Des arbres de dérivation aux forêts de dépendance : un chemin via les forêts partagées

Djamé Seddah - Bertrand Gaiffe
Laboratoire LORIA, Equipe Langue et Dialogue
Campus Scientifique, BP 239
F-54506 Vandœuvre-lès-Nancy Cedex
{djame.seddah,bertrand.gaiffe}@loria.fr

Mots-clefs : TAG, analyse syntaxique, sémantique, arbre de dépendance,forêt partagée, forêt de dérivation

Keywords: TAG, syntax, semantic, dependency tree, shared forest, derivation forest

Résumé L'objectif de cet article est de montrer comment bâtir une structure de répresentation proche d'un graphe de dépendance à l'aide des deux structures de représentation canoniques fournies par les Grammaires d'Arbres Adjoints Lexicalisées . Pour illustrer cette approche, nous décrivons comment utiliser ces deux structures à partir d'une forêt partagée.

Abstract This paper aims describing an approach to semantic representation in the Lexicalized Tree Adjoining Grammars (LTAG) paradigm : we show how to use all the informations contained in the two representation structures provided by the LTAG formalism in order to provide a dependency graph.

1 Introduction

Dans cet article[1], nous montrerons comment construire un graphe de dépendance dont les principales propriétés sont de matérialiser des liens syntaxiques non représentés dans l'arbre de dérivation et de décrire toutes les analyses dans une seule structure compacte.

1.1 Les grammaires d'arbres adjoints

Une grammaire LTAG est essentiellement un lexique où chaque lemme est associé à un ensemble d'arbres. Ces arbres, appelés arbres élémentaires, sont manipulables par deux opérations : la substitution et l'adjonction. La substitution opère sur un ensemble restreint d'arbres appelés *arbres initiaux* et correspond à une dérivation hors-contexte. Cette opération est obligatoire, contrairement, d'une façon générale, à l'adjonction qui opère sur des arbres appelés *auxiliaires* et qui correspond à l'insertion d'un arbre spécifique au sein d'un arbre élémentaire (indifféremment initial ou auxiliaire) .

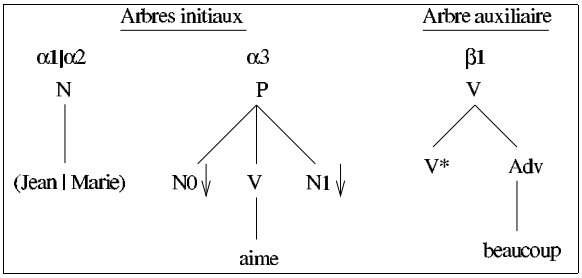


Figure 1: Arbres élémentaires d'une LTAG

Tous les arbres élémentaires sont des projections d'entrées lexicales et par définition décrivent tous les arguments syntaxiques des ancres associées. De fait quand une LTAG suit les principes de bonnes constructions (Abeillé, 1991) tels que le principe de co-occurence prédicat-argument et le principe de minimalité sémantique, les arguments syntaxiques correspondent à des arguments sémantiques. C'est pourquoi l'une des deux structures de représentation des LTAG, **l'arbre de dérivation**, qui formellement n'est que l'enregistrement des opérations résultant d'une analyse syntaxique, peut être vu comme une structure prédicat-argument.
La figure 2 nous montre que l'arbre de dérivation [2] reflète la structure prédicat-argument de la phrase *"Jean aime beaucoup Marie"*.

Étant donné que chaque nœud de l'arbre de dérivation correspond à une projection d'un arbre élémentaire et que chacune de ses branches décrit l'opération de combinaison entre deux nœuds, on associe à ces nœuds une adresse de Gorn[3] indiquant où l'opération a eu lieu. Ainsi, l'arbre de dérivation décrit de façon univoque **l'arbre dérivé** qui est l'arbre syntagmatique d'un énoncé (fig. 2).

[1] *Nous tenons à remercier vivement les reviewers de TALN 2005 pour leurs commentaires et leurs aimables corrections.*

[2] Pour des raisons de simplicité, chaque arbre dont le nom commence par β, resp. α, est un arbre auxiliaire, resp. initial. γ quand ce type est indifférent.

[3] Cette adresse correspond à une séquence d'entiers positifs définie par induction de la façon suivante : la séquence vide notée 0 est l'adresse du nœud racine de l'arbre et p.k est l'adresse du k-ième fils du nœud d'adresse p.

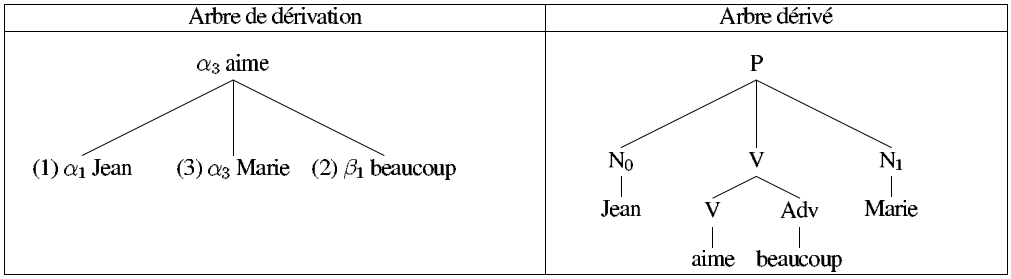


Figure 2: Arbre dérivé et arbre de dérivation pour *Jean aime beaucoup Marie*

1.2 Un formalisme idéal pour une interface syntaxe sémantique idéale ?

Dans un monde idéal il serait possible de travailler à partir de cette structure afin de construire une représentation logique dans l'optique d'une sémantique compositionnelle *à la Montague*. Pour établir les fondements d'un tel modèle il suffirait d'associer un λ-terme à chaque arbre élémentaire et d'utiliser l'arbre de dérivation comme support aux différentes β-réductions induites par les combinaisons syntaxiques de l'analyse. Ainsi, si l'on se base sur la propriété obligatoire de complétion d'un nœud de substitution, on peut considérer ce type de nœuds comme support à des variables argumentales. Comme l'adjonction a comme propriétés d'être non prédictible et optionnelle, on pourrait considérer l'arbre où l'adjonction a lieu comme un argument à la fonction associée à l'arbre auxiliaire qui s'adjoint. De cette façon, on pourrait interpréter l'arbre de dérivation figure 2 de la façon suivante (figure 3) sachant que dans ce mini modèle, on remplace les variables classiques du λ-calcul par des positions argumentales entre crochets[4].

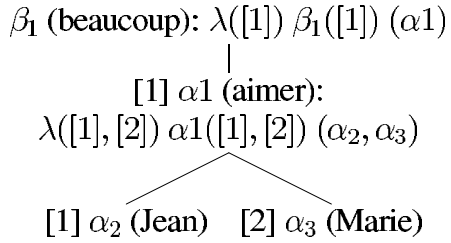


Figure 3: Interface syntaxe sémantique dans un monde idéal

Le problème de ce type d'interfaces syntaxe-sémantique est qu'elles fonctionnent uniquement sur un sous-ensemble restreint du langage avec des structures prédicatives simples. Les LTAG en tant que formalisme de départ ne font pas de distinctions entre adjonctions prédicatives et modifieures (les premières inversant le sens des dépendances sémantiques (Candito & Kahane, 1998)) ; de plus il n'y a pas de mécanismes permettant de résoudre les problèmes d'ambiguïtés des quantifieurs et, pire, certains liens syntaxiques entre des compléments verbaux et leurs sujets n'apparaissent pas dans l'arbre de dérivation.
Les solutions analysant les déficiences de l'arbre de dérivation sont nombreuses (Candito & Kahane, 1998; Schabes & Shieber, 1994; Rambow & Joshi, 1994). Ces solutions peuvent être divisées en deux groupes : celles où l'arbre de dérivation est considéré comme inutilisable, l'arbre dérivé est donc utilisé comme support à une interface syntaxe-sémantique (Gardent & Kallmeyer, 2003; Franck & van Genabith, 2001) ; et celles où l'on considère que les informations portées par l'arbre de dérivation doivent être enrichies, le formalisme LTAG étant modifié

[4]Les λ-termes lexicaux sont ici, bien sûr, simplifiés

sinon remplacé par les TAG Ensemblistes[5] afin de pouvoir gérer les informations de portées des modifieurs (Kallmeyer & Joshi, 1999).

Les solutions basées sur l'arbre dérivé ont pour principal problème d'être basées sur l'unification de structures de traits et sur l'utilisation conjointe d'une sémantique plate comme *liant* via unification des arguments afin d'obtenir une structure prédicative correcte ou une formule logique. Le problème est que chaque événement qui a lieu durant ce processus advient sur un nœud où une dérivation a pris place (adjonction ou substitution) ; ces solutions simulent donc de façon implicite l'arbre de dérivation au sein d'une structure que lui-même décrit sans équivoque[6].

Nous partons du principe que pour établir une interface syntaxe sémantique à partir des LTAG, nous devons d'abord nous assurer que tous les liens argumentaux sont présents dans la structure de représentation.
Sur la base de l'analyse de la problématique des verbes à contrôle, nous proposons une façon de construire un graphe de dépendance à partir des informations contenues tant dans l'arbre dérivé que dans l'arbre de dérivation à l'aide d'une structure décrivant ces deux arbres : la forêt partagée.

2 La problématique posée par les verbes à contrôle

Nous rappellerons brièvement dans cette section la difficulté d'analyse que posent les verbes à contrôle dans le formalisme LTAG[7].
On utilise souvent les verbes à contrôle comme témoin d'un hiatus entre syntaxe et sémantique en TAG pour la simple raison que s'il existe un lien de sous-catégorisation entre un sujet et le verbe qui le sous-catégorise, ce lien devrait être représenté dans l'arbre de dérivation (principe de co-occurence prédicat-argument).
Or l'arbre de dérivation (fig. 5) issu de l'analyse de la phrase *(1) Jean espère dormir* , suivant la grammaire jouet figure 4[8], ne contient pas de lien entre le sujet non réalisé de "dormir" et le sujet qu'il sous-catégorise : "Jean". Or ce lien est présent via la structure de traits dans l'arbre dérivé[9] figure 5.

En réalité la structure que nous voudrions obtenir est un graphe dans lequel ce lien est présent (figure 6).

Dans cette analyse, un verbe à contrôle ancre un arbre auxiliaire (*i.e arbre à contrôle*) qui s'adjoint sur la racine d'un arbre initial ayant au plus un nœud de substitution non-réalisé. On peut donc considérer qu'un verbe à contrôle *transfère* l'un de ses arguments vers l'arbre sur lequel il s'adjoint. L'objectif est donc de formaliser ce processus à travers une opération

[5](Weir, 1988) pour les TAG ensemblistes.

[6]Un autre problème de ce type de solutions basées sur l'arbre dérivé et les structures de traits est l'emploi d'un nombre non fini de traits ce qui a pour conséquence d'accroître la capacité générative du formalisme(Kallmeyer, 2004).

[7]On pourra se reporter à (Abeillé, 1999) pour une analyse complète mise en perspective avec les phénomènes des verbes à montées et comparée à d'autres formalismes.

[8]On notera la position vide dominée par le nœud N de l'arbre α_5, nous appelons ce nœud **"nœud de substitution non réalisée"** ou sujet non réalisé

[9]comme le trait d'accord pour la phrase *(2) Marie espère être belle.* A des fins de lisibilité, nous marquons cet accord par un indice de co-indiciation.

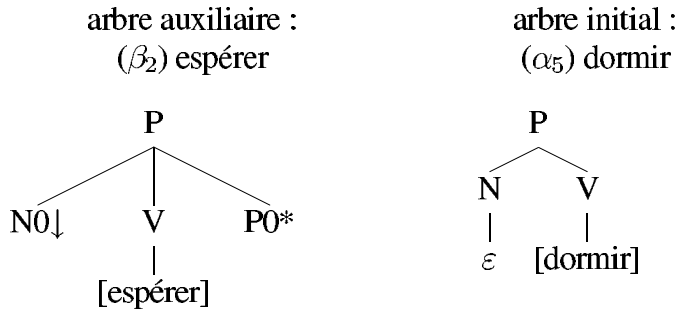


Figure 4: Grammaire jouet verbe à contrôle

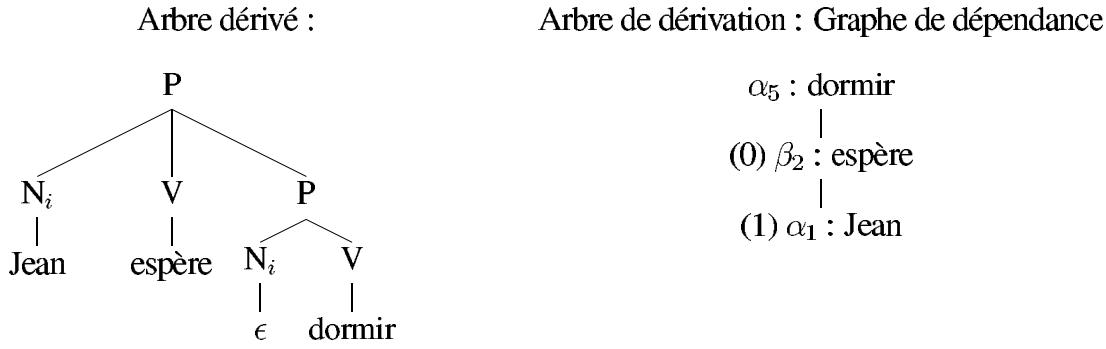


Figure 5: Analyse LTAG : "Jean espère dormir"

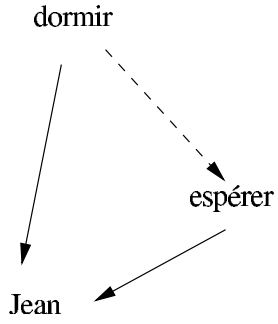


Figure 6: Graphe pour "Jean espère dormir"

appelée : fusion argumentale.

2.1 Informations préalables

Les seules opérations visibles sur un arbre de dérivation sont la substitution et l'adjonction parce qu'elles témoignent du passage d'un arbre à un autre. Si nous voulons faire apparaître ce lien manquant, nous devons donc simuler la dérivation qu'il induit forcément. De quelles informations disposons-nous ?

- Nous connaissons le nombre d'arguments dans un arbre (nœuds de substitution)
- Nous connaissons quel argument doit être transféré car il s'agit d'une information lexi-

cale, **le canevas de contrôle**, que nous marquons directement sur le nœud concerné. Le canevas de contrôle associé au nœud N d'un arbre ancré par un verbe à contrôle transférant son i-eme argument vers le j-eme argument est noté $\boxed{\mathrm{N}_{i\rightarrow j}}$.

Exemple :	*Jean interdit à Marie de dormir*	*Jean espère dormir*
canevas de contrôle	$N_1 \rightarrow N_0$	$N_1 \rightarrow N_0$

- Nous savons qu'un arbre dont un nœud domine une position vide, un argument non réalisé, va recevoir un argument provenant de l'adjonction d'un arbre à contrôle via l'opération de fusion. Ce transfert d'argument marquera la création effective du lien.

Dans ce cas, si nous savons à l'avance que ce lien va être créé, nous pouvons d'ores et déja prévoir la position de ce lien dans le graphe de dérivation. C'est pourquoi nous créons une nouvelle dérivation que nous appelons **Dérivation incomplète** (marquée par la variable libre $\boxed{?}$ dans la figure 7) et qui témoigne d'une substitution non réalisée sur le nœud N d'un arbre α.

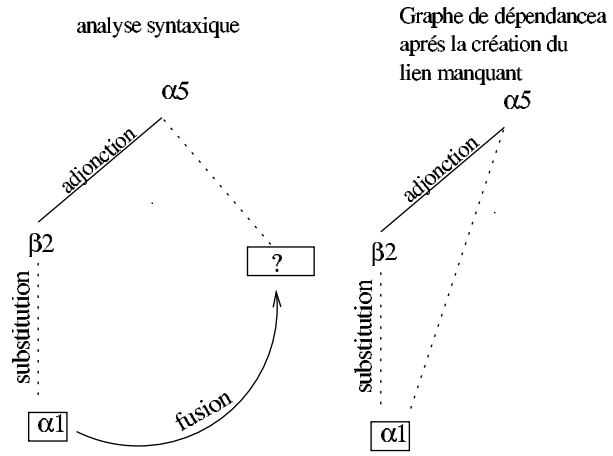


Figure 7: Vision synthétique de l'opération de fusion

Avant de définir formellement l'opération de fusion sous la forme de règles d'inférence, la section suivante va expliciter le processus d'extraction des items de dérivations.

3 Forêt partagées et items de dérivations

Cette section présente l'extraction d'une forêt de dérivation classique, c'est-à-dire sans prise en compte du processus de fusion. La forêt partagée sur laquelle nous basons notre processus résulte de l'intersection d'une grammaire TAG et d'un automate d'entrée, (Vijay-Shanker & Weir, 1993) et (Billot & Lang, 1989). La définition donnée par (Vijay-Shanker & Weir, 1993) est étendue par (Seddah, 2004) afin de lier la reconnaissance du sous arbre dominé par le nœud site d'une adjonction au nœud pied de l'arbre auxiliaire qui s'adjoint. Ceci afin de permettre un parcours synchrone de la forêt partagée. Nous la représentons sous la forme d'une grammaire hors-contexte augmentée d'une pile bornée contenant les adjonctions en cours. Chaque partie de règle correspond à un item de type Cocke-Kasami-Younger de la forme $< N, POS, I, J, Pile >$ où N est un nœud d'un arbre, Pos indique si l'on se situe après une adjonction prévisible ou

pas (marquée $\top$ si une adjonction sur le nœud est possible et $\bot$ si elle ne l'est plus, symbolisée par un point gras en position haute ou basse d'un nœud sur les schémas), I et J sont les indices de début et de fin de la chaîne reconnue par le nœud N et $Pile$ est la pile contenant les appels des sous arbres ayant démarré une adjonction et qui doivent être reconnus par la règle de reconnaissance du pied.
Le processus a lieu en deux temps : 1) La forêt partagée est générée à partir de la grammaire TAG initiale et d'une chaîne d'entrée 2) la forêt partagée est ensuite parcourue afin d'en extraire les dérivations.
L'extraction est simple : si une règle témoignant d'une dérivation est validée, un item de dérivation est inféré.

3.1 Forme des items de dérivation

A chaque occurrence d'une règle de dérivation, nous produisons un item de dérivation en témoignant. Cet item que nous appelons **Deriv** est de la forme $< N, \gamma, \alpha, type >$ avec N, le nœud qui reçoit la substitution ; γ, l'arbre de ce nœud, α l'arbre qui va s'y substituer, $type$ est le type de dérivation.
Trois dérivations sont possibles :

- une première évidente lors d'une substitution : on passe d'un nœud N d'un arbre γ à un nœud d'un arbre α :
  Item de substitution : $< N, \alpha, \gamma, subst >$

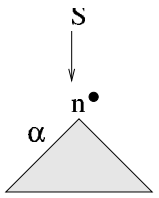
- lors d'une adjonction : on passe d'un nœud d'un arbre γ au nœud racine d'un arbre β. Nous ne considérons pas comme une dérivation le passage du nœud pied de l'arbre β au sous arbre du nœud site de l'adjonction de l'arbre α. Cette information est de fait évidemment redondante : une adjonction étant une insertion, il doit y avoir retour et analyse de l'arbre appelant.
  Item d'adjonction : $< N, \beta, \gamma, adj >$

- lors de la reconnaissance d'un arbre initial α ayant portée sur toute la chaîne, on crée une dérivation de type "tête".
  Item axiomes : $< N, \alpha, -, - >$

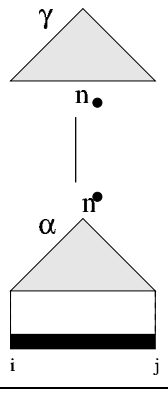
3.2 Règles d'inférence de l'algorithme d'extraction du graphe de dérivation

Pour toute règle $r \in R$ se situant sur le chemin succès, nous appliquons les règles d'inférence suivantes :

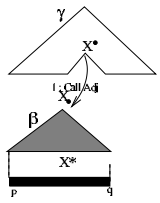
- **Dérivation de l'axiome**
  Si la règle r = S $\longrightarrow < \top, N_\alpha, 0, n, -, -, \emptyset >$ est validée lors de l'exécution de la grammaire, alors nous produisons l'item $< N, \alpha, -, - >$.

$S \longrightarrow < \top, N_\alpha, 0, n, -, -, \emptyset >$	
Reconnaissance d'un axiome $\frac{S \longrightarrow < \top, N_\alpha, 0, n, -, -, \emptyset >}{< N, \alpha, -, - >}$ *avec n=longueur de la chaine*	

- **Dérivation d'une substitution**
  Si la règle $r = < \bot, n_\gamma, i, j, -, -, Pile > \longrightarrow < \top, n_\alpha, i, j, -, -, Pile >$ est validée alors nous produisons l'item $< N, \alpha, \gamma, subst >$.

$< \bot, n_\gamma, i, j, -, -, Pile > \longrightarrow < \top, n_\alpha, i, j, -, -, Pile >$	
Reconnaissance d'une substitution $\frac{< \bot, n_\gamma, i, j, -, -, Pile > \longrightarrow < \top, n_\alpha, i, j, -, -, Pile >}{< N, \alpha, \gamma, subst >}$	

- **Dérivation d'une adjonction**
  Si la règle $r = < \top, n_\gamma, i, j, p, q, Pile > \longrightarrow < \top, n_\beta, i, j, p, q, Pile' >$ est validée alors nous produisons l'item $< N, \beta, \gamma, adj >$.

$< \top, n_\gamma, i, j, p, q, Pile > \longrightarrow < \top, n_\beta, i, j, p, q, Pile' >$	
Reconnaissance d'une adjonction $\frac{< \top, n_\gamma, i, j, p, q, Pile > \longrightarrow < \top, n_\beta, i, j, p, q, Pile' >}{< N, \beta, \gamma, adj >}$	

Les variables Pile et Pile′ représentent l'état de la pile d'appel des adjonctions lors de l'exécution d'une forêt produite. Leurs valeurs n'est pas prise en compte par cette règle d'inférence, car l'appel d'une adjonction est une opération hors contexte. Quelque soit le contenu de ces 2 variables, la dérivation sera celle témoignant de l'adjonction de l'arbre β sur le nœud X de l'arbre γ.

3.3 item de dérivation pour la dérivation incomplète

Nous avons décrit (section 2.1) un nœud dominant une position vide comme témoignant d'une substitution non réalisée et, par conséquent témoin d'une **dérivation incomplète**. La règle d'inférence suivante définit cette nouvelle dérivation :

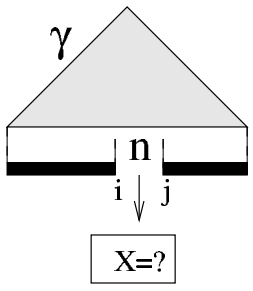
- Si la règle $< \bot, N_\gamma, i, j, -, -, Pile > \longrightarrow$ vrai est validée alors on produit la dériva-

tion suivante :

$< \perp, N_\gamma, i, j, -, -, Pile > \longrightarrow$ vrai

Dérivation d'une substitution non-réalisée

$$\frac{< \perp, N_\gamma, i, j, -, -, Pile >}{< N, \alpha, X, subst >} \textit{ avec } X, \textit{ non instancié}$$



L'ensemble des items de dérivation est généré dans un *chart* spécifique et correspond à l'ensemble des analyses possibles. Partant d'un item axiome, nous décrivons très exactement un arbre de dérivation. Si la grammaire suit les recommandations de (Rambow & Joshi, 1994), cet arbre peut être vu comme un arbre de dépendance, ainsi l'ensemble des arbres décrits par cette forêt d'items est dans ce cas une forêt de dépendance.

4 Formalisation du processus de fusion argumentale

L'analyse de l'adjonction d'un arbre à contrôle sur un arbre élémentaire met en jeu 3 dérivations : la dérivation $D1$ témoignant de la substitution d'un arbre α_1 sur le nœud N,dont le canevas de contrôle est $\boxed{N_{i\rightarrow j}}$, d'un arbre à contrôle β_2 ; la dérivation $D3$ témoignant de l'adjonction de l'arbre à contrôle β_2 sur la racine d'un arbre élémentaire γ[10] et la dérivation incomplète $D2$ témoignant de la présence d'un nœud N_j de substitution non réalisée sur l'arbre γ. L'opération de fusion est donc la règle d'inférence permettant de simuler la création du lien manquant via un nouvel item de dérivation $D4$ qui remplace l'item de dérivation incomplète $D3$.

$$\frac{D3 :< X, \beta_2, \gamma, adj > \; D1 :< N_{i\rightarrow j}, \alpha_1, \beta_2, subst > \; D2 :< N_j, \boxed{?}, \gamma, subst >}{D4 :< N_j, \alpha_1, \gamma, subst >}$$

5 Conclusion

Ce travail a été implémenté dans (Seddah, 2004). Sa caractéristique principale est de faire un usage intensif des propriétés des forêts partagées afin de parcourir simultanément les nœuds de l'arbre dérivé et de l'arbre de dérivation. Les nœuds qui sont utilisés lors de l'opération de fusion proviennent de l'arbre de dérivation pour les dérivations complètes et de l'arbre dérivé pour la dérivation incomplète. Si l'on considère que chaque item de dérivation correspond à un argument sémantique, le graphe de dérivation correspond à un graphe sémantique similaire au DSyntS de la théorie sens-texte (Mel'Cuk, 1997). Une extension du modèle pour traiter des phénomènes liés aux coordinations elliptiques est actuellement en cours d'élaboration. Pour bâtir ce modèle nous avons dû modifier légèrement le formalisme TAG pour inclure une information lexicale nécessaire aux règles d'inférence. La méthodologie consistant à travailler

[10] L'arbre élémentaire γ généralise le cas présenté figure 7 où la fusion opérait sur l'arbre initial α_5.

systématiquement au coeur de la forêt partagée nous permet de travailler sur toutes les analyses à la fois et donc de générer tous les graphes de dépendance au sein d'une forêt compacte.

References

ABEILLÉ A. (1991). *Une grammaire lexicalisée d'arbres adjoints pour le français.* PhD thesis, Paris 7.

ABEILLÉ A. (1999). Verbes "à monté" et auxiliaires dans une grammaire d'arbres adjoints. *LINX,Linguistique Institut Nanterre Paris X.*

BILLOT S. & LANG B. (1989). The structure of shared forests in ambigous parsing. In *33rd Conference of the Association for Computional Linguistics (ACL'89).*

CANDITO M.-H. & KAHANE S. (1998). Can the TAG derivation tree represent a semantic graph ? In *Proceedings TAG+4, Philadelphie*, p. 21–24.

FRANCK A. & VAN GENABITH J. (2001). Gluetag : Linear logic based semantics for LTAG -and what it teaches us about LFG and LTAG-. In *Proceddings of the LFG01 Conference, University of Hong Kong, Hong Kong.*

GARDENT C. & KALLMEYER L. (2003). Semantic construction in feature-based tag. In *Procedings of EACL 2003.*

KALLMEYER L. (2004). LTAG Semantics with Semantic Unification. In *Proceedings of TAG+7, To appears.*

KALLMEYER L. & JOSHI A. (1999). Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. In *Proceedings of the 12th Amsterdam Colloquium, December.*

MEL'CUK I. (1997). *Vers une linguistique Sens-Texte. Leçon inaugurale.* College de France, Paris.

RAMBOW O. & JOSHI A. K. (1994). *A Formal Look at Dependency Grammar and Phrase Structure Grammars, with Special consideration of Word Order Phenomena.* Leo Wanner, Pinter London, 94.

SCHABES Y. & SHIEBER S. (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics*, **20**(1), 91–124.

SEDDAH D. (2004). *Synchronisation des connaissances syntaxiques et sémantiques pour l'analyse d'énoncés en langage naturel à l'aide des grammaires d'arbres adjoints lexicalisées.* PhD thesis, Université Henry Poincaré, Nancy.

VIJAY-SHANKER K. & WEIR D. (1993). The use of shared forests in tree adjoining grammar parsing. In *EACL '93*, p. 384–393.

WEIR D. (1988). *Characterizing Midly Context-Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania.