNDE K. & MERTENS P. (2006). Le dictionnaire de valence DICOVALENCE : manuel d'utilisation. http ://bach.arts.kuleuven.be/dicovalence/manuel_061117.pdf.