SemTAG, une architecture pour le développement et l'utilisation de grammaires d'arbres adjoints à portée sémantique

Claire Gardent[1] Yannick Parmentier[2]
(1) CNRS / LORIA
Campus scientifique - BP 259
F - 54 506 Vandœuvre-Lès-Nancy CEDEX
(2) INRIA / LORIA - Nancy Universités
Campus scientifique - BP 259
F - 54 506 Vandœuvre-Lès-Nancy CEDEX
{gardent | parmenti}@loria.fr

Résumé Dans cet article, nous présentons une architecture logicielle libre et ouverte pour le développement de grammaires d'arbres adjoints à portée sémantique. Cette architecture utilise un compilateur de métagrammaires afin de faciliter l'extension et la maintenance de la grammaire, et intègre un module de construction sémantique permettant de vérifier la couverture aussi bien syntaxique que sémantique de la grammaire. Ce module utilise un analyseur syntaxique tabulaire généré automatiquement à partir de la grammaire par le système DyALog. Nous présentons également les résultats de l'évaluation d'une grammaire du français développée au moyen de cette architecture.

Abstract In this paper, we introduce a free and open software architecture for the development of Tree Adjoining Grammars equipped with semantic information. This architecture uses a metagrammar compiler to facilitate the grammar extension and maintnance, and includes a semantic construction module allowing to check both the syntactic and semantic coverage of the grammar. This module uses a tabular syntactic parser generated automatically from this grammar using the DyALog system. We also give the results of the evaluation of a real-size TAG for French developed using this architecture.

Mots-clefs : Analyseur syntaxique, Grammaires d'Arbres Adjoints, Construction sémantique, Architecture logicielle

Keywords: Syntactic parser, Tree Adjoining Grammars, Semantic construction, Software architecture

1 Introduction

Un objectif central du traitement automatique des langues est de construire une représentation du sens des textes afin de pouvoir raisonner sur leur contenu. Suivant la granularité de sens désirée, plusieurs approches sont possibles. Typiquement, la recherche d'information s'appuie sur une représentation « à gros grain » où le sens d'un texte est un « sac de mots » (cf. 1a) ; l'extraction d'information demande une représentation plus fine où en particulier les relations sémantiques entre (sens de) constituants doivent être spécifiées (cf. 1b) ; et les systèmes de dialogue, systèmes questions-réponses ou systèmes de détection d'implications textuelles, s'appuient souvent sur une représentation dite « profonde » où des phénomènes tels que la quantification et les modalités pourront être pris en compte (cf. 1c).

(1) L'homme regarde souvent la maison
   a. { *homme, regarde, maison* }
   b. *homme(h), regarde(h,m), maison(m)*
   c. $\exists x \exists y \exists e$.*homme(x)* $\wedge$ *souvent(e)* $\wedge$ *regarde(e,h,m)* $\wedge$ *maison(m)*

Pour construire le troisième type de représentation c.-à-d., une représentation profonde, une approche communément adoptée est de suivre Montague (Montague, 1974) et de développer des grammaires et des lexiques permettant une sémantique compositionnelle c'est-à-dire, une sémantique où le sens d'un constituant est une fonction de la syntaxe de ce constituant et du sens de ses sous-constituants. Ainsi, les grammaires syntagmatiques guidées par les têtes (HPSG, (Copestake *et al.*, 2005)) intègrent une sémantique basée sur les structures à recursion minimale (MRS), les grammaires lexicales fonctionnelles (LFG, (Frank & Van Genabith, 2001)) couplent la construction syntaxique avec une construction sémantique basée sur la sémantique « colle » (glue semantics) et les grammaires catégorielles combinatoires (CCG, (Bos *et al.*, 2004)) utilisent l'isomorphisme de Curry-Howard pour associer de façon systématique, constituants syntaxiques et termes lambda. Pour chacune de ces grammaires, une implantation existe qui démontre la faisabilité de l'approche théorique sous-jacente et en permet l'utilisation pratique dans des systèmes de TAL.

Une exception notoire concerne la construction sémantique dans les grammaires d'arbres adjoints (Joshi *et al.*, 1975). Pour ces grammaires en effet, des propositions théoriques existent mais aucune implantation. Dans cet article, nous reprenons la proposition théorique avancée par (Gardent & Kallmeyer, 2003) et décrivons sa mise en oeuvre dans un système implanté. Nous présentons les différentes composantes du système (grammaire, compilateur de grammaire, module de construction sémantique) et donnons les résultats d'une première évaluation sur une grammaire noyau du français. Utilisé pour développer une grammaire d'arbres adjoints à dimension sémantique pour le français, ce système est à notre connaissance, le premier système logiciel libre permettant la construction de représentations sémantiques profondes pour le français. En effet, il existe une grammaire HPSG pour le français (Tseng, 2003) mais sa couverture est limitée. Une grammaire LFG existe également mais étant développée par Xerox, elle n'est pas disponible pour la recherche. Par contraste, SEMTAG est un logiciel libre et ouvert. Le logiciel de développement est disponible à l 'url `http://trac.loria.fr/~semtag` avec une grammaire jouet. La grammaire est accessible sur demande et sera rendue disponible prochainement.

L'article est structuré de la façon suivante. Nous commençons (section 2) par présenter le modèle linguistique utilisé c.-à-d., les grammaires d'arbres adjoints, la sémantique plate à trous et l'interface syntaxe/sémantique. Nous présentons ensuite brièvement (section 3) la grammaire

du français utilisée et donnons quelques chiffres sur sa couverture actuelle. Dans la section 4, nous présentons le module de construction sémantique. Enfin, la section 5 donne les résultats d'une première évaluation du système en termes de couverture et d'ambiguïté syntaxique et sémantique.

2 Modèle linguistique

Le modèle linguistique inclut une grammaire d'arbres adjoints, un langage de représentation sémantique et une modélisation de l'interface syntaxe/sémantique. Les restrictions d'espace nous empêchant de décrire chacune de ces composantes en détail, nous renvoyons le lecteur aux publications sources pour plus de détails.

Formalisme syntaxique : les grammaires d'arbres adjoints (TAG) Les grammaires d'arbres adjoints (Tree Adjoining Grammars, TAG) (Joshi *et al.*, 1975) appartiennent à la famille des grammaires légèrement sensibles au contexte. Une TAG est un système de réécriture d'arbres composé de deux ensembles d'arbres (*arbres initiaux* et *arbres auxiliaires*) et de deux opérations de réécriture (*substitution* et *adjonction*).

Un arbre initial est un arbre dont les noeuds feuilles sont soit étiquetés par des mots, soit des noeuds de substitution (marqués $\downarrow$) c.-à-d., des noeuds où une substitution *doit* prendre place. Un arbre auxiliaire est un arbre contenant un noeud pied (marqué $\star$) – ce noeud pied doit être étiqueté avec la même catégorie que le noeud racine.

Dans la version de TAG que nous utilisons, à savoir les grammaires d'arbres adjoints lexicalisées à structures de traits (FLTAG, (Vijay-Shanker & Joshi, 1988)), les arbres élémentaires sont lexicalisés, c'est-à-dire que pour chaque arbre, au moins un terminal est un lemme ou une forme fléchie. En outre, les noeuds des arbres sont étiquetés par deux structures de traits appelées TOP et BOTTOM. En fin de dérivation, les traits TOP et BOTTOM de chaque noeud sont unifiés.

L'opération de substitution permet d'insérer un arbre élémentaire ou dérivé τ_δ à la frontière d'un arbre initial τ_α : le noeud racine de τ_δ est alors identifié avec un noeud de substitution dans τ_α et les traits TOP des nœuds en question sont unifiés ($Top_{\tau_\alpha} = Top_{\tau_\delta}$). L'opération d'adjonction permet d'insérer un arbre auxiliaire τ_β dans un arbre quelconque τ_α à un noeud n : les traits TOP_n et BOTTOM_n du noeud n où se fait l'adjonction sont alors unifiés respectivement avec les traits TOP du noeud racine de l'arbre auxiliaire et les traits BOTTOM de son noeud pied ($Top_n = Top_{Root_{\tau_\beta}}$ et $Bottom_n = Bottom_{Foot_{\tau_\beta}}$).

Formalisme sémantique : la sémantique plate à trous (Hole Semantics) Comme la MRS mentionnée en section 1, le formalisme des sémantiques plates à trous (Bos, 1995) se caractérise par deux points importants. Premièrement, le formalisme permet de sous-spécifier les ambiguïtés de portée – ainsi les interprétations multiples dues à ces ambiguïtés peuvent être représentées de façon compacte. Deuxièmement, (Copestake *et al.*, 2005) ont montré que la structure non récursive des formules plates facilite la réalisation sémantique c.-à-d., la procédure qui permet de produire, à partir d'une représentation sémantique donnée, l'ensemble des phrases associées par la grammaire à cette sémantique. C'est là un point important puisque de fait, la grammaire présentée ici est également utilisée pour la réalisation.

Très brièvement (cf. (Gardent & Kallmeyer, 2003) pour plus de détails), le langage de représentation sémantique L_U utilisé est une reformulation de la logique PLU (Bos, 1995) qui inclut des variables d'unification. Soit I_{var} un ensemble de variables d'unification et I_{con} un ensemble de constantes. Soit H un ensemble de constantes « trous », L_{con}, un ensemble de constantes « étiquettes » et L_{var} un ensemble de variables d'étiquettes ; soit R un ensemble de relations n-aires sur $I_{var} \cup I_{con} \cup H$; et soit $\geq$ une relation sur $H \cup L_{con}$ nommée « a-portée-sur ». Alors, la syntaxe de L_U est la suivante :

Etant donnés $l \in L_{var} \cup L_{con}$, $h \in H$, $i_1, \ldots, i_n \in I_{var} \cup I_{con} \cup H$ et $R^n \in R$. Alors :

1. $l : R^n(i_1, \ldots, i_n)$ est une formule de L_U
2. $h \geq l$ est une formule de L_U
3. ϕ, ψ est une formule de L_U si ψ est une formule de L_U et ϕ est une formule de L_U
4. Rien d'autre n'est une formule de L_U

En d'autres termes : les formules de L_U sont soit des prédications élémentaires, soit des contraintes de portée, soit des conjonctions de formules. Sémantiquement, ces formules décrivent (ont pour modèle) des formules de la logique du premier ordre.

Par exemple, la représentation sémantique de la phrase « Tout yogi a un guru » est :

(2) $l_0 : \forall(x, h_1, h_2), h_1 \geq l_1, l_1 : Yo(x), h_2 \geq l_2, l_2 : A(x, y), l_3 : \exists(y, h_3, h_4), h_3 \geq l_4, l_4 : Gu(y), h_4 \geq l_2$

Cette formule a deux modèles reflétant les deux interprétations possibles de la phrase d'entrée : soit un même guru existe pour tous les yogis, soit plusieurs.

(3) $l_0 : \forall(x, l_1, l_3), l_1 : Yo(x), l_3 : \exists(x, l_4, l_2), l_4 : Gu(y), l_2 : A(x, y)$
$l_3 : \exists(x, l_4, l_0), l_4 : Gu(y), l_0 : \forall(x, l_1, l_2), l_1 : Yo(x), l_2 : A(x, y)$

Interface syntaxe / sémantique. L'interface entre grammaire et sémantique spécifie la correspondance entre constituants syntaxiques et constituants sémantiques. Cette spécification se fait conformément à la proposition de (Gardent & Kallmeyer, 2003). Chaque arbre élémentaire de la grammaire TAG est associé à une formule sémantique plate où des variables d'unification sont utilisées pour représenter les arguments sémantiques. Ces variables d'unification sont partagées avec des variables apparaissant dans les structures de traits étiquetant les nœuds de l'arbre. Lors de la dérivation TAG, les structures de traits des arbres élémentaires sont unifiées (cf. *supra*), ce qui indirectement, entraîne l'unification des arguments sémantiques. La composition sémantique est ainsi prise en charge par l'opération d'unification inhérente au formalisme TAG. A l'issue de la dérivation, la représentation sémantique de l'arbre dérivé est obtenue en prenant la conjonction des formules élémentaires modulo les unifications ayant eu lieu.

Par exemple, pour la phrase « Jean aime vraiment Marie », la dérivation TAG correspondante est donnée dans la figure 1[1]. Lors de la substitution de l'arbre associé à *Jean* (τ_{Jean}) sur l'arbre associé au prédicat *aimer* (τ_{aimer}), le nœud racine de τ_{Jean} est unifié avec le nœud GN de τ_{aimer} représentant la fonction grammaticale sujet. Le nœud GN de l'arbre résultant contient alors une structure *Top* avec un trait *idx* de valeur x et une structure *Bottom* avec le même trait *idx* ayant la valeur j. A l'issue de la dérivation, les structures *Top* et *Bottom* étant unifiées, la variable x est liée à la constante j. De façon similaire, la variable y est liée à la constante m lors de la substitution de l'arbre τ_{Marie} sur τ_{aimer}. Enfin, l'adjonction de l'adverbe *vraiment* sur le nœud

[1] Les structures *top* sont notées en exposants et les structures *bot* en indices. Seuls les traits sémantiques pertinents pour l'exemple sont indiqués.

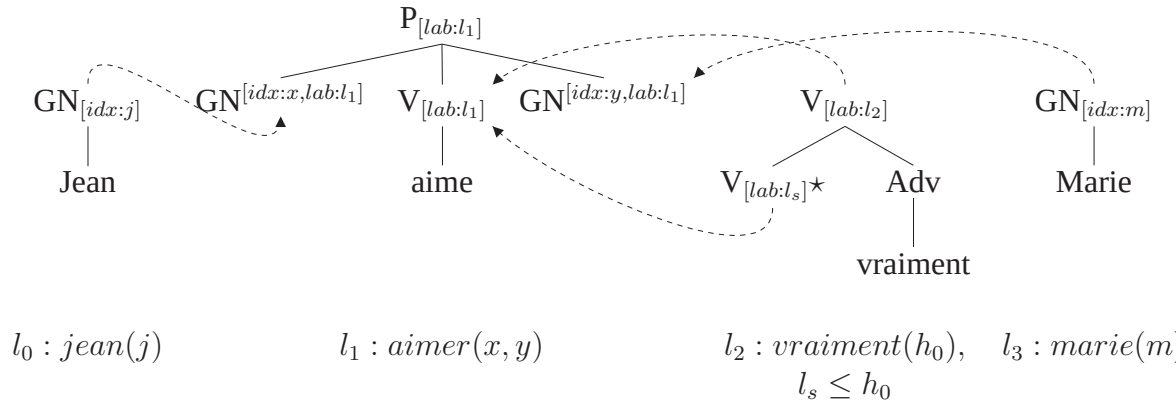

FIG. 1 – Dérivation TAG pour « Jean aime vraiment Marie »

de catégorie V de τ_{aimer} entraîne l'unification de la structure *Bottom* du nœud pied de $\tau_{vraiment}$ avec la structure *Bottom* du nœud d'étiquette V en question, ce qui provoque l'unification de la variable l_s avec la constante l_1. Ainsi, après dérivation et unifications correspondantes, la conjonction des formules sémantiques élémentaires nous donne le résultat escompté, à savoir la représentation sémantique sous-spécifiée suivante :

$$l_0 : jean(j), l_1 : aime(j, m), l_2 : vraiment(h_0), l_1 \leq h_0, l_3 : marie(m)$$

3 Grammaire informatique

La grammaire SEMFRAG est une implantation du modèle linguistique présenté ci-dessus. Spécifiée à l'aide du formalisme XMG (Duchier *et al.*, 2005), cette grammaire est produite par compilation à partir d'une spécification linguistique relativement abstraite et fortement factorisée. La composante syntaxique de la grammaire a été décrite dans (Crabbé, 2005) et la composante sémantique par (Gardent, 2006). Brièvement, l'intégration de l'information sémantique dans une grammaire TAG est facilitée par deux points.

Premièrement, la décoration des arbres élémentaires avec les variables nécessaires à un traitement à grande échelle de la sémantique obéit à un ensemble de principes limités en nombre et relativement rapides à implanter dans le formalisme XMG grâce au haut degré de factorisation permis par ce formalisme. Ces principes sont explicités dans (Gardent, 2007).

Deuxièmement, l'expressivité de XMG facilite la spécification de l'interface syntaxe/sémantique et plus spécifiquement, du partage des variables d'unification entre formules sémantiques et arbres élémentaires. En effet, XMG permet de gérer de manière flexible la portée des variables d'unification manipulées au sein des classes spécifiées par le linguiste. En particulier, ces classes peuvent être associées à des matrices de traits appelées *interfaces* qui sont unifiées lorsque deux fragments sont combinés conjonctivement ou par héritage. Indirectement, cela permet d'unifier des variables introduites dans différentes classes et en particulier, des variables introduites dans des classes syntaxiques (fragments d'arbres) d'une part et dans des classes sémantiques (formules de sémantique plate) d'autre part. Cette fonctionnalité du formalisme nous permet d'encoder de manière relativement aisée l'interface syntaxe / sémantique au niveau métagrammatical, en utilisant la méthodologie suivante :

– chaque fragment d'arbre contenant un nœud lié à une fonction grammaticale représentant un argument sémantique, se voit associé un trait *idx* dont la valeur correspond à une variable partagée avec un trait de l'interface, nommé *FGidx* (où *FG* correspond à la fonction grammaticale en question),
– chaque foncteur sémantique est associé avec une formule sémantique où les arguments sont des variables partagées avec des traits de l'interface. Ces traits sont nommés en fonction du rôle thématique de l'argument (*p. ex. arg0* ...),
– enfin, dans la règle de combinaison de ces fragments (munie également d'une interface), on ajoute dans l'interface une coindexation entre *FGidx* et l'argument sémantique correspondant (ce qui nous permet également de gérer le cas du passif).

Ce procédé est illustré figure 2, en prenant l'exemple d'un verbe intransitif[2].

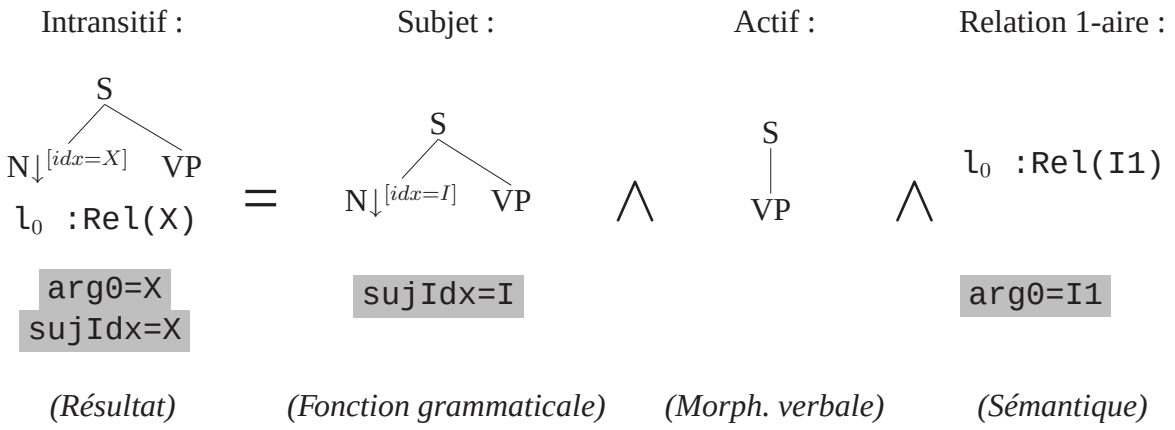

FIG. 2 – Interface syntaxe / sémantique au niveau métagrammatical.

Comme dans le système XTAG, la grammaire SEMFRAG se décompose en 3 sous-modules : un module contenant des d'arbres non lexicalisés groupés en familles[3] ; un lexique de lemmes associant à chaque lemme un prédicat sémantique et une ou plusieurs familles d'arbres ; et un lexique de formes fléchies associant à chaque forme fléchie, un lemme et l'information morpho-syntaxique appropriée. Lors de l'analyse, ces trois modules sont consultés pour associer à chaque mot m de la phrase analysée un arbre lexicalisé (c.-à-d., ancré avec m) dont la sémantique inclut le prédicat spécifié pour m par le lexique de lemme.

4 Construction sémantique

La grammaire SEMFRAG *décrit* l'association entre constituants syntaxiques et représentations sémantiques. Comme le montrent (Gardent & Parmentier, 2005), pour *calculer* cette association (c.-à-d., pour produire la (ou les) représentations sémantique(s) associée(s) par la grammaire à une expression langagière), deux options sont possibles : soit la construction sémantique est intégrée dans l'analyse syntaxique (la construction sémantique se fait pendant la dérivation), soit elle se fait après la dérivation sur la base de cette dérivation et d'un lexique sémantique produit à partir de la grammaire et d'un lexique.

[2]On remarque que la fonction grammaticale pourrait très bien correspondre à une disjonction des différentes réalisations syntaxiques.

[3]En TAG, une famille d'arbres regroupe tous les arbres élémentaires correspondant à un cadre de sous catégorisation donné *p. ex.*, intransitive.

SEMTAG implante la deuxième option, ce qui permet à la fois de rester dans le formalisme TAG (*cf* (Kallmeyer & Romero, 2004)) et de garder une approche modulaire où analyse syntaxique et construction sémantique restent indépendants l'un de l'autre[4]. Concrètement, le procédé de construction sémantique repose sur le schéma suivant.

Etape 1. Dans un premier temps, toute l'information sémantique incluse dans la grammaire est extraite et stockée dans un lexique sémantique. Ce lexique est en quelque sorte le parallèle sémantique de la grammaire syntaxique TAG au sens où il associe à chaque arbre élémentaire TAG un arbre sémantique correspondant. La figure 3 illustre ce procédé d'extraction pour l'arbre associé à la forme fléchie « dort » (arbre pour une utilisation avec un sujet nominal canonique). L'arbre du haut est celui produit par la compilation de SEMFRAG (suivie de la phase d'ancrage des schémas d'arbres par l'information contenue dans les lexiques de lemmes et de formes fléchies), l'arbre en bas à gauche est l'arbre purement syntaxique extrait de cet arbre et l'arbre en bas à droite, l'arbre sémantique (entrée du lexique sémantique)[5].

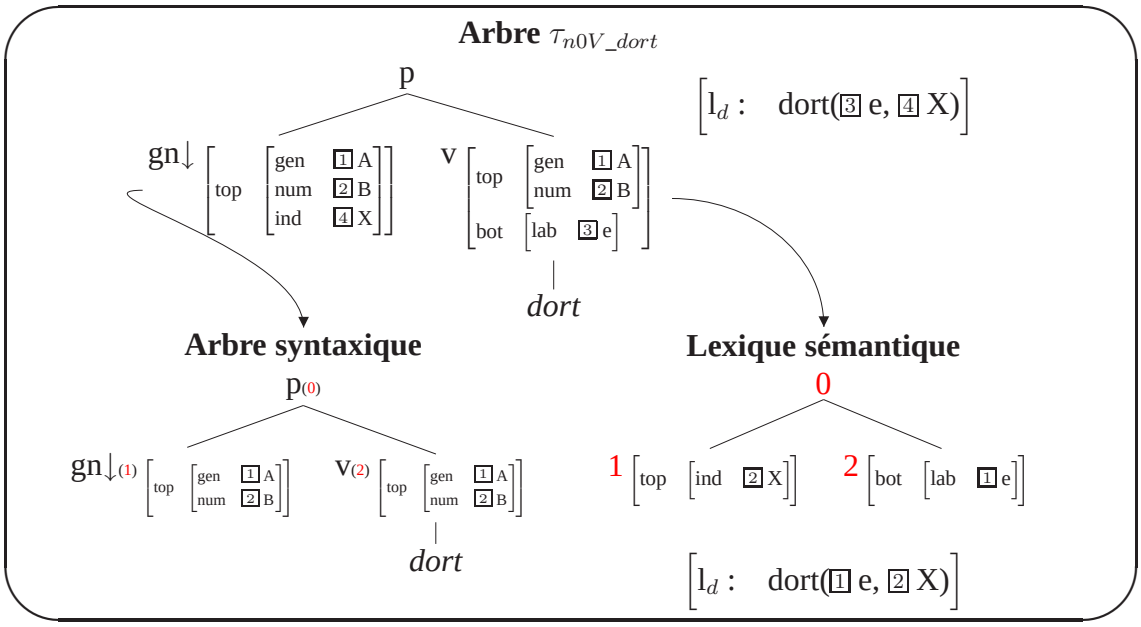

FIG. 3 – Entrée du lexique sémantique.

Etape 2. La deuxième étape consiste à faire une analyse syntaxique de la phrase d'entrée en utilisant uniquement la partie syntaxique de SEMFRAG. Cette analyse est réalisée au moyen du système DyALog (Villemonte de la Clergerie, 2005), un compilateur de programmes logiques avec tabulation des calculs intermédiaires qui permet en particulier de compiler un analyseur syntaxique à partir d'une grammaire TAG donnée. L'analyseur résultant de cette compilation prend en entrée une chaîne préalablement segmentée, et retourne une forêt de dérivation décrivant de façon compacte l'ensemble des dérivations couvrant la chaîne d'entrée. Par exemple, la forêt de dérivation pour la phrase ambiguë « Jean regarde Anne avec le télescope » est celle

[4]Cette implantation correspond à la proposition de (Kallmeyer & Romero, 2004), l'avantage étant que les structures étiquetant les nœuds ne contiennent pas de traits à valeur en nombre théoriquement infini (*p. ex.*, les variables de label de la sémantique plate).

[5]Le lexique sémantique est donc calculé par rapport aux arbres ancrés lors de l'analyse syntaxique.

donnée en figure 4. Cette forêt représente les deux dérivations possibles de la façon suivante. Les nœuds de l'arbres sont étiquetés avec les noms des arbres élémentaires mis en jeu dans la dérivation tandis que les arcs indiquent soit une substitution (trait plein), soit une adjonction (trait en pointillés). Plus précisément, une flèche étiquetée avec l'information $\langle O, n\rangle$ et allant du nœud étiqueté X vers le nœud étiqueté Y, indique que l'arbre X a été combiné par l'opération O ($O \in \{s, a\}$ avec s pour substitution et a pour adjonction) avec l'arbre Y en son nœud n.

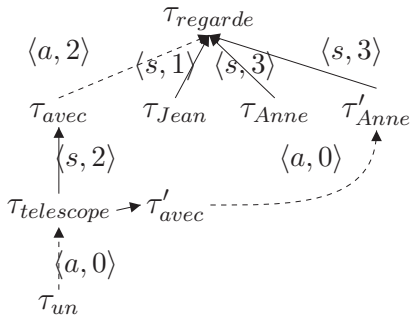

FIG. 4 – Forêt de dérivation de la phrase « Jean regarde Anne avec un télescope ».

Etape 3. Enfin la troisième étape consiste à produire à partir de la forêt de dérivation produite par DYALOG et du lexique sémantique extrait de SEMFRAG, la représentation sémantique de la phrase d'entrée. Pour ce faire, nous avons défini et implanté en Prolog un algorithme de construction sémantique qui traverse la forêt de dérivation dans un processus descendant, et réalise les unifications entre indices sémantiques comme résumé ci-dessous.

On note *Lex(x)* = (τ_x, ϕ_x) l'association spécifiée par le lexique sémantique entre un nom d'arbre syntaxique x, l'arbre sémantique τ_x et la représentation sémantique ϕ_x : *Lex*^{1}*(x)* = τ_x et *Lex*^{2}*(x)* = ϕ_x.
Etant donnée une forêt de dérivation et un lexique sémantique *Lex*, pour construire la (ou les) représentations sémantiques associées, FAIRE :

1. (Initialisation) Pour chaque racine(s) a de la forêt de dérivation, extraire *Lex(a)* = (τ_a, ϕ_a) du lexique sémantique *Lex*. Initialiser la sémantique de a à ϕ_a.
2. (Parcours descendant de la forêt) Pour chaque arc de dérivation de la forme $a_i \overset{o,n}{\rightarrow} a_j$ (où a_i, a_j sont des nœuds représentant des arbres élémentaires, o l'opération utilisée et n l'adresse de Gorn du noeud où a lieu l'opération dans l'arbre désigné par a_j), FAIRE :
   – ajouter ϕ_{a_j} (= *Lex*$^2(a_j)$) à la représentation sémantique ϕ_{a_i} de a_i
   – combiner τ_{a_j} (= *Lex*$^1(a_j)$) avec τ_{a_i} (= *Lex*$^1(a_i)$) conformément à l'opération spécifiée par o, n (les unifications correspondantes prennent place cf. section 2 instanciant par « ric hochet » les variables d'unification présentes dans les représentations sémantiques).
3. Lorsque toutes les dérivations ont été traitées, les structures *Top* et *Bottom* étiquetant chacun des nœuds des arbres sémantiques impliqués dans la dérivation sont unifiées.

En résumé : l'algorithme parcourt chaque arbre de la forêt de dérivation ; collecte la sémantique associée par le lexique sémantique avec chaque nœud (c.-à-d., arbre élémentaire) de cet arbre de dérivation ; et utilise les arbres sémantiques associés par le lexique sémantique aux noms d'arbres syntaxiques, pour retranscrire au niveau sémantique, les unifications correspondants aux opérations d'adjonction et de substitution réalisées au niveau syntaxique. Par exemple, pour la phrase « Jean court », l'algorithme procède comme suit :

– Initialisation : ϕ = { *l0 :courir(X)* }
– Traitement de l'arc $\tau_{Jean} \overset{s,1.0}{\rightarrow} \tau_{court}$:
  – Incrément de la sémantique : ϕ = { *l0 :courir(X), l1 :jean(j)* }
  – Effets des unifications dues à la substitution de l'arbre sémantique associé à τ_{Jean} dans l'arbre sémantique associé à τ_{court} : ϕ = { *l0 :courir(j), l1 :jean(j)* }

5 Evaluation

L'évaluation de cette architecture repose sur une évaluation de la grammaire du français qu'elle a permis de développer, en l'occurence la grammaire SEMFRAG présentée précédemment. Cette grammaire décrit 87 familles d'arbres (cadres de sous-catégorisation), les lexiques utilisés contiennent 1 471 formes fléchies, rattachées à 603 lemmes. L'évaluation consiste à vérifier les caractéristiques suivantes :
– la couverture syntaxique et sémantique sur une suite de tests combinant la *Test Suite for Natural Language Processing (TSNLP)* avec une suite de tests complémentaire (SEMTEST)[6],
– le taux moyen d'ambiguïté sémantique (nombre d'analyses sémantiques par phrase).
Développée dans les années 90s, la TSNLP (Lehmann *et al.*, 1996) est une suite de tests visant à permettre l'évaluation et la comparaison d'analyseurs syntaxiques sur un ensemble controlé et annoté de données. Sur un ensemble de 1 495 phrases tests, SEMTAG a actuellement une couverture syntaxique de 62.88 % et une couverture sémantique de 61.27 %. Le taux d'ambiguïté sémantique moyen est de 2.46.

Bien qu'elle ait été pensée pour une évaluation systématique des constructions syntaxiques, la TSNLP échoue à prendre en compte certains types de variations dont en particulier, les variations sur la réalisation des arguments (canonique, relatif, questionné, cliticisé, clivé, etc.), les variations sur la sous-catégorisation des verbes, les variations sur le type de verbe (verbes à contrôle, à montée, semi-auxiliaire, etc). Pour pallier ce manque, nous l'avons complémentée, avec une suite de phrases illustrant ces variations. Pour cette suite complémentaire, la couverture syntaxique est de 86.78 % et la couverture sémantique de 85.02 %. Le taux d'ambiguïté sémantique moyen est de 3.14.

6 Conclusion

SEMTAG permet d'associer à une phrase du français une représentation profonde de sa sémantique compositionnelle. Comme la section précédente l'a montré, la grammaire utilisée est insuffisante pour avoir une couverture large. Pour traiter du passage à échelle, il serait intéressant d'intégrer dans SEMTAG les techniques de fouilles d'erreur et d'analyse à partir d'arbres factorisés utilisées par (Sagot & Villemonte de La Clergerie, 2006). Par ailleurs, il importe d'évaluer la qualité et l'utilité des représentations sémantiques produites soit par le biais d'applications telles que la reconnaissance d'implications textuelles, soit par le biais de la génération (la sémantique produite permet elle de re-générer la phrase de départ ?).

[6]Par couverture, nous entendons la production d'une représentation syntaxique / sémantique validée manuellement dans un premier temps.

Références

BOS J. (1995). Predicate Logic Unplugged. In *Proceedings of the tenth Amsterdam Colloquium, Amsterdam*, p. 133–142.

BOS J., CLARK S., STEEDMAN M., CURRAN J. R. & HOCKENMAIER J. (2004). Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, p. 1240–1246, Geneva, Switzerland.

COPESTAKE A., FLICKINGER D., POLLARD C. & SAG I. A. (2005). Minimal Recursion Semantics : An introduction. *Research on Language and Computation*, **3.4**, 281–332.

CRABBÉ B. (2005). *Représentation informatique de grammaires fortement lexicalisées : Application à la grammaire d'arbres adjoints*. PhD thesis, Université Nancy 2.

DUCHIER D., LE ROUX J. & PARMENTIER Y. (2005). XMG : Un Compilateur de Méta-grammaire Extensible. In *Actes de TALN 2005, Dourdan, France*, p. 13–22.

FRANK A. & VAN GENABITH J. (2001). GlueTag - Linear Logic based Semantics for LTAG – and what it teaches us about LFG and LTAG –. In *Proceedings of LFG01, Hong Kong*, p. 104–126.

GARDENT C. (2006). Intégration d'une dimension sémantique dans les grammaires d'arbres adjoints. In *Actes de la conférence TALN 2006*, p. 149–158.

GARDENT C. (2007). Tree Adjoining Grammar, Semantic Calculi and Labelling Invariants. In *Proceedings of IWCS 7*, p. 75–85.

GARDENT C. & KALLMEYER L. (2003). Semantic construction in FTAG. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL'03), Budapest*, p. 123–130.

GARDENT C. & PARMENTIER Y. (2005). Large scale semantic construction for tree adjoining grammars. In *Proceedings of LACL05, Bordeaux, France*, p. 131–146.

JOSHI A., LEVY L. & TAKAHASHI M. (1975). Tree adjunct grammars. p. 136–163. Journal of Comput. Syst. Sci., Vol. 10-1.

KALLMEYER L. & ROMERO M. (2004). LTAG Semantics with Semantic Unification. In *Proceedings of TAG+7. Vancouver*, p. 155–162.

LEHMANN S., OEPEN S., REGNIER-PROST S., NETTER K., LUX V., KLEIN J., FALKEDAL K., FOUVRY F., ESTIVAL D., DAUPHIN E., COMPAGNION H., BAUR J., BALKAN L. & ARNOLD D. (1996). TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 1996*, p. 711–716, Kopenhagen.

MONTAGUE R. (1974). English as a formal language. *Formal Philosophy. Selected papers of Richard Montague, pages 188-221*.

SAGOT B. & VILLEMONTE DE LA CLERGERIE E. (2006). Error mining in parsing results. In *Proceedings of ACL 2006*, p. 329–336, Sydney, Australia.

TSENG J. (2003). Lkb grammar implementation : French and beyond. In E. B. ET AL, Ed., *Workshop on Ideas and Strategies for Multilingual Grammar Development*, p. 91–97, Technische Universität Wien.

VIJAY-SHANKER K. & JOSHI A. K. (1988). Feature structures based tree adjoining grammars. In *COLING*, p. 714–719.

VILLEMONTE DE LA CLERGERIE E. (2005). DyALog : a tabular logic programming based environment for NLP. In *Proceedings of CSLP'05*, p. 18–33, Barcelona.