Annotation sémantique du French Treebank à l'aide de la réécriture modulaire de graphes

Bruno Guillaume[1, 2] Guy Perrier[1, 3]
(1) LORIA - Campus Scientifique - BP 239 - 54506 Vandœuvre-lès-Nancy cedex
(2) INRIA Grand Est - 615, rue du Jardin Botanique - 54600 Villers-lès-Nancy
(3) Université de Lorraine - 34, cours Léopold - CS 25233 - 54502 Nancy cedex
bruno.guillaume@loria.fr, guy.perrier@loria.fr

RÉSUMÉ

Nous proposons d'annoter le French Treebank à l'aide de dépendances sémantiques dans le cadre de la DMRS en partant d'une annotation en dépendances syntaxiques de surface et en utilisant la réécriture modulaire de graphes. L'article présente un certain nombre d'avancées concernant le calcul de réécriture utilisé : l'utilisation de règles pour faire le lien avec des lexiques, en particulier le lexique des verbes de Dicovalence, et l'introduction de filtres pour écarter à certaines étapes les annotations incohérentes. Il présente aussi des avancées dans le système de réécriture lui-même, qui a une plus large couverture (constructions causatives, verbes à montée, ...) et dont l'ordre des modules a été étudié de façon plus systématique. Ce système a été expérimenté sur l'ensemble du French Treebank à l'aide du prototype GREW, qui implémente le calcul de réécriture utilisé.

ABSTRACT

Semantic Annotation of the French Treebank using Modular Graph Rewriting

We propose to annotate the French Treebank with semantic dependencies in the framework of DMRS starting from an annotation with surface syntactic dependencies and using modular graph rewriting. The article presents some new results related to the rewriting calculus: the use of rules to make a link with lexicons, especially with the lexicon of verbs Dicovalence, and the introduction of filters to discard inconsistent annotations at some computation steps. It also presents new results related to the rewriting system itself: the system has a larger coverage (causative constructions, rising verbs, ...) and the order between modules has been studied in a more systematic way. This system has been experimented on the whole French Treebank with the prototype GREW, which implements the used rewriting calculus.

MOTS-CLÉS : réécriture de graphes, interface syntaxe-sémantique, dépendances, DMRS.

KEYWORDS: graph rewriting, syntax-semantics interface, dependencies, DMRS.

Introduction

Les résultats présentés dans cet article se situent dans la continuité de (Bonfante *et al.*, 2010, 2011; Morey, 2011). Le but de ces différents travaux est de produire une annotation sémantique de gros corpus à partir d'une annotation syntaxique en dépendances. L'annotation sémantique se situe au niveau de la phrase et elle est réalisée, comme pour la syntaxe, sous forme de dépendances.

L'idée est d'avoir une annotation lisible et minimale, c'est-à-dire évitant tout engagement dans des choix linguistiques trop pointus, qui susciteraient la controverse. Nous avons choisi comme cadre formel la DMRS (Dependency Minimal Recursion Semantics) (Copestake, 2009), car elle répond à ces exigences.

Pour le français, il existe très peu de corpus annotés syntaxiquement ; le plus grand actuellement est le *French Treebank*. Le French Treebank est un corpus de phrases extraites du journal « Le Monde » qui ont été annotées en constituants (Abeillé *et al.*, 2003). Ces annotations ont ensuite été converties en dépendances syntaxiques (Candito *et al.*, 2009) en suivant le format décrit dans le guide mis au point à cet effet[1]. C'est ce dernier corpus qui est utilisé ici sous l'abréviation FTB.

Même si les ambiguïtés syntaxiques sont levées par l'annotation du FTB, la production des structures sémantiques est encore ambiguë et plusieurs structures DMRS peuvent correspondre à un seule phrase du FTB.

Les structures DMRS produites font intervenir la notion d'actant sémantique mais les actants ne sont pas classés selon leur rôle. La DMRS n'a pas voulu s'engager par rapport à un choix particulier de rôles thématiques. Les actants d'un prédicat donné sont simplement distingués par un numéro et notés $arg1$, $arg2$, ... en suivant un ordre d'oblicité syntaxique fixé. Quand le prédicat est un verbe dans notre application, ces arguments font référence à un lexique externe qui est Dicovalence (Van den Eynde et Mertens, 2003). Bien que celui-ci soit avant tout un lexique syntaxique, il dispose d'informations fines et fait notamment des distinctions de lemmes en fonction de leur traduction en néerlandais et en anglais ; il peut donc être utilisé comme un lexique sémantique.

Pour le calcul, le cadre formel que nous utilisons est celui de la *réécriture de graphes* qui est motivée par la forme des objets que nous manipulons. En effet, les structures DMRS produites sont des graphes de dépendances sémantiques. Même si, en entrée, les structures sont le plus souvent des arbres de dépendances syntaxiques, pour le calcul de la sémantique, nous avons besoin de compléter ces arbres avec, par exemple, certains actants syntaxiques des infinitifs et certains antécédents des pronoms déterminés par la syntaxe. Nous obtenons alors des graphes dans toute leur généralité, avec des nœuds qui ont plusieurs antécédents et avec des cycles.

Contrairement au cas de la réécriture de termes, il n'y a pas de définition canonique de la réécriture de graphes. Nous avons choisi une définition opérationnelle (cf. Section 1) où la partie droite d'une règle est décrite par des commandes. Nous l'avons implanté dans un prototype baptisé GREW[2] qui a été utilisé pour l'expérimentation sur le FTB.

Les résultats présentés dans cet article constituent des avancées par rapport aux travaux précédents sur deux plans. D'un part, le calcul de réécriture de graphes a été enrichi par l'introduction de règles lexicales qui permettent de faire le lien avec des lexiques (Dicovalence dans notre application) et par l'introduction de filtres qui opèrent à la fin des modules pour ne conserver que les annotations cohérentes selon certains critères linguistiques ; ces nouveautés sont présentées en Section 2. D'autre part, le système de règles a été significativement étendu et l'ordonnancement des modules a été revu de façon systématique ; ces nouveautés sont présentées en Section 3. La Section 4 décrit alors l'ensemble des règles de notre système. Nous sommes ainsi en mesure de fournir des résultats expérimentaux plus riches et plus complets qui sont exposés à la Section 5.

[1] `http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html`
[2] `http://grew.loria.fr`

1 Présentation du calcul

1.1 La forme des règles de réécriture

Les règles de réécriture sont décrites en trois parties ; considérons une règle R qu'on cherche à appliquer à un graphe G ; R se compose de :

- un *patron positif* qui est un graphe qu'on va chercher à apparier avec une partie de G, l'appariement se faisant de façon injective ;
- un ensemble éventuellement vide de *patrons négatifs* qui sont des extensions[3] du patron positif et qui imposent des conditions négatives sur G ; s'il est possible d'étendre un appariement du patron positif de R avec l'un des patrons négatifs de R, la règle R ne s'applique pas ;
- une liste de *commandes* qui vont être exécutées dans l'ordre ; une commande va effectuer une transformation élémentaire sur G (ajout ou suppression d'un nœud, d'un arc ou d'un trait, fusion de deux nœuds, ...).

Si une règle s'applique à un graphe G et produit un graphe G', nous écrirons : $G \rightarrow G'$.

On peut maintenant s'interroger sur le coût de l'application d'une règle R à un graphe G. On doit procéder à l'appariement d'un graphe (un patron de la règle R) avec un sous-graphe du graphe G et on sait que ce problème, dans toute sa généralité, est NP-complet. Ici, néanmoins, cela se fait dans des conditions particulières :

- les patrons utilisés dans les règles sont très petits ;
- les patrons sont toujours connexes et ont généralement une seule racine, exe exceptionnellement deux (une racine est un nœud qui est un ancêtre de chacun des autres nœuds du graphe) ;
- le nombre d'arcs sortant d'un nœud d'un patron est borné (pour le système utilisé dans ce travail, la borne est de 3).

Dans ces conditions, pour un patron fixé, l'appariement est linéaire en temps par rapport à la taille du graphe auquel s'applique le patron. Il est important également de noter que dans le cas où le patron et le graphe sont tous les deux des arbres, l'appariement qui est utilisé suit le même algorithme que les outils standards de réécriture d'arbres.

1.2 L'organisation des règles en modules

Dans notre utilisation de la réécriture de graphes pour l'interface syntaxe-sémantique, chaque principe linguistique est traduit en quelques règles lisibles et simples. Leur intérêt est que leur effet est local mais le revers de la médaille est qu'il est difficile de contrôler l'interaction de centaines de règles pouvant agir en parallèle.

Pour faciliter la lisibilité et la maintenance de la cohérence globale, les règles sont regroupées en *modules*, chaque module comprenant un ensemble de règles qui ont une unité linguistique propre. L'ordre d'application des règles au sein d'un module est libre alors que l'ordre entre

[3] Un patron négatif respecte la même syntaxe qu'un patron positif et peut faire référence à des nœuds définis dans le patron positif.

modules peut être contraint.

Les modules jouent en plus un rôle décisif dans la mise en œuvre du calcul, sous l'angle de la terminaison et de la confluence. Un module M a la *propriété de terminaison* si pour tout graphe G, il n'existe pas de réécriture infinie $G \rightarrow G_1 \rightarrow G_2 \rightarrow \cdots$ par application de règles de M. Dans notre système, tous les modules ont la propriété de terminaison. Le problème aurait pu venir des règles créant de nouveaux nœuds dans un graphe. Or, tous les nœuds créés sont rattachés à un nœud initial et il n'est possible d'en créer qu'un nombre fini. Dans ces conditions, la quasi-totalité des modules peuvent être munis d'une mesure proportionnelle à la taille du graphe et qui décroit à chaque application de règles. Pour les autres modules, une mesure quadratique dans la taille du graphe existe.

Etant donné un module M, un graphe G est une M-forme normale si aucune règle de M ne s'applique à G. Compte tenu de la propriété de terminaison, on sait que tous les graphes se réécrivent en un nombre fini de M-formes normales. L'application d'un module M à un graphe consiste donc à calculer ses M-formes normales.

Un module M est *confluent* si tout graphe a une unique M-forme normale. Cette propriété est intéressante car elle rend le calcul indifférent à l'ordre d'application des règles. On ne peut pas espérer avoir tous les modules confluents car la transformation d'une annotation syntaxique en annotation sémantique est par essence non confluente, dans la mesure où une même annotation syntaxique peut donner lieu à plusieurs lectures sémantiques. Avec l'organisation en modules, on peut néanmoins contrôler les effets négatifs de la non confluence en la restreignant à certains modules particuliers.

En résumé, ce n'est ni l'appariement des patrons de règles, ni la longueur des chemins de réécriture qui est source de complexité des calculs. C'est avant tout la non confluence de certains modules mais ce n'est pas la méthode, la réécriture de graphes, qui est en cause. C'est l'essence même du problème auquel elle est confrontée, la production d'une représentation sémantique à partir d'une représentation syntaxique. On peut même ajouter que la source en est essentiellement l'ambiguité lexicale.

2 L'enrichissement du calcul

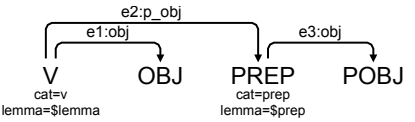
2.1 Branchement de lexiques

Dans les précédents travaux était déjà mis en évidence le fait que certaines règles font appel à des informations lexicales. C'est typiquement le cas pour les règles qui transforment les actants syntaxiques des verbes en arguments sémantiques. Elles utilisent les cadres de sous-catégorisation attachés aux différents sens des verbes. Des informations lexicales sont nécessaires pour traiter les expressions figées, les diverses classes sémantiques d'adverbes (Bonami *et al.*, 2004), d'adjectifs (Dixon et Aikhenvald, 2006; Partee, 2010) et les cadres de sous-catégorisation des noms prédicatifs(Giry-Schneider, 1987).

Dans les premières versions de notre système, les informations lexicales étaient gérées manuellement dans les règles elles-mêmes, ce qui était particulièrement lourd pour la maintenance et nuisait à la lisibilité du système. De plus, il n'était possible de paramétrer l'utilisation d'une règle que par un élément : la valeur du lemme. Les nouvelles règles lexicales permettent :

- d'utiliser plusieurs paramètres (repérés par le symbole $) dans les patrons de la règle (dans l'exemple ci-dessous, le lemme et une préposition) ;
- d'utiliser des paramètres (repérés par le symbole @) dans la partie commandes des règles (dans l'exemple, le numéro de l'entrée dicovalence).

Pour l'exemple des verbes transitifs avec un objet indirect introduit par une préposition (différente de *« à »* et *« de »*), la règle `suj_V_obj_pobj` qui transforme les actants syntaxiques en arguments sémantiques utilise le motif ci-contre et est décrite par le code suivant :




```
1 lex_rule suj_V_obj_pobj (feature $lemma, $prep, @dicoval_id; file "suj_V_obj_pobj.lp")  {
2   match{
3     V [cat=v, lemma=$lemma];
4     OBJ [];
5     e1: V -[obj]-> OBJ;
6     PREP [cat=prep, lemma=$prep];
7     e2:V -[p_obj]-> PREP;
8     POBJ [];
9     e3:PREP -[obj]-> POBJ;
10  }
11  without { V [frame=*] }
12  without { V -[a_obj|de_obj|ato|ats]-> * }
13  commands {
14    del_edge e1; del_edge e2; del_edge e3;
15    del_node PREP;
16    add_edge V -[arg2]-> OBJ; add_edge V -[arg3]-> POBJ;
17    V=@dicoval_id;
18  }
19 }
```

Cette règle fait référence au fichier `suj_V_obj_pobj.lp` qui contient la liste des 149 entrées de Dicovalence correspondant à ce cadre de sous-catégorisation. Chaque ligne décrit, dans l'ordre, le lemme, la préposition régie et le numéro de l'entrée Dicovalence. Les deux premiers éléments constituent les valeurs possibles des paramètres d'entrée de la règle alors que le dernier représente la valeur correspondante du paramètre de sortie.

```
accommoder#avec##900
accorder#avec##1090
accoupler#avec##1305
     ...
troquer#contre##84610
voir#en##86390
```

Lorsque l'on applique la règle à un graphe de dépendance, on cherche à apparier le patron positif de la règle (lignes 2 à 10) avec une partie du graphe et on vérifie que l'appariement ne peut être étendu en un appariement pour aucun des patterns négatifs. Si l'application de la règle réussit, on vérifie que le couple de valeurs dans le graphe qui coïncide avec `$lemma` et avec `$prep` correspond à l'une des lignes du fichier `suj_V_obj_pobj.lp`. Si c'est le cas, on instancie `@dicoval_id` avec la valeur correspondante de sortie du lexique. Les informations lexicales extraites de Dicovalence sont transformées automatiquement en une suite de règles lexicales (301 actuellement), accompagnées chacune d'un fichier d'instanciation des paramètres.

Lorsqu'il n'existe pas de ressources directement utilisables, des embryons de lexiques ont été construits manuellement. C'est le cas pour les cadres de sous-catégorisation des adjectifs et

des noms prédicatifs. C'est aussi le cas pour les adverbes qui ont une portée flottante, pour les adjectifs qui ne sont pas intersectifs.

2.2 Les filtres

Au sein de chaque module, on a défini précédemment la notion de graphe en forme normale. Or, parmi les formes normales, on peut distinguer celles qui sont cohérentes selon certains critères linguistiques de celles qui ne le sont pas.

Pour effectuer le tri entre les unes et les autres, nous avons introduit des *filtres*. Un filtre se présente comme une règle de réécriture standard mais sans la partie *commandes* ; il comporte seulement un patron positif et éventuellement des patrons négatifs. Exécuté en fin du module dans lequel il se situe, il ne fait que supprimer les graphes pour lesquels l'appariement entre le filtre et une partie du graphe réussit.

Les filtres peuvent être utiliser de deux façons différentes. Ils peuvent supprimer des graphes mal annotés initialement et dans lesquels l'information linguistique n'est plus cohérente. Mais ils peuvent également être utiliser pour simplifier l'écriture d'un jeu de règles : il est parfois plus simple d'écrire un ensemble de règles qui surgénèrent et d'utiliser ensuite un filtre pour ne garder que les solutions pertinentes. Par exemple, le filtre qui a été introduit dans le module qui transforme les actants syntaxiques des verbes en actants sémantiques joue ce double rôle. Il supprime les annotations dans lesquelles il subsiste des arguments syntaxiques. C'est le cas notamment des verbes intransitifs qui auraient un objet direct en syntaxe profonde.

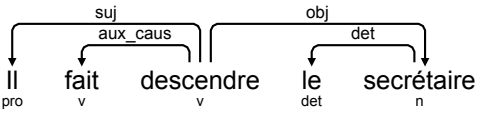
3 L'enrichissement du système de règles

3.1 Extension de la couverture grammaticale

Les systèmes de règles présentés dans (Bonfante *et al.*, 2010, 2011; Morey, 2011) avaient une couverture grammaticale relativement limitée. Celle-ci a été étendue significativement. Nous en donnons ici les exemples les plus marquants.

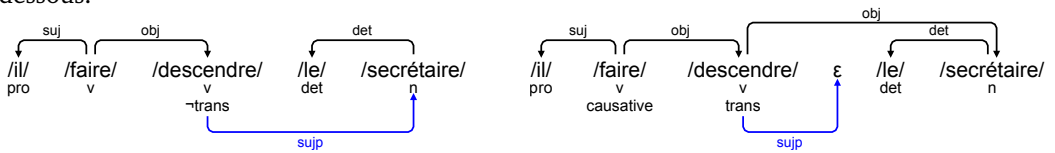
Parmi les constructions qui ont été intégrées, on trouve les constructions causatives. Dans le FTB, un verbe causatif est traité comme auxiliaire de l'infinitif complément. Cet infinitif est considéré comme tête du noyau verbal et comme gouverneur de tous les arguments. L'annotation en dépendances suivant le guide du FTB de la phrase (1) est présentée ci-dessous.

(1) *Il fait descendre le secrétaire*



Cette approche est un peu trop simplificatrice et ne permet pas de prendre en compte certains aspects. Dans l'exemple ci-dessus, « *secrétaire* » est objet de « *descendre* » mais c'est ambigu : si par exemple le secrétaire est une personne, « *secrétaire* » peut être le sujet profond de « *descendre* » ; s'il s'agit d'un meuble, ou si « *descendre* » est

employé avec le sens de « *tuer* », « *secrétaire* » est alors objet profond du verbe transitif « *descendre* ». Afin de distinguer les deux lectures, notre système de réécriture transforme l'auxiliaire du causatif en verbe plein, ce qui permet d'exprimer les deux lectures à travers les deux annotations ci-dessous.



Dans les deux schémas ci-dessus, a été introduite une dépendance représentant le sujet profond de « *descendre* ». Dans le premier cas, il s'agit de « *secrétaire* » et dans le second, ce sujet n'est pas exprimé dans la phrase et il est représenté par un mot vide noté ϵ. D'une façon générale, les dépendances syntaxiques de surface sont représentées au-dessus du texte qu'elles annotent, alors que les dépendances profondes, qu'elles soient syntaxiques (notées avec un 'p' terminal : **sujp**, **objp**, ...) ou sémantiques, sont placées au-dessous du texte.

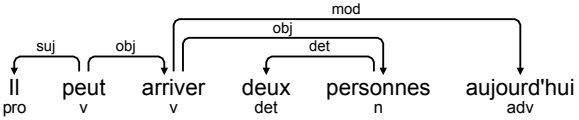
Les verbes à montée du sujet sont traités dans le FTB de la même façon que les verbes à contrôle comme des verbes pleins. Mais il y a de bonnes raisons (Rooryck, 1989) pour les traiter aussi comme des éléments hybrides qui se comportent parfois comme des auxiliaires. Considérons les exemples suivants où les verbes à montée sont indiqués en gras et où les infinitifs sont soulignés.

(2) *Il* **peut** *arriver deux personnes aujourd'hui*
(3) *La maison* **peut** *être vendue aujourd'hui*
(4) *La maison* **peut** *se vendre aujourd'hui*

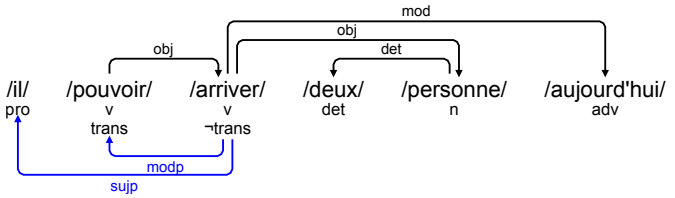
Nous avons successivement une construction impersonnelle, une construction à la voix passive et une construction à la voix moyenne. Si on veut éviter la multiplication du nombre des règles pour le calcul des arguments sémantiques, il est nécessaire de transformer chaque construction en une forme canonique. Classiquement, c'est la construction personnelle à la voix active. Pour nos trois exemples, on obtient :

(2') *Deux personnes* **peuvent** *arriver aujourd'hui.*
(3'),(4') *On* **peut** *vendre la maison aujourd'hui.*

Les règles de réécriture qui effectuent cette transformation sont grandement facilitées si les verbes à montée sont traités comme des auxiliaires. On peut alors appliquer les mêmes règles que pour les verbes non gouvernés par des verbes à montée. Détaillons la méthode sur l'exemple (2). Voici l'étiquetage de surface qui est donné au départ par le guide du FTB.

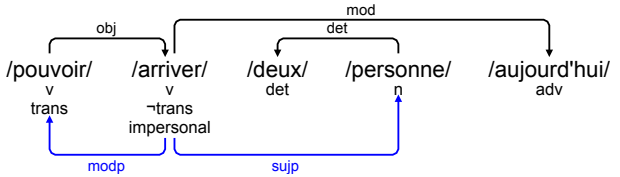


Une première règle transforme le verbe à montée « *pouvoir* » en auxiliaire de « *arriver* » qui devient la tête de la phrase.



Le fait que nous ne limitions pas la structure de dépendance à un arbre permet un plus grand pouvoir d'expression. Ainsi, le verbe *« pouvoir »* garde *« arriver »* comme objet tout en étant un modificateur de ce dernier, ce qui donne lieu à un cycle dans le graphe. Bien entendu, la tête syntaxique de la phrase n'est plus alors la racine d'un arbre de dépendance mais c'est elle qui reste destinée à recevoir toute dépendance externe, notamment quand la phrase est insérée comme proposition subordonnée dans une phrase complexe.

Maintenant, nous sommes en mesure de traiter la construction impersonnelle, qui est dorénavant ancrée à *« arriver »* comme n'importe quelle construction impersonnelle, et de lui appliquer une règle pour retrouver la construction canonique active correspondante. Cette règle consiste à supprimer le sujet impersonnel et l'objet de surface devient le sujet profond du verbe.



3.2 Ordonnancement des modules

L'organisation des règles de réécriture en modules est indissociable d'une définition d'un ordre partiel d'exécution de ces modules. Cet ordre partiel est déterminé par des choix linguistiques et des choix de représentation. Ainsi par exemple, le module de traitement des constructions impersonnelles précède celui du traitement de la voix passive, de façon à ce que soient correctement transformées les constructions passives impersonnelles.

Dans les premières version du système, cet ordre n'avait pas été étudié de façon systématique et certains choix simplificateurs posaient problème. En effet, le choix avait été fait d'enrichir en premier l'annotation du FTB en ajoutant les actants syntaxiques des infinitifs qui étaient présents dans la phrase. L'idée était d'enchaîner ensuite par une redistribution des actants pour effectuer le calcul des arguments sémantiques à partir d'une construction canonique. Or, en réalité, les choses peuvent être plus complexes. Dans chacune de ces phrases (5) et (6), on a un verbe à contrôle en gras qui gouverne le sujet d'un infinitif qui est souligné.

(5) *Jean est* **autorisé** *à* *arriver* *plus tard*
(6) *Jean* **permet** *à Marie d'* *être accompagnée* *par sa fille*

Pour la phrase (5), il est nécessaire d'appliquer en premier le module de redistribution du passif, noté PASSIVE_ARG dans notre système, pour retrouver la construction canonique (5'). Ensuite seulement, on est en mesure d'appliquer le module de détermination des sujets des infinitifs

dépendant de verbes à contrôle, noté INF_SUBJ. Dans ce module, une règle lexicale va indiquer que le sujet de *« arriver »* est l'objet direct de *« autorise »*.

(5') *On* **autorise** *Jean à arriver plus tard.*
(6') *Jean* **permet** *à Marie que sa fille l'accompagne*

Pour la phrase (6), c'est le contraire. Il faut d'abord appliquer le module INF_SUBJ pour trouver que le sujet de *« être accompagnée »* est l'objet indirect de *« permet »*. Ensuite, l'application du module PASSIVE_ARG permet de transformer la voix passive en voix active (6').

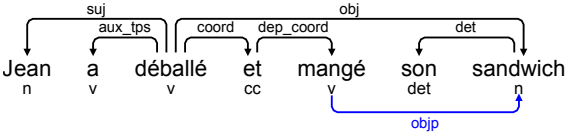
Ceci explique que l'on trouve dans l'ordre d'application des modules la séquence PASSIVE_ARG, INF_SUBJ, PASSIVE_ARG. On peut imaginer que cette suite de modules soit insuffisante et qu'il faille itérer l'aller-retour entre ces deux modules mais en pratique l'ordre ci-dessus est suffisant.

La coordination, de par sa spécificité, ne peut pas être traitée comme la plupart des constructions syntaxiques à l'aide d'un module particulier qui viendrait s'insérer quelque part dans le graphe d'ordonnancement des modules. Comme elle permet un partage de structures, elle interagit avec les autres constructions syntaxiques et cette interaction doit être modélisée par des règles qui sont distribuées dans les différents modules de calcul de la syntaxe profonde.

Bien entendu, chacune des règles est dépendante du choix qui a été fait pour annoter la coordination dans le FTB ; en particulier, le fait d'imposer que la structure de dépendance soit un arbre a de lourdes conséquences. Ainsi, on ne peut pas exprimer le partage de structures introduit fréquemment par la coordination, qui nécessite le partage de dépendants entre gouverneurs. Les règles de notre calcul vont donc consister essentiellement à retrouver le partage. Selon que la structure partagée est le sujet d'un verbe, un auxiliaire de temps ou du passif, ou encore l'antécédent d'un pronom relatif, le phénomène est traité respectivement dans le module d'introduction de sujets SUBJ_INTRO, celui de suppression des auxiliaires VERB_AUX, ou enfin dans celui de détermination de l'antécédent des pronoms relatifs ANT_REL_PRO.

Le partage de compléments est plus délicat à traiter dans la mesure où un complément n'est pas toujours obligatoire, ce qui rend l'annotation du FTB ambigüe. Considérons par exemple la phrase suivante annotée selon le guide du FTB.

(7) *Jean a déballé et mangé son sandwich*



Si on veut exprimer que *« sandwich »* est un objet partagé par *« déballé »* et *« mangé »*, on devrait ajouter la dépendance **objp** en dessous du dessin. Malheureusement ce n'est pas possible dans le FTB car le mot *« sandwich »* aurait alors deux gouverneurs. En conséquence, uniquement avec la structure de dépendance, on ne peut pas distinguer la phrase où *« son sandwich »* est un objet partagé de la phrase *« Jean a déballé son sandwich et mangé »*. Seul l'ordre linéaire des mots permettra de faire la différence. Dans notre système, nous avons choisi provisoirement de laisser ce problème de côté.

4 Description du système de règles de réécriture

Le système actuel comprend 458 règles, dont 325 règles lexicales, organisées en 35 modules. La Figure 1 présente le système de modules sous forme d'un graphe où les nœuds représentent les modules et les arcs les contraintes sur l'ordre d'application des modules. Seuls 9 des 35 modules sont non confluents ; sur le diagramme, ils se distinguent des autres parce qu'ils sont représentés par des ovales opaques.

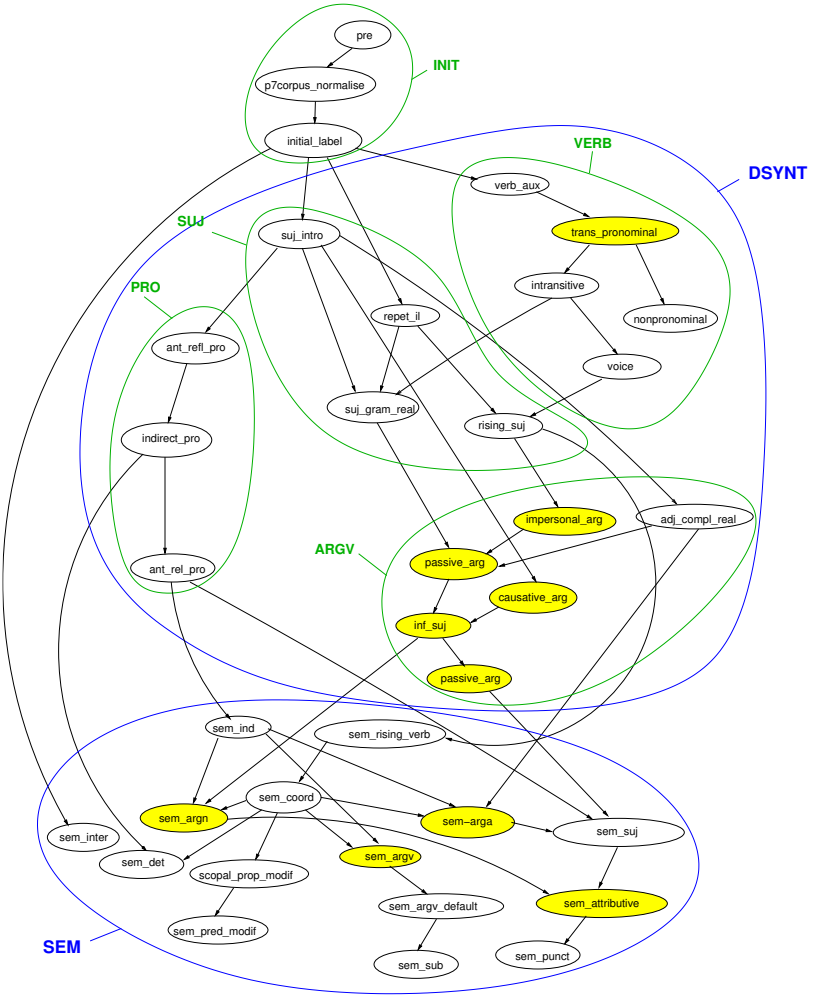


FIGURE 1 – Diagramme du système de modules

Ces modules sont eux-mêmes rangés dans 6 paquets :

- INIT qui transforme l'annotation CONLL initiale pour la mettre à un format compatible avec les règles de calcul de la syntaxe profonde,
- VERB dédié à l'uniformisation du noyau verbal par suppression des auxiliaires et à la

détermination du caractère transitif et pronominal des verbes et à la reconnaissance de leur voix (active, passive ou moyenne),
- SUBJ dédié au traitement du sujet des verbes et des adjectifs considérés comme prédicatifs,
- ARGV dédié au calcul des actants syntaxiques profonds des verbes, soit par redistribution des actants de surface (transformation des constructions impersonnelles et causatives et des voix passive et moyenne), soit par détermination lexicale de certains actants des infinitifs,
- PRO qui uniformise la syntaxe des différents pronoms et détermine l'antécédent des pronoms relatifs et des pronoms personnels réfléchis,
- SEM qui transforme l'annotation syntaxique profonde en annotation sémantique.

Les quatre paquets VERB, SUBJ, ARGV et PRO contribuent à la production de l'annotation des phrases en dépendances syntaxiques profondes. C'est pourquoi elles sont rassemblées dans un même super-paquet DSYNT. Ensuite, à partir de celle-ci, les règles du paquet SEM calculent une représentation sémantique de la phrase dans le format DMRS.

Le FTB ne suit pas complètement le guide d'annotation et les mêmes phénomènes ne sont pas annotés dans tout le corpus de façon cohérente. Notre système de modules tient compte de ces aspects et tente des récupérer certains de ces problèmes d'annotation qui sont les plus systématiques (comme par exemple des relations **p-obj** étiquetées **dep** ou des incohérences dans les coordinations de complétives).

(8) *Jean a pu être autorisé à partir et à regagner son domicile*

La Figure 2 illustre, pour la phrase (8), les deux étapes de calcul avec les trois structures :

- l'annotation initiale en dépendances syntaxiques de surface effectuée conformément au guide d'annotation du FTB ;
- l'annotation en dépendances profondes obtenue après exécution des règles des paquets INIT, VERB, SUBJ, ARGV et PRO ;
- la représentation sémantique en DMRS après exécution des règles du paquet SEM.

5 Résultats expérimentaux

Le système de modules présenté à la section précédente a été appliqué aux 12 351 phrases du FTB. Les statistiques complètes d'utilisation des règles par module sur l'ensemble du corpus et les résultats détaillés (avec les structures produites) sur 1% sur corpus sont accessible en ligne[4].

Le calcul étant non confluent, on s'intéresse en premier lieu au nombre de formes normales produites par notre système : on peut en effet avoir pour certaines longues phrases une forte ambiguïté. De plus, à la fin de certains modules, on filtre des structures que l'on considère comme incohérentes. Il peut donc arriver que certaines phrases ne conduisent à aucune forme normale : dans l'ensemble du corpus, c'est le cas pour 12,8% des phrases. Pour 31,3% des phrases, le système retourne une forme normale, pour 21,7%, il retourne 2 formes normales. 90,3% des phrases conduisent à un nombre de formes normales inférieur ou égal à 8.

Pour estimer la couverture de notre système nous avons observé le nombre de relations de dépendances présentes dans le FTB qui ne sont pas traitées lors de la réécriture. Dans le tableau

[4] `http://wikilligramme.loria.fr/doku.php?id=taln_2012`

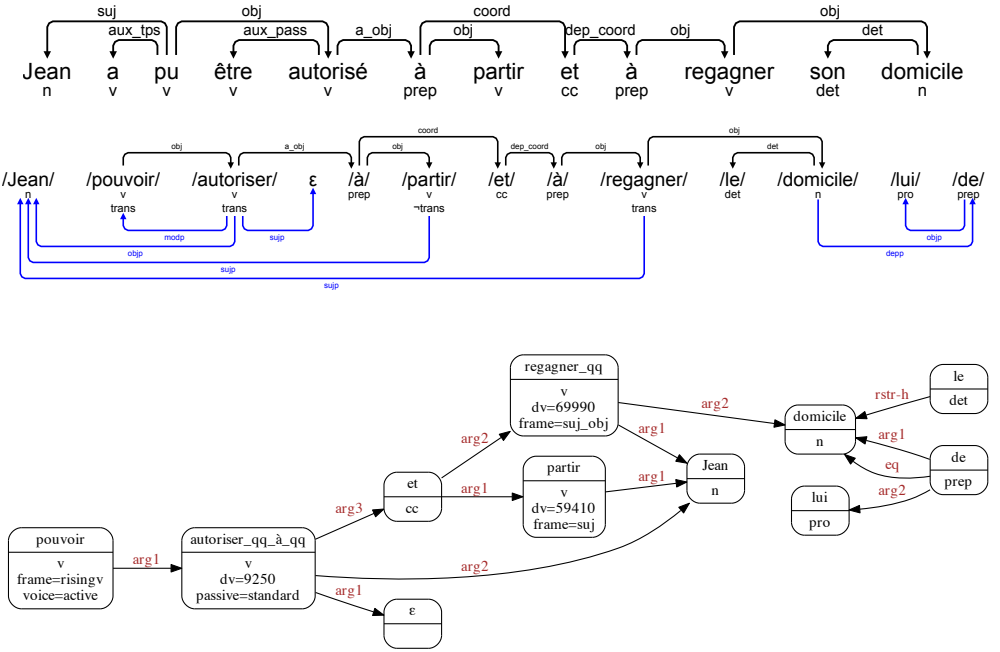


FIGURE 2 – Syntaxe de surface, syntaxe profonde et sémantique pour la phrase (8).

suivant, pour les phrases qui ont au moins une forme normale et pour chaque type de lien de dépendance du FTB, nous donnons le nombre total de liens au départ et le nombre qui restent après réécriture (en valeur absolue et relative) :

aff	**arg**	**dep**	aux_caus	aux_tps	aux_pass	comp	coord	**dep_coord**	det
1 581	465	32 821	128	3 806	1 672	37	6 358	7 299	40 355
91	375	2 331	5	27	15	37	203	1 251	608
5,8%	80,6%	6,8%	3,9%	0,7%	0,9%	100,0%	3,2%	17,1%	1,5%

mod	**mod_rel**	ponct	suj	**obj**	a_obj	de_obj	p_obj	ato	ats
58 353	2 352	35 810	15 100	58 132	2 192	1 668	1 663	160	2 521
10 452	427	1 561	162	3 222	16	23	4	15	15
17,9%	17,7%	4.3%	1,0%	5,5%	0,7 %	1,4%	0,2%	9,4%	0,6%

Il est difficile de séparer les cas qui proviennent d'une erreur d'annotation des cas qui se sont pas gérés par notre système. Néanmoins, pour les relations pour lesquelles il reste un nombre important de liens, et pour les cas où l'annotation est correcte, nous pouvons donner quelques commentaires. Les comparatifs ne sont pas traités par notre système, la relation **comp** n'est donc pas réécrite. La relation **arg** n'a pas été traitée. Les liens **mod** restant concernent les noms qui modifient un nom ou un verbe et les liens **mod_rel** peuvent correspondre à des cas d'ellipse dans la relative. Pour la coordination, il n'est pas rare d'avoir plusieurs liens **dep_coord** qui ont le même gouverneur. Le lien **dep** est sous-spécifié, il est utilisé dans des contextes assez différents et pas tous prévus par le guide d'annotation ou par notre système. Pour finir, la relation **obj** est utilisé pour les objets directs de verbe mais aussi pour le lien entre une préposition et le terme

qu'elle introduit ; c'est ces dernières relations qui peuvent rester.

Par ailleurs, pour estimer la couverture de notre usage de Dicovalence, nous donnons des statistiques sur les verbes repérés. Le FTB complet contient 39 108 verbes et les phrases pour lesquelles on fournit au moins une forme normale contiennent 32 255 verbes. Chaque fois que la réécriture réussit, on s'intéresse au nombre de verbes qui sont associés à une entrée de Dicovalence dans au moins une des structure finales. C'est le cas pour 17 987 verbes ; cela représente 55,8% des 32 255 verbes réécrits et 46,0% de l'ensemble des 39 108 verbes.

Pour 120 phrases du corpus, les résultats complets sont disponibles. Parmi elles, 11 phrases ne conduisent à aucune forme normale. Pour 5 de ces 11 phrases, Dicovalence ne décrit pas le cadre nécessaire : *hispaniser* transitif, *acheter* avec à-objet et sans objet, *maintenir* avec un complément locatif, *souscrire* transitif et *vendre* intransitif. Les autres cas correspondent à des problèmes d'annotation.

Plus précisément, nous avons évalué manuellement les résultats pour les 30 premières phrases des 120. Lorsque notre système fournit pour l'une d'elles plusieurs annotations sémantiques en sortie, nous avons considéré uniquement celle qui se rapproche le plus de l'annotation souhaitée. Pour 3 phrases, le système ne fournit aucune annotation en sortie et pour 3 autres, il fournit une annotation sans erreurs. Pour le reste, nous avons noté 67 erreurs qui se répartissent ainsi : 21 sont dues à des annotations incorrectes ou insuffisantes dans le FTB au départ ; dans 23 autres cas, le système échoue à produire une annotation sémantique parce qu'il ne couvre pas certains phénomènes (les 8 cas les plus fréquents concernent des expressions entre parenthèses insérées au milieu d'une phrase) ; pour les 23 cas restants, le système produit une annotation sémantique mais celle-ci n'est pas correcte ; les deux erreurs les plus fréquentes concernent les expressions figées (7 cas) qui ne sont pas considérées comme telles au niveau sémantique et les négations (6 cas) qui sont exprimées généralement par un couple de mots grammaticaux mais pour lesquels le système ignore le lien entre les deux éléments du couple.

Conclusion

A travers l'exemple du FTB, nous avons montré qu'il est pertinent d'utiliser la réécriture de graphes pour annoter de façon automatique un gros corpus en dépendances sémantiques à partir d'une annotation en dépendances syntaxiques de surface.

Les modules dans le système de règles de réécriture jouent un rôle majeur tant pour contrôler les calculs que pour construire et maintenir le système. Celui qui a été présenté dans cet article est conçu en fonction des formats d'entrée et de sortie choisis. Toutefois le caractère modulaire du système fait qu'il est possible de l'adapter à peu de frais à d'autres formats.

Pour en rester au FTB, l'annotation obtenue présente deux limites principales : l'annotation de départ présente de nombreuses erreurs et incohérences et le cadre formel choisi pour l'annotation sémantique, celui de la DMRS, n'offre aucune réponse quant à la modélisation de certaines propriétés sémantiques (l'intentionnalité ou les constructions comparatives par exemple). Pour ce qui est de la première limite, on peut espérer utiliser la réécriture de graphes pour corriger les erreurs qui sont les plus systématiques. Pour ce qui est de la seconde, nous ne pouvons que nous en remettre aux linguistes travaillant sur ces questions.

Références

Abeillé, A., Clément, L. et Toussenel, F. (2003). *Building a Treebank for French*, chapitre 10. Kluwer Academic Publishers.

Bonami, O., Godard, D. et Kampers-Manhe, B. (2004). Adverb classification. *In Handbook of French Semantics*, chapitre 11, pages 143–184. CLSI Publications.

Bonfante, G., Guillaume, B., Morey, M. et Perrier, G. (2010). Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique. *In Actes de TALN 2010 (Traitement automatique des langues naturelles)*, Montréal. ATALA.

Bonfante, G., Guillaume, B., Morey, M. et Perrier, G. (2011). Modular graph rewriting to compute semantics. *In IWCS 2011*, pages 65–74, Oxford, UK.

Candito, M.-H., Crabbé, B., Denis, P. et Guérin, F. (2009). Analyse syntaxique statistique du français : des constituants aux dépendances. *In Actes de TALN 2009 (Traitement automatique des langues naturelles)*, Senlis. ATALA, LIPN.

Copestake, A. (2009). *Invited Talk :* Slacker semantics : Why superficiality, dependency and avoidance of commitment can be the right way to go. *In Proceedings of EACL 2009*, pages 1–9, Athens, Greece.

Dixon, R. W. et Aikhenvald, A. Y., éditeurs (2006). *Adjective Classes - a Cross-Linguistic Typology*. Oxford University Press.

Giry-Schneider, J. (1987). *Les prédicats nominaux en français*. Librairie DROZ Genève-Paris.

Morey, M. (2011). *Étiquetage grammatical symbolique et interface syntaxe-sémantique des formalismes grammaticaux lexicalisés polarisés*. Thèse, Université de Lorraine.

Partee, B. H. (2010). Privative adjectives : subsective plus coercion. *In* Bäuerle, R. et Zimmermann, T. E., éditeurs : *Presuppositions and Discourse : Essays Offered to Hans Kamp*, pages 273–285. Bingley, UK : Emerald Group Publishing.

Rooryck, J. (1989). Les verbes à montée et à contrôle "ambigus". *Revue québécoise de linguistique*, 18(1):189–206.

Van den Eynde, K. et Mertens, P. (2003). La valence : l'approche pronominale et son application au lexique verbal. *French Language Studies*, 13:63–104.