Jouer avec des analyseurs syntaxiques

Éric Villemonte de la Clergerie INRIA - Rocquencourt - B.P. 105
78153 Le Chesnay Cedex, FRANCE
Eric.De_La_Clergerie@inria.fr

Résumé. Nous présentons DYALOG-SR, un analyseur syntaxique statistique par dépendances développé dans le cadre de la tâche SPRML 2013 portant sur un jeu de 9 langues très différentes. L'analyseur DYALOG-SR implémente un algorithme d'analyse par transition (à la MALT), étendu par utilisation de faisceaux et de techniques de programmation dynamique. Une des particularité de DYALOG-SR provient de sa capacité à prendre en entrée des treillis de mots, particularité utilisée lors de SPMRL13 pour traiter des treillis en Hébreu et reprise plus récemment sur des treillis produits par SXPIPE pour le français. Disposant par ailleurs avec FRMG d'un analyseur alternatif pour le français, nous avons expérimenté un couplage avec DYALOG-SR, nous permettant ainsi d'obtenir les meilleurs résultats obtenus à ce jour sur le French TreeBank.

Abstract. We present DYALOG-SR, a statistical dependency parser developed for the SPRML 2013 shared task over 9 very different languages. DYALOG-SR implements a shift-reduce parsing algorithm (a la MALT), extended with beams and dynamic programming techniques. One of the specificities of DYALOG-SR is its ability to handle word lattices as input, which was used for handling Hebrew lattices and more recently French ones produced by SXPIPE. Having access to FRMG, an alternative parser for French, we also tried a coupling with DYALOG-SR, providing us the best results so far on the French TreeBank

Mots-clés : Analyse syntaxique, Analyse syntaxique par dépendances, faisceaux, Programmation Dynamique, Treillis de mots, Couplage d'analyseurs.

Keywords: Parsing, Dependency Parsing, Beams, Dynamic Programming, Word Lattice, Parser coupling.

1 Introduction

Nous présentons diverses expériences d'analyse syntaxique pour le français menées avec l'analyseur statistique DYALOG-SR (Villemonte De La Clergerie, 2013a). Initialement développé pour participer à la campagne organisée en marge de SPMRL 2013 (Seddah *et al.*, 2013), cet analyseur en dépendances a été testé sur 9 langues très diverses, comme l'hébreu, le hongrois ou le coréen, et a terminé second dans sa catégorie. S'appuyant sur une stratégie d'analyse par transitions (à la MALT (Nivre, 2003)), DYALOG-SR utilise de plus la programmation dynamique pour gérer des choix non-déterministes au travers de faisceaux (*beams*). Mais la principale originalité de DYALOG-SR réside dans sa capacité à traiter en entrée des treillis de mots, pouvant représenter des ambiguïtés lexicales mais également des ambiguïtés de segmentation. Cette capacité a été utilisée dans le cadre de SPMRL sur des treillis en hébreu.

À ce stade, une première expérience préliminaire a consisté à entraîner et à évaluer DYALOG-SR sur la version en dépendances du French TreeBank (FTB) (Candito *et al.*, 2010a), de manière à pouvoir le comparer avec les autres systèmes évalués sur ce même treebank (Candito *et al.*, 2010b; Urieli & Tanguy, 2013; Le Roux *et al.*, 2012). Il nous a paru intéressant de tester également DYALOG-SR sur les treillis de mots produits par le segmenteur SXPIPE (Sagot & Boullier, 2008), en particulier pour vérifier son comportement sur les treillis pour une autre langue que l'hébreu. De fait, nous avons du affiner certains détails de l'algorithme, en particulier pour une meilleure prise en compte des mots en lecture avant (*lookahead*).

Par ailleurs, au travers de FRMG (de La Clergerie, 2005b), nous disposons d'un autre analyseur pour le français, basé sur une grammaire linguistique, de bonne qualité et de large couverture, capable de produire des analyses respectant le schéma d'annotation du FTB (Villemonte De La Clergerie, 2013b). Ceci nous a incité à tenter une expérience de couplage entre DYALOG-SR et FRMG, en utilisant les résultats de FRMG comme traits de guidage pour DYALOG-SR. Cette manipulation, finalement très simple à mettre en oeuvre, s'est révélée fructueuse et assure les meilleurs performances, à notre connaissance, obtenues à ce jour sur le FTB.

Dans la section 2, nous présentons les grandes lignes de l'algorithme d'analyse mis en oeuvre par DYALOG-SR, ainsi que, dans la section 3, les adaptations apportées pour le traitement de treillis de mots. Nous rappelons quelques résultats obtenus par DYALOG-SR pendant la campagne SPRML. Partant de ce point de départ, les expériences menées sur le français sont ensuite décrites en section 4, en précisant certaines modifications supplémentaires apportées à DYALOG-SR pour améliorer ses performances et son efficacité. Enfin, en section 5, nous présentons et discutons les résultats obtenus pour ces expériences sur le corpus FTB, ainsi que sur le corpus hétérogène SEQUOIA.

2 Un algorithme d'analyse par faisceau

DYALOG-SR met en oeuvre une stratégie d'analyse par transitions utilisant le système déductif *arc-standard* de la figure 1(a) sur des configurations formées d'une position j dans la chaîne d'entrée et d'une pile S dont les éléments sont des arbres (partiels) de dépendances. À l'étape m, soit une transition d'empilement (*shift*) empile le prochain mot de la chaîne sur le sommet de la pile, soit une transition de réduction établit une dépendance de label l entre le sommet r_0 de l'arbre s_o comme gouverneur et le sommet r_1 de l'arbre s_1 comme gouverné (réduction $re_{_l\curvearrowleft}$) ou le contraire (réduction $re_{\curvearrowright_l}$). Les configurations sont complétées par un score c mis à jour lors de l'application des transitions par consultation d'un modèle statistique donnant leur coût élémentaire (ξ, λ et ρ). La stratégie *arc-standard* ne permet que de construire des arbres de dépendances projectifs (sans croisement de dépendances) et est proche de la stratégie *arc-eager* [1] généralement mise en oeuvre dans MALT pour les cas projectifs.

entrée : $w_0 \dots w_{n-1}$

axiome $0{:}\langle 0, \epsilon\rangle{:}0$

$$\text{shift}\frac{m{:}\langle j, S\rangle{:}c}{m+1{:}\langle j+1, S|w_j\rangle{:}c+\xi}$$

$$re_{_l\curvearrowleft}\frac{m{:}\langle j, S|s_1|s_0\rangle{:}c}{m+1{:}\langle j, S|s_1\ {}_l\!\curvearrowleft\! s_0\rangle{:}c+\lambda}$$

$$re_{\curvearrowright_l}\frac{m{:}\langle j, S|s_1|s_0\rangle{:}c}{m+1{:}\langle j, S|s_1\ \curvearrowright_l\ s_0\rangle{:}c+\rho}$$

but $2n-1{:}\langle n, s_0\rangle{:}c$

(a) sur les configurations

$$\text{shift}\frac{I = m{:}\langle j, s_0, s_1\rangle{:}(c, \iota)}{\begin{array}{c}J = m+1{:}\langle j+1, w_j, s_0\rangle{:}(c+\xi, \xi)\\ \text{tail}(J) \mathrel{+}= I\\ \text{back}(J) \mathrel{+}= (\text{shift}, I, \text{nil}, c+\xi)\end{array}}$$

$$re_{_l\curvearrowleft}\frac{\begin{array}{c}I = m{:}\langle j, s_0, s_1\rangle{:}(c, \iota)\\ I_t = \_{:}\langle \_, \_, s_2\rangle{:}(c', \iota') \in \text{tail}(I)\end{array}}{\begin{array}{c}J = m+1{:}\langle j, s_1\ {}_l\!\curvearrowleft\! s_0, s_2\rangle{:}(c'+\delta, \iota'+\delta)\\ \delta = \iota + \lambda\\ \text{tail}(J) \cup\!= \text{tail}(I_t)\\ \text{back}(J) \mathrel{+}= ({}_l\!\curvearrowleft, I, I_t, c'+\delta)\end{array}}$$

(b) sur les items en programmation dynamique (fragment)

FIGURE 1 – Systèmes déductifs pour la stratégie *Arc-standard*
les _ dans les items dénotent des champs dont les valeurs ne nécessitent pas d'être consultées

Néanmoins, là où MALT explore l'espace de recherche de manière déterministe et gloutonne en prenant une suite de décisions locales, DYALOG-SR utilise une approche non-déterministe en maintenant, pour chaque étape m, un faisceau de possibilités de largeur k. Pour cela, suivant (Huang & Sagae, 2010), il met en oeuvre des principes de programmation dynamique en identifiant des ensembles de configurations se comportant de manière équivalente par rapport aux transitions et représentables sous forme d'*items*. Plusieurs choix sont possibles pour définir ces items, et dans notre cas, la composante principale d'un item I est fournie par $m{:}\langle j, s_0, s_1\rangle{:}(c, \iota)$ où j dénote la position courante dans la chaîne d'entrée, s_0 et s_1 les 2 premiers éléments de pile (plus précisément des approximations sur les noeuds racines et leurs dépendances immédiates), c le coût préfixe maximum menant à I, et ι le coût *intérieur* maximum depuis l'empilement d'un élément au dessus de s_1, De plus, suivant (Goldberg *et al.*, 2013), nous maintenons pour chaque item I un ensemble d'items queue (*tail*) I_t nécessaires lors des réductions pour retrouver le bas de la pile. Enfin, des pointeurs arrière (typés) sont associés à chaque item permettant de reconstruire une dérivation et un arbre de dépendance à partir d'un item final. Par ailleurs, le système déductif est adapté pour les items comme illustré par la figure 1(b).

L'utilisation de la programmation dynamique signifie en pratique que, même si seuls les k meilleurs items sont conservés à chaque étape m, nettement plus de k configurations sont accessibles au travers des pointeurs arrières. Par contre, à chaque étape, seule l'information présente dans un item est accessible, soit bien moins que dans une configuration complète.

La phase d'apprentissage s'appuie sur les transitions fournies par un oracle et utilise un perceptron moyennée pour mettre

1. Cependant, la stratégie *arc-eager* tend à créer les dépendances au plus tôt, au contraire de *arc-standard*.

à jour le modèle statistique (Daume, 2006), en suivant une stratégie agressive de mise à jour au plus tôt (*early strategy*, (Huang *et al.*, 2012)) dès que l'item oracle $\mathcal{O}_m$ obtenu à l'étape m par application des m premières transitions données par l'oracle se trouve en dehors du faisceau d'items.

Le choix des transitions à appliquer s'appuie sur un jeu assez classique de traits, comprenant en particulier des traits lexicaux liés aux champs du format tabulaire CONLL comme `lex`, `lemma`, `cat` (CPOS), `fullcat` (POS), `mstag` (traits morphosyntaxique FEATS). Ces traits sont déclinés pour le prochain mot la_1 dans la chaîne, pour les deux mots suivants de lecture avant la_2 et la_3 (*lookahead*), et (quand présents) pour les racines r_0 et r_1 des arbres s_0 et s_1 ainsi que pour leurs fils extremum à gauche et à droite. Nous avons aussi des traits liés aux dépendances comme les labels, les valences et domaines, ainsi que les distances discrétisées entre r_0, r_1 et la_1. Les traits ont en général des valeurs atomiques (symboliques ou numériques) mais acceptent également des listes de valeurs, par exemple pour décomposer le trait `mstag` en sous-traits (`gender`, `number`, ...). Ces traits peuvent être combinées sous forme de séquences (*motifs*) pour prendre en compte des corrélations multi-facteurs.

La table 1(a) rappelle quelques résultats obtenus par DYALOG-SR sur le français lors de la campagne SPMRL 2013. Quatre configurations étaient proposées aux participants, avec l'apprentissage réalisé sur un corpus de 14K phrases (`full`) ou un corpus de 5K phrases (`5k`), et avec en entrée un étiquetage de référence (`gold`) ou prédit (`pred`). Le schéma d'annotation utilisé était une variante plus riche du schéma FTB, avec en particulier une relation `dep_cpd` servant à relier les composants d'une forme composée (comme par exemple les composants du terme *motion de censure*). En mode `pred`, la détection de ces relations `dep_cpd` était une des principales difficultés de la tâche.

Dans la table 1(a), les résultats de DYALOG-SR sur la partie `test` sont comparés à ceux du meilleur système, à un système baseline de type MALT, et à la moyenne des résultats fournis par les participants. Les scores sont exprimés en *Labeled Attachement Score* (LAS), en prenant en compte la ponctuation. Enfin, la valeur pour b indique la largeur optimale du faisceau calculée sur la partie `dev`.

On observe que DYALOG-SR reste relativement loin du meilleur système pour toutes les configurations, mais se place néanmoins systématiquement dans la moyenne haute des participants, un résultat somme toute honorable pour un système développé et configuré en environ un mois. Sur l'ensemble des 9 langues, DYALOG-SR se classe second [2], juste devant une version de MALT optimisée sur son jeu de traits grâce à l'emploi de MALTOPTIMIZER (Ballesteros & Nivre, 2012). Enfin, la figure 4(a) montre l'impact des faisceaux dans l'amélioration significative des performances pour toutes les langues, à l'exception encore mystérieuse du coréen.

	DYALOG-SR		autres systèmes		
configuration	test	beam	meilleur	baseline	moyenne
gold/full	87,69	8	90,29	79,86	85,99
gold/5k	85,66	8	88,73	78,16	84,49
pred/full	82,06	8	85,86	77,98	81,03
pred/5k	80,11	4	83,60	76,54	79,31

(a) sur le français (métrique LAS)

	DYALOG-SR	
	chaîne	treillis
baseline beam6	87.34	76.35
modifié + `length` b6		83.47
modifié + `length` b16		86.75
modifié - `length` b16		86.50

(b) sur l'hébreux (métrique TED)

TABLE 1 – Résultats SPMRL 2013 pour le français et l'hébreu

3 Traiter des treillis de mots

Une des particularité de DYALOG-SR réside dans sa capacité à prendre en entrée des treillis de mots. Un tel treillis est un graphe dirigé acyclique (DAG) dont les arcs sont décorés par des mots et dont les noeuds dénotent des positions dans la phrase, comme illustré par la figure 2 pour une phrase en hébreu. Ces treillis sont classiquement utilisés pour représenter des ambiguïtés au niveau des mots mais aussi des ambiguïtés de segmentation et une sous-tâche de SPMRL était dédiée au traitement de treillis, en particulier pour l'hébreu.

Si le traitement de treillis de mots est relativement classique pour les approches d'analyse syntaxique par chartes [3], ce n'est pas le cas pour les approches statistiques et en particulier pour les approches par transitions. En effet, les chemins

2. L'ensemble des résultats sont disponibles dans (Seddah *et al.*, 2013; Villemonte De La Clergerie, 2013a)
3. et est formalisé en tant qu'intersection du langage engendré par une grammaire avec le langage régulier engendré par le treillis

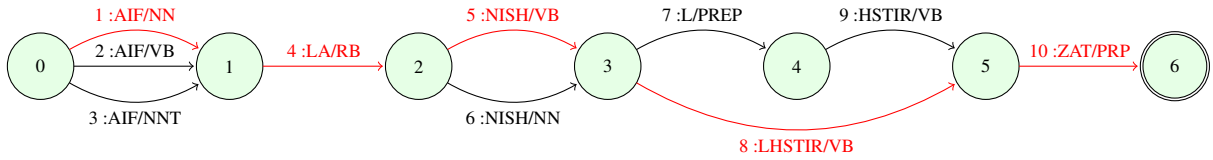


FIGURE 2 – Un treillis ambiguë (avec, en rouge, le chemin de segmentation référence *AIF LA NISH LHSTIR ZAT*)

possibles dans un treillis peuvent être de longueurs différentes, rendant difficile la comparaison des scores associés aux configurations (et aux items). Une première solution à ce problème consiste à normaliser les scores en fonction de la longueur des chemins, mais les expériences menées ne se sont pas révélées très concluantes, même si un gain de quelques points a été obtenu.

Finalement, un trait `length` a été ajouté à chaque mot, dénotant la longueur de l'arc sous-jacent dans le treillis, laissant au mécanisme d'apprentissage le soin de déterminer les meilleurs poids relatifs aux diverses longueurs (en conjonction avec les autres traits).

Une autre difficulté des treillis de mots est lié à l'utilisation des mots en lecture avant. Le choix de ces mots est maintenant non-déterministe et doit rester cohérent avec un chemin valide dans le treillis. En pratique, il devient nécessaire d'étendre les configurations pour y ajouter explicitement le chemin choisi de lecture avant, comme formalisé par le système déductif de la figure 3. Les trois composants $\mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3$ dénotent des identifiants d'arcs dans le treillis qui forment un début de chemin valide à partir de la position j. La transition `shift` nécessite de choisir un nouvel identifiant la_4 prolongeant le chemin, suite à la consommation de la_1. Les items et transitions sur les items ont similairement été révisés. Bien entendu, l'ajout des identifiants peut grandement multiplier le nombre possible d'items, en gros par le nombre de chemins possibles (de taille 3) dans le treillis, d'où l'importance d'utiliser des faisceaux plus larges.

$$re_{l\curvearrowleft}\frac{m{:}\langle j, S|s_1|s_0, \mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3\rangle{:}c}{m+1{:}\langle j, S|s_1\ {}_l\!\curvearrowleft s_0, \mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3\rangle{:}c+\lambda}$$

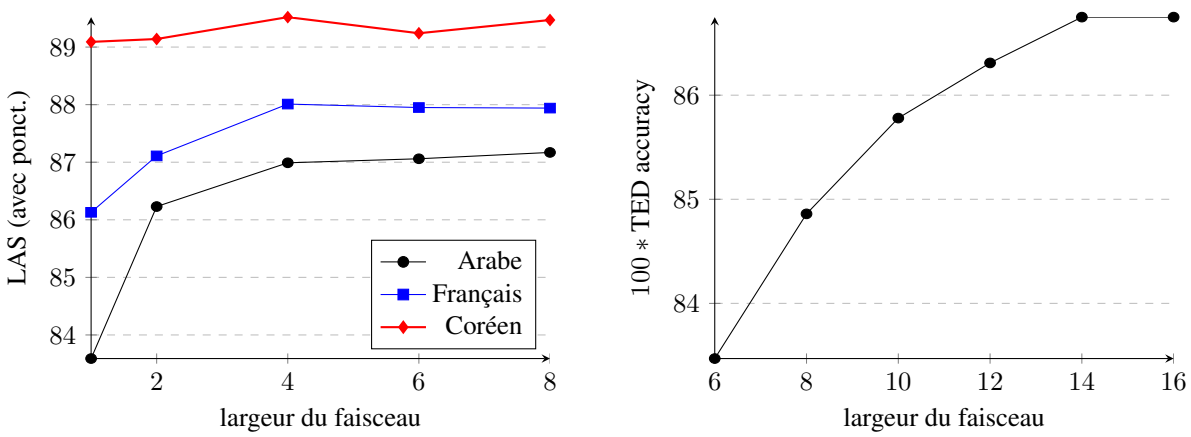
$$re_{\curvearrowright l}\frac{m{:}\langle j, S|s_1|s_0, \mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3\rangle{:}c}{m+1{:}\langle j, S|s_1\ \curvearrowright_l\ s_0, \mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3\rangle{:}c+\rho}$$

$$\mathrm{shift}\frac{m{:}\langle j, S, \mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3\rangle{:}c}{m+1{:}\langle k, S|\mathrm{la}_1, \mathrm{la}_2, \mathrm{la}_3, \mathrm{la}_4\rangle{:}c+\xi}$$

FIGURE 3 – Version révisée de la stratégie arc-standard, pour les treillis

Enfin, lors de la phase d'apprentissage, il est nécessaire de compléter l'oracle avec des informations sur le bon chemin à suivre dans le treillis de manière à pouvoir choisir l'arc suivant la_4 lors des transitions `shift`. En pratique, ce chemin est obtenu par alignement du treillis avec la segmentation de référence.

La table 1(b) donne les résultats obtenus pour l'hébreu, d'abord avec DYALOG-SR non modifié sur des chaînes (non ambiguës) et sur des treillis (ambiguës), puis avec DYALOG-SR modifié (et le trait `length`). Les scores utilisent la métrique TED (Tsarfaty *et al.*, 2011), adaptée à la mise en correspondance d'analyses ne s'appuyant pas sur une même segmentation. La version non modifiée est très mauvaise sur les treillis avec une chute de 11 points mais la version modifiée, pour un faisceau de taille 16, arrive quasiment à combler son retard, avec une perte de seulement 0,6 point. La figure 4(b) montre que des faisceaux plus larges semblent effectivement nécessaires pour compenser l'ambiguité des treillis (avec en moyenne 2,76 arcs par token), mais même avec un faisceau de taille 6, on observe des gains importants par rapport à la version non modifiée. Enfin, de manière surprenante, le trait `length`, bien qu'utile, ne semble pas essentiel et n'assure en définitive qu'un gain de 0,25, peut-être parce qu'il est en partie redondant avec les traits lexicaux.



(a) chaînes (b) treillis hébreux

FIGURE 4 – Influence des faisceaux

4 Trois expériences sur le français

La campagne SPMRL étant maintenant close, il est intéressant de prendre plus de temps et de recul pour explorer les capacités de DYALOG-SR, en particulier sur le français, pour lequel nous disposons d'outils et de ressources supplémentaires pouvant être utiles.

4.1 DYALOG-SR avec étiquetage préalable par MELT

La première expérience a surtout pour vocation de calibrer DYALOG-SR par rapport aux résultats déjà publiés sur le français, en particulier sur la version en dépendances du FTB (Candito *et al.*, 2010b; Urieli & Tanguy, 2013). Nous avons utilisé une version du FTB étiquetée et lemmatisée avec MELT [4] (Denis & Sagot, 2009) en mode validation croisée 10 fois avec une f-mesure de 97,88% sur la partie test du FTB, comparable au taux de 97,81% mentionné dans (Urieli & Tanguy, 2013). Les traits morpho-syntaxiques sont fournis par le lexique LEFFF (Sagot *et al.*, 2006), que nous utilisons également pour fournir des informations de sous-catégorisation pour les verbes. Nous incluons de plus des traits de *clustering* appris sur un large corpus par application de l'algorithme de Brown (Liang, 2005). Suivant (Urieli & Tanguy, 2013), nous avons également ajouté quelques traits liés à la présence (ou non) de ponctuations entre s_0 et le premier mot à lire la_1.

Enfin, nous avons réalisé que les règles de déduction sur les items (fig. 1(b)) permettent en fait d'accéder à l'élément de pile s_2 lors des réductions (au travers des items de queue I_t) et qu'il est donc possible d'utiliser des traits liés à s_2 lors des réductions. Par contre, pour les transitions `shift`, nous ne cherchons pas à retrouver s_2 et les traits associés.

Ces traits supplémentaires ont été également utilisés pour les deux autres expériences.

Tout dernièrement, nous avons également cherché à rendre encore plus agressive la stratégie de mise à jour du modèle lors de l'apprentissage. En particulier, en sus des règles existantes, le modèle est maintenant mis à jour dès que le coût de la dernière action conduisant à l'item oracle $\mathcal{O}_m$ se trouve en dessous d'une certaine marge avec le coût de la dernière action conduisant au meilleur item, ce même si $\mathcal{O}_m$ se trouve dans le faisceau. La valeur de la marge est automatiquement ajustée en fonction du coût moyen des transitions. Enfin, lors d'une mise à jour, le perceptron normalement incrémente ou décrémente les poids de 1 pour les traits concernés, mais cet incrément est maintenant porté à 2 en cas de détection de pertes de dépendances, c'est à dire quand le choix de la transition sur l'item à pénaliser conduit à perdre au moins une dépendance prédite par l'oracle.

De nouveaux tests réalisés avec la nouvelle version de DYALOG-SR sur les données SPRML du français (pred/full) ont montré une évolution des performances sur la partie `dev` de 82,88 à 83,51.

4. Ces données nous ont été gentiment préparées et fournies par Benoît Sagot.

4.2 DYALOG-SR sur des treillis produits par SXPIPE

Le fait que DYALOG-SR puisse prendre des treillis de mots en entrée nous a incité à le tester sur les treillis fournis par SXPIPE pour le français (Sagot & Boullier, 2008), avec des information lexicales fournies par LEFFF. En pratique, SXPIPE est configuré pour respecter le découpage en tokens fourni par le FTB mais peut reconnaître des lexèmes couvrant plusieurs tokens FTB (par exemple pour des entités nommées) ou, au contraire, avoir plusieurs lexèmes couvrant un même token FTB. Pour pouvoir à terme réaligner les treillis SXPIPE avec les données du FTB, nous leurs appliquons des transformations supplémentaires, comme illustré par la table 2.

token(s) FTB	**lexème(s) SXPIPE**	**après transformation**
10 avril	_DATE_artf/`nc`	10/`nc` : `cmpd` (part=1) avril/`nc` : `cmpd` (part=2)
en_surface	en/`prep` surface/`nc`	en_surface/`P+V`
avec_un_bel_ensemble	avec/`prep` un/`det` bel/`adj` ensemble/`nc`	avec_un_bel_ensemble/`aggl`

TABLE 2 – Exemples de transformations pour SXPIPE

Ainsi, un lexème SXPIPE multi-tokens de catégorie `X` est découpé en morceaux pour chaque token, avec des catégories de la forme `X` : `cmpd`. Un trait `part=i` est aussi ajouté aux traits morpho-syntaxiques pour chaque composant i. Dans l'autre sens, les lexèmes SXPIPE couvrant un même token FTB sont fusionnés. Ainsi la séquence `de/prep les/det` associée au token `des` est fusionnée en `des/prep+det`. Dans certains cas, ce processus de fusion peut amener à combiner un composant `X` : `cmpd` avec d'autres lexèmes.

Le processus de fusion crée un nombre très important de catégories complexes $X_1+\cdots+X_n$ (autour de 10 000 catégories), ce qui nous a amené à en rejeter certaines (hautement improbables) et surtout à remplacer toutes les séquences avec $n > 2$ par la catégorie générique `aggl`. Au final, nous obtenons 188 catégories, ce qui représente encore un nombre très conséquent de catégories. Par ailleurs, la plupart des catégories simples (correspondant à celles du LEFFF) ont été converties vers les catégories simples du French TreeBank (`N`, `V`, `A`, `ADV`, `PRO`, `C`, ...). Cette conversion facilite en particulier l'alignement avec les données FTB pour retrouver le chemin de référence dans les treillis lors de la phase d'apprentissage. Les lemmes et traits morpho-syntaxiques sont également exploités pour cet alignement. La table 3 fournit quelques statistiques sur les treillis obtenus pour la partie test du FTB, avec ainsi 2 arcs en moyenne par token FTB.

#tokens	**#arcs**	**arcs/token**	**#catégories**	**dont X+Y**	**#(arcs cmpd)**	**#(arcs X+Y)**	**#(arcs aggl)**
278 083	560 176	2,01	188	131	30 980 (5,5%)	13 174 (2,3%)	2 318 (0,4%)

TABLE 3 – Quelques statistiques sur les treillis SXPIPE pour FTB test

Nos premiers tests ont permis de mettre en évidence des faiblesses dans l'algorithme initial de DYALOG-SR qui n'avaient pas été perçues pour l'hébreu. En effet, pour un faisceau de taille 1, les scores se situaient très bas, entre 40% et 45%, sur les corpus de développement et de test. Une analyse des résultats a montré que le choix de la bonne étiquette syntaxique posait souvent problème. Nous avons alors réalisé que la transition `shift` (fig. 3) ne prenait pas en compte le prochain mot en avant la_4 (le plus à droite) pour évaluer son score. Comme la_4 est ensuite intégré dans le nouvel item et ne peut plus être remis en cause, cela conduisait à un mauvais étiquetage, et en cascade à de mauvaises dépendances. Nous avons donc modifié DYALOG-SR pour prendre en compte des traits relatifs à la_4 lors des transitions `shift` et avons également ajouté quelques informations relatives au token suivant la_4 comme l'ensemble des catégories possibles.

Les expériences menées sur les treillis pour l'hébreu ont montré l'importance de faisceaux dont la largeur est en relation avec le niveau d'ambiguité de ceux-ci. Plutôt que d'augmenter globalement cette largeur, nous avons testé un mécanisme d'adaptation locale de la largeur du faisceau en fonction du niveau local d'ambiguïté. Empiriquement, pour l'étape m, nous appliquons un coefficient multiplicateur donné par $b(i) = \max(1, \log(|\mathcal{P}_{i,3}|))$ pour les positions i entre $m/2 \pm 2$ avec $\mathcal{P}_{i,3}$ dénotant l'ensemble des chemins de longueur 3 partant de i. Cela signifie que même pour un faisceau globalement de largeur 1, nous pouvons localement avoir une largeur > 1. En pratique, sur le FTB, $b(i)$ se situe entre 1 et 5.

Cependant, le coût en temps sur les treillis reste relativement élevé, en particulier à cause du mombre d'items créés et finalement rejetés car étant hors du faisceau. L'article (Choi & McCallum, 2013) nous a fourni une piste pour contrer ce phénomène : il suggère en effet que de bons résultats peuvent étre obtenus en ne considérant que les 2 meilleures configurations par étape, en se limitant de plus aux étapes à faible confiance, c'est à dire les étapes où les coûts des meilleures actions se situent dans une faible marge. Cela nous a conduit, pour DYALOG-SR, à ne considérer que les k meilleurs items dérivables par réduction à partir d'un item I, à condition que leurs coûts soient dans une certaine marge

m. Par défaut, nous prenons $k = 3$ (au lieu de 2 dans le cas de (Choi & McCallum, 2013)) et $m = 200$ (pour des coûts élémentaires de l'ordre de ± 1000).

4.3 DYALOG-SR guidé par FRMG

FRMG est une grammaire d'arbres adjoints (TAG) à large couverture du français, dérivée d'une description grammaticale de haut-niveau sous forme d'une méta-grammaire (de La Clergerie, 2005b). [5] Un analyseur par charte, compilé à partir de la grammaire, permet de retourner l'ensemble des analyses complètes possibles pour une phrase, sous forme d'une forêt partagée de dérivations. En cas d'échec de l'analyse, l'analyseur bascule en mode *robuste* et fournit des séquences d'analyses partielles, de manière à couvrir la phrase. La forêt de dérivation est convertie en forêt de dépendances, et est ensuite désambiguisée à l'aide d'un ensemble de règles heuristiques portant sur les dépendances. Enfin, l'arbre de dépendances ainsi obtenu peut être converti suivant divers schémas d'annotation syntaxique, dont celui utilisé pour les campagnes EASy et Passage ainsi que celui utilisé pour la version dépendance du FTB (schéma FTB). Récemment, des techniques d'apprentissage partiellement supervisé ont permis de fortement améliorer la phase de désambiguisation de FRMG (Villemonte De La Clergerie, 2013b), pour amener ses performances au niveau de celles d'analyseurs statistiques.

Disposant de deux analyseurs syntaxiques de bonne qualité pour le français, il était intéressant de les combiner. Dans le cas présent, nous avons choisi d'utiliser les résultats de FRMG pour guider les décisions de DYALOG-SR. Plus précisément, nous avons ajouté des traits statiques `frmg_label` et `frmg_delta`, indiquant respectivement le label de la dépendance entrante sur chaque mot, et la distance (positive ou négative) à son gouverneur. Les résultats obtenus avec ces traits préliminaires étant déjà extrêmement encourageants, nous avons alors recherché des traits plus proches d'un vrai guidage, liés aux configurations et pas seulement aux tokens.

Ainsi, lorsqu'un item I avec les 2 éléments de piles s_0 et s_1 de racines r_0 et r_1 est compatible avec une réduction donnant une dépendance de label l prédite par FRMG entre r_0 et r_1, un trait dynamique de guidage `reduce_right` ou `reduce_left` est ajouté, lors de l'exécution. Comme les réductions droites peuvent être en compétition avec une transition `shift`, un trait de guidage `shift` est également émis si le premier mot la_1 précède ou égale le descendant de r_0 le plus à droite pour FRMG. Enfin, le label l de la dépendance est ajouté au trait de guidage.

5 Résultats et discussions

Les trois expériences précédemment décrites ont été conduites sur la version en dépendances du FTB (Candito *et al.*, 2010a), en utilisant classiquement la partie `train` (9881 phrases) pour l'entraînement, la partie `dev` (1235 phrases) pour les mises au point et la partie `test` (1235 phrases) pour l'évaluation.

La table 5(a) fournit les performances en LAS (sans prise en compte de la ponctuation) pour les diverses versions de DYALOG-SR, pour FRMG, et pour divers analyseurs statistiques. Pour FRMG, nous fournissons les performances pour la version de base et pour une version `+tuning` après amélioration de la phase de désambiguisation par apprentissage sur la partie `train` du FTB. C'est cette version `+tuning` qui est utilisée pour l'expérience de couplage. Pour les analyseurs statistiques, nous reprenons comme baseline les résultats publiés dans (Candito *et al.*, 2010b) pour les systèmes BERKELEY (constituance + conversion en dépendances), MALT et MST (*Maximum Spanning Tree*).

Nous avons ensuite les résultats plus récents du système TALISMANE (Urieli & Tanguy, 2013), qui partage un certain nombre de points communs avec DYALOG-SR, comme l'emploi d'un algorithme par transition, l'emploi de faisceaux et l'emploi de LEFFF. TALISMANE intègre aussi, en plus des traits statistiques, des règles/contraintes linguistiques censées bloquer certaines configurations impossibles, ce qui a, en partie, motivée le couplage de DYALOG-SR avec une source d'information linguistique comme FRMG.

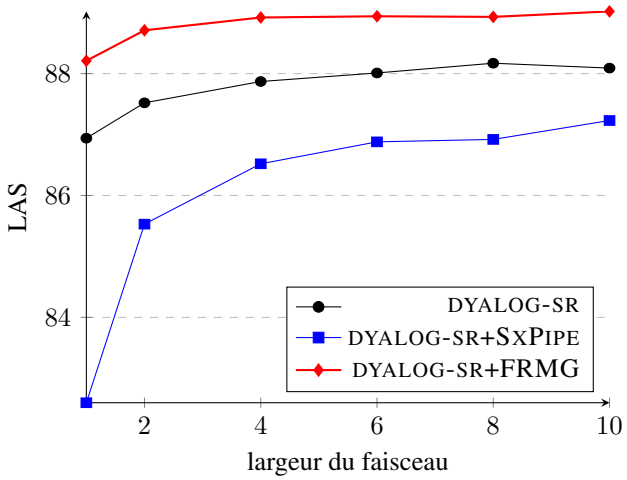
Enfin, nous mentionnons les résultats publiés pour le français dans (Le Roux *et al.*, 2012) qui s'appuient sur l'application d'un modèle discriminatif pour le ré-ordonnancement des n meilleurs analyses produites par une grammaire PCFG (avec un modèle génératif) et transformées en dépendances. La méthode donne de très bons résultats et présente quelques similarités avec l'idée de coupler plusieurs modèles statistiques. Pour information, nous donnons aussi les résultats obtenus par MATE (Bohnet, 2010), une version améliorée de MST exploitant des informations fournies par MELT.

Avec un étiquetage préalable par MELT, DYALOG-SR obtient de bons résultats, similaire à ceux de MALT pour un faisceau

5. Accessible en ligne sur `http://alpage.inria.fr/frmgwiki`.

système	dev	test
FRMG init	80,85	82,08
FRMG +tuning	86,20	87,49
BKY	86,50	86,80
MALT	86,90	87,30
MST	87,50	88,20
TALISMANE (b=1)	86,80	87,20
TALISMANE (b=20)	88,10	88,50
Le Roux (2012)	–	89,20
MATE+MELT	–	89,20
DYALOG-SR (b=1 i=5)	86,94	87,71
DYALOG-SR (b=8 i=10)	88,17	89,01
DYALOG-SR treillis (b=1 i=8)	82,60	83,68
DYALOG-SR treillis (b=10 i=9)	87,23	87,90
DYALOG-SR treillis (b=12 i=7)	87,15	88,15
DYALOG-SR+FRMG (b=1 i=6)	88,21	89,38
DYALOG-SR+FRMG (b=10 i=7)	89,02	90,25

(a) comparaison des systèmes



(b) impact des faisceaux (sur FTB `dev`)

FIGURE 5 – Performances sur le FTB (métrique LAS, sans ponctuations)

de taille 1. Les meilleurs résultats, obtenus avec un faisceau de taille 8 (et itération 10), dépassent ceux de la 1ère génération d'analyseurs (BKY, MALT et MST) et même ceux plus récemment obtenus par TALISMANE. Ils s'approchent des scores atteints par (Le Roux *et al.*, 2012). Il est intéressant de noter que des choix locaux en partie compensés par un faisceau assez large rivalisent avec des choix plus globaux, comme mis en oeuvre par MST ou MATE ou par réordonnancement dans (Le Roux *et al.*, 2012).

Les résultats obtenus par DYALOG-SR sur les treillis SXPIPE, sans étiquetage préalable et avec réalignement, sont finalement assez proches de ceux obtenus avec étiquetage, avec une perte d'un peu plus d'un point, malgré un jeu très important de 188 catégories syntaxiques. Ils dépendent cependant fortement de l'utilisation des faisceaux. La figure 5(b) montre en effet que les performances croissent de manière importante avec la largeur des faisceaux, avec plus de quatre points de gains, mais que grâce à l'adaptation locale des faisceaux, il n'est pas cependant nécessaire d'utiliser des faisceaux aussi larges que pour l'hébreu. Il est probable qu'il existe encore une marge de progression, en intégrant de nouveaux traits plus spécifiquement dédiés au choix du prochain lexème dans le treillis et en gérant mieux les processus de fusion des lexèmes SXPIPE. Comme les catégories syntaxiques ne sont pas celles du FTB, il est difficile de fournir une évaluation précise de l'étiquetage finalement obtenu. Néanmoins, en projetant sur les 15 catégories simples du FTB, on obtient une f-mesure de 95,35% sur FTB test (et $b = 10, i = 9$) alors qu'elle était autour de 70% avant les modifications apportées à la gestion de la_4 lors des transitions `shift`.

Enfin, les résultats les plus intéressants sont ceux fournis par le couplage de DYALOG-SR avec FRMG. On observe que les scores obtenus sont bien meilleurs que ceux obtenus par FRMG et DYALOG-SR pris isolément, et qu'à notre connaissance, ils sont les meilleurs à ce jour sur le FTB. Les deux analyseurs semblent donc complémentaires, ce qui conforte l'intérêt d'injecter des connaissances linguistiques dans un analyseur statistique, ici au travers des traits de guidage.

Pour confirmer cette complémentarité, nous avons examiné plus précisément le comportement de FRMG, DYALOG-SR, et DYALOG-SR+FRMG pour certaines relations de dépendance, comme illustré par la figure 6. On observe ainsi que DYALOG-SR+FRMG dépasse presque systématiquement les 2 autres analyseurs, en rappel et en précision. On note que certaines dépendances difficiles pour les analyseurs statistiques, comme `coord` et `mod_rel`, bénéficient pleinement des informations fournies par un analyseur linguistique comme FRMG, et que DYALOG-SR semble même capable de généraliser les informations de FRMG pour le dépasser. Dans l'autre sens, la faiblesse de FRMG sur la relation `mod` et, en rappel, sur les relations pour les arguments verbaux propositionnels (`a_obj`, `de_obj`, `p_obj`) ne perturbe pas le couplage. Enfin, le mode robuste, utilisé par FRMG quand il ne trouve pas d'analyse complète, introduit de fausses racines, ce qui fait chuter sa précision sur le label `root`. Mais le couplage résout totalement ce problème.

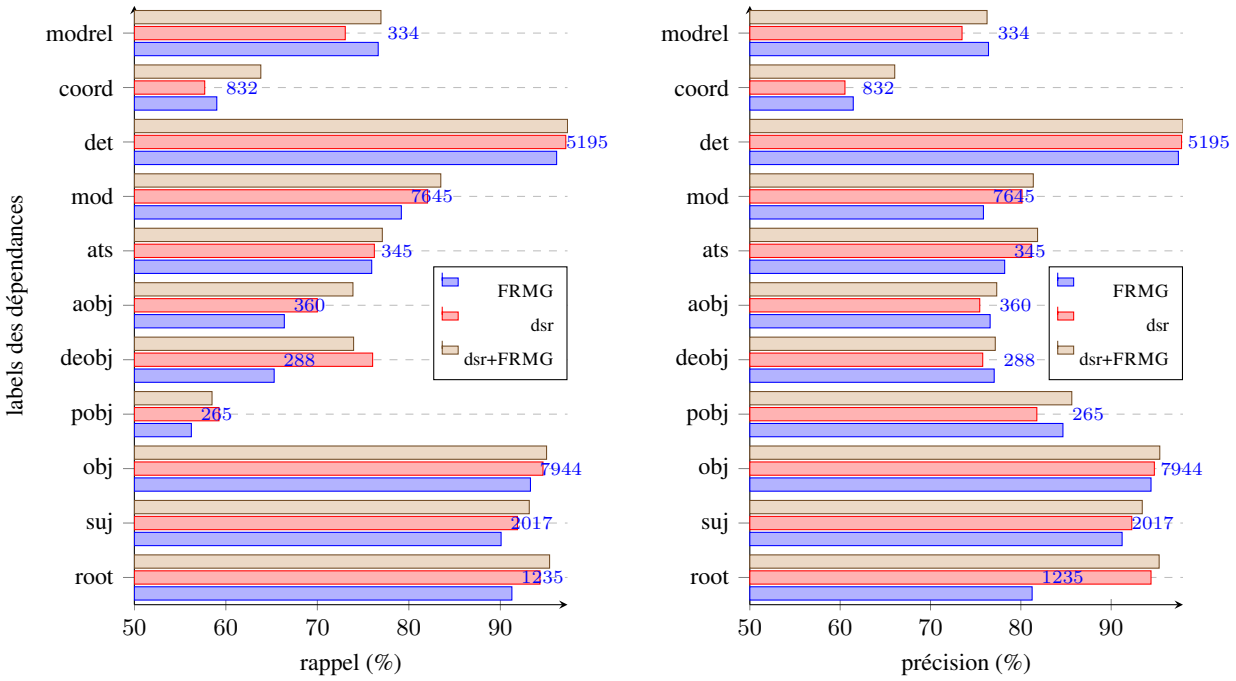


FIGURE 6 – Comparaison des systèmes sur les labels des dépendances
le volume de dépendances concernées est fourni en regard des barres

Toujours pour étudier les différences entre les analyseurs, nous les avons comparé selon divers critères *topologiques*, comme la distance entre gouverneur et gouverné (fig. 7(a)), la profondeur dans l'arbre de dépendance (fig. 7(b)), la taille de la fratrie (fig. 7(c)), et enfin le rang (gauche ou droit) dans la fratrie (fig. 7(d)). Nous notons ainsi que pour les dépendances à longue distance, difficiles pour un analyseur local comme DYALOG-SR, nous obtenons de bien meilleurs résultats par couplage avec FRMG. Pour la profondeur, qui pose problème à FRMG [6], nous notons une nette amélioration de DYALOG-SR+FRMG, en rappel et en précision. Pour le rang, et en particulier pour les rangs négatifs (dépendances gauches) qui posent problème pour DYALOG-SR, on voit que le couplage hérite des meilleures performances de FRMG. Enfin, la trace des faiblesses du mode robuste pour FRMG est bien visible en précision pour la distance nulle, la profondeur nulle et le rang nul, mais ces faiblesses sont totalement compensées pour DYALOG-SR+FRMG.

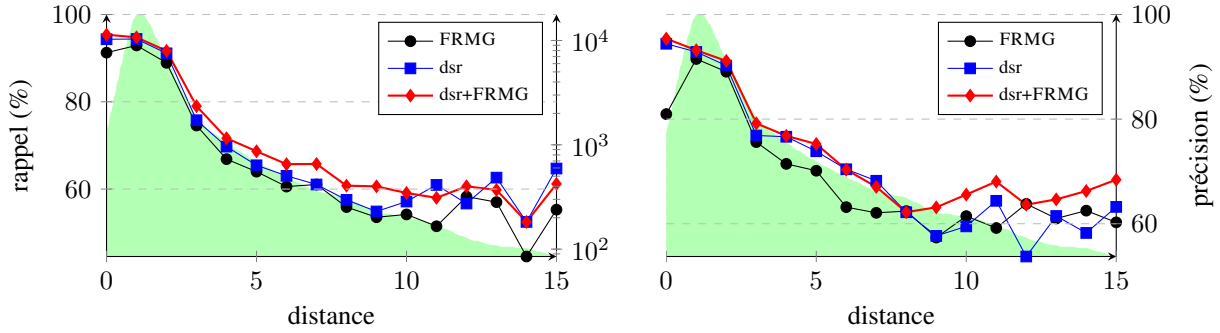
Enfin, nous avons examiné la stabilité des analyseurs sur le corpus SEQUOIA (Candito & Seddah, 2012). Les 3 204 phrases de ce corpus sont aussi annotées suivant le schéma du FTB mais couvrent divers domaines autres que le domaine journalistique du FTB. En particulier, les sous-parties EMEA correspondent à des textes médicaux difficiles à traiter. Pour des analyseurs entraînés sur FTB train, la table 4 montre que les pertes, par rapport à FTB test, sont plus faibles pour FRMG que pour DYALOG-SR, avec même FRMG dépassant DYALOG-SR en performance sur certains sous-corpus (Europar, annodis, et emea-fr-dev). [7] On observe par contre que le couplage préserve partiellement la stabilité fournie par FRMG et permet d'obtenir de bons scores sur SEQUOIA.

Conclusion

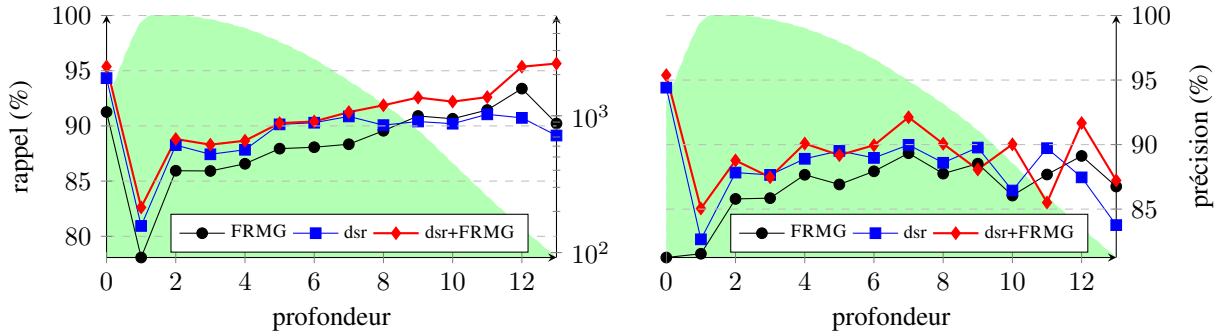
Nous avons présenté trois expériences menées sur le français et s'appuyant sur l'analyseur DYALOG-SR. Celles-ci montrent qu'un analyseur relativement simple par transition est néanmoins susceptible d'obtenir de bons résultats. Sa simplicité permet de facilement et rapidement tester diverses hypothèses et extensions, comme l'adaptation locale des faisceaux dans les treillis, et les expériences menées nous ont déjà permis de faire largement évoluer DYALOG-SR depuis ses débuts dans la campagne SPMRL 2013. Néanmoins, même si les faisceaux ont un impact important sur les performances et permettent

6. à priori parce que le modèle de FRMG n'utilise pas de traits directement ou indirectement liés à la profondeur, au contraire de DYALOG-SR avec les traits liés à s_0, s_1 et s_2.

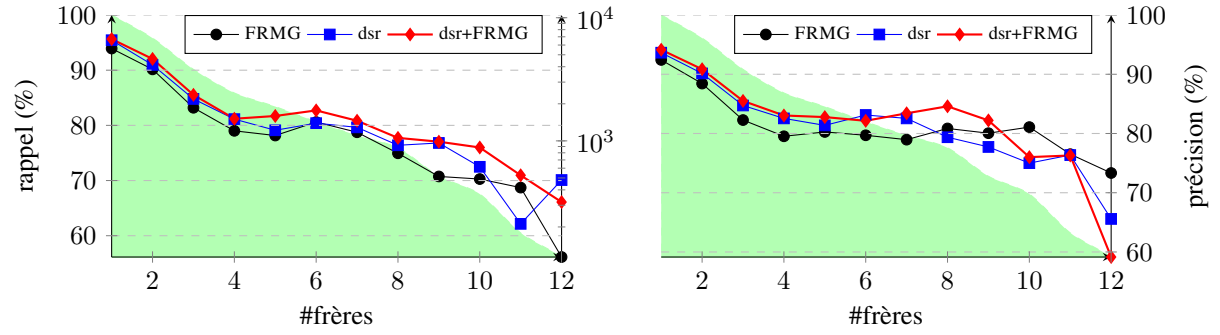
7. Les mauvaises performances de FRMG sur Wikipedia semblent dues à quelques longues phrases provoquant un timeout et une perte complète d'analyses.



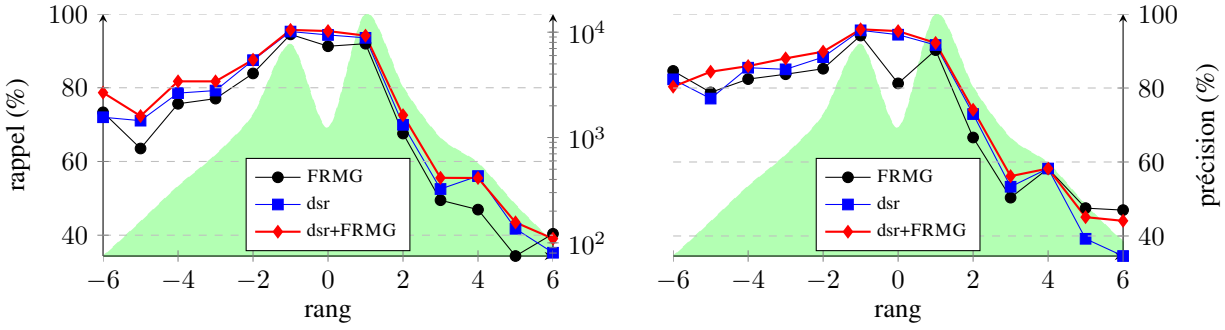
(a) par rapprort à la distance



(b) par rapport à la profondeur



(c) par rapport au nombre de frères



(d) par rapport au rang dans la fratrie

FIGURE 7 – Comparaison des systèmes selon divers critères *topologiques*
les fonds (en vert) indiquent le volume de dépendances concernées (échelle logarithmique à droite)

		FRMG			DYALOG-SR			DYALOG-SR+FRMG		
corpus	#phrases	LAS	Δerr	%Δerr	LAS	Δerr	%Δerr	LAS	Δerr	%Δerr
FTB test	1235	87,49	–	–	89,00	–	–	90,25	–	–
Europar	561	87,97	−0,5	−3,8	87,00	+2,0	+18,2	88,94	+1,3	+13,4
annodis	529	86,11	+1,4	+11,0	85,80	+3,2	+29,1	88,21	+2,0	+20,9
emea-fr-dev	574	85,16	+2,3	+18,6	83,50	+5,5	+50,0	86,26	+4,0	+40,9
emea-fr-test	544	84,67	+2,8	+22,5	85,01	+4,0	+36,3	86,87	+3,4	+34,7
frwiki	996	83,53	+4,0	+31,7	84,39	+4,6	+41,9	86,23	+4,0	+41,2

TABLE 4 – Stabilité des systèmes sur le corpus hétérogène SEQUOIA, après apprentissage sur FTB

en partie de corriger les choix locaux faits un analyseur par transition (comme MALT, TALISMANE ou DYALOG-SR), il paraît difficile à terme de concurrencer des approches plus globales, éventuellement obtenues par couplage de modèles.

En allant dans cette direction, l'expérience de couplage de DYALOG-SR avec FRMG nous a ainsi fourni d'excellents résultats. Elle confirme l'intérêt d'injecter une information plus linguistique dans un analyseur statistique comme DYALOG-SR (et comme tenté aussi dans TALISMANE). Elle confirme également l'intérêt d'exploiter des contraintes de *localité étendue*, telles que fournis (indirectement) par les arbres TAG de FRMG pour mieux guider les choix plus locaux de DYALOG-SR. Il est aussi intéressant de noter que ce couplage ne permet pas seulement d'hériter des meilleures propriétés de chaque système, mais permet en fait de les dépasser.

Le couplage de plusieurs analyseurs statistiques n'est pas nouveau (Sagae & Lavie, 2006), mais celui d'un analyseur statistique avec un analyseur fondé sur une grammaire linguistique est déjà plus original et prometteur (Ovrelid *et al.*, 2009). (Villemonte De La Clergerie, 2013b) a montré qu'il est possible d'améliorer les performances d'un analyseur symbolique en lui permettant d'apprendre à partir d'un corpus annoté comme le FTB, et nous voyons ici que la combinaison d'un analyseur statistique avec un analyseur symbolique est aussi une piste intéressante pour fortement améliorer les performances, ce qui est une incitation à utiliser au mieux les divers analyseurs du français qui existent (et pas forcément les seuls analyseurs statistiques entraînés sur le FTB). Cette approche, qui reste néanmoins assez lourde à mettre en oeuvre, assure aussi une meilleure stabilité pour des corpus hors domaine d'entraînement, comme montré sur le corpus SEQUOIA.

Ce couplage ouvre également la voie à d'autres expériences. Ainsi, l'examen fin des améliorations apportées par le couplage peut sûrement permettre d'identifier et de corriger certaines faiblesses de FRMG, en particulier pour le mode robuste. D'autre part, dans le cas présent, nous avons utilisé les informations de FRMG pour guider DYALOG-SR, mais il est aussi envisageable d'exploiter les résultats de DYALOG-SR pour guider FRMG, d'autant plus que DYALOG-SR peut travailler sur les mêmes treillis SXPIPE servant d'entrée à FRMG. Par ailleurs, les sorties de FRMG sur de gros corpus peuvent être exploitées pour entraîner DYALOG-SR. Enfin l'expérience de guidage que nous avons menée après étiquetage préalable du FTB pourrait aussi être tentée directement sur les treillis SXPIPE, modulo quelques efforts d'alignement pour attacher les informations sur les bons arcs des treillis.

Enfin, DYALOG-SR continue d'évoluer. Dans le cadre de sa participation à la campagne SEMEVAL 2014 (tâche 8), il a été modifié pour produire des graphes de dépendances sémantiques (plutôt que des arbres de dépendances syntaxiques), en s'appuyant sur un jeu plus riche de transitions, permettant de définir une stratégie de type arc-eager (attachement au plus tôt) ainsi qu'une forme faible de non-projectivité. D'autres développements sont prévus pour un traitement plus complet des cas de non-projectivité.

Références

BALLESTEROS M. & NIVRE J. (2012). MaltOptimizer : an optimization tool for MaltParser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 58–62.

BOHNET B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*.

CANDITO M., CRABBÉ B. & DENIS P. (2010a). Statistical French dependency parsing : treebank conversion and first results. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10)*, La Valette, Malte.

CANDITO M., NIVRE J., DENIS P. & HENESTROZA ANGUIANO E. (2010b). Benchmarking of statistical dependency parsers for French. In *Proceedings of COLING'2010 (poster session)*, Beijing, China.

CANDITO M. & SEDDAH D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *TALN 2012-19e conférence sur le Traitement Automatique des Langues Naturelles*.

CHOI J. D. & MCCALLUM A. (2013). Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria*.

DAUME H. C. (2006). *Practical structured learning techniques for natural language processing*. PhD thesis, University of Southern California.

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

DE LA CLERGERIE É. (2005b). From metagrammars to factorized TAG/TIG parsers. In *Proceedings of IWPT'05*, p. 190–191, Vancouver, Canada.

DENIS P. & SAGOT B. (2009). Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings PACLIC 23*, Hong Kong, China.

GOLDBERG Y., ZHAO K. & HUANG L. (2013). Efficient implementation of beam-search incremental parsers. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sophia, Bulgaria.

HUANG L., FAYONG S. & GUO Y. (2012). Structured perceptron with inexact search. In *Proceedings of HLT-NAACL 2012*, p. 142–151.

HUANG L. & SAGAE K. (2010). Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 1077–1086 : Association for Computational Linguistics.

LE ROUX J., FAVRE B., NASR A. & MIRROSHANDEL S. (2012). Generative constituent parsing and discriminative dependency reranking : Experiments on english and french. In *SP-SEM-MRL*.

LIANG P. (2005). Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

NIVRE J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, p. 149–160, Nancy, France.

OVRELID L., KUHN J. & SPREYER K. (2009). Improving data-driven dependency parsing using large-scale LFG grammars. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, p. 37–40 : Association for Computational Linguistics.

SAGAE K. & LAVIE A. (2006). Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume : Short Papers*, p. 129–132.

SAGOT B. & BOULLIER P. (2008). SXPIPE 2 : architecture pour le traitement présyntaxique de corpus bruts. *Traitement Automatique des Langues (T.A.L.)*, **49**(2), 155–188.

SAGOT B., CLÉMENT L., DE LA CLERGERIE É. & BOULLIER P. (2006). The Le*fff* 2 syntactic lexicon for French : architecture, acquisition, use. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC'06)*, Genova, Italie.

SEDDAH D., TSARFATY R., KÜBLER S., CANDITO M., CHOI J., FARKAS R., FOSTER J., GOENAGA I., GOJENOLA K., GOLDBERG Y., GREEN S., HABASH N., KUHLMANN M., MAIER W., NIVRE J., PRZEPIORKOWSKI A., ROTH R., SEEKER W., VERSLEY Y., VINCZE V., WOLIŃSKI M., WRÓBLEWSKA A. & VILLEMONTE DE LA CLERGERIE E. (2013). Overview of the SPMRL 2013 shared task : A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages : Shared Task*, Seattle, WA.

TSARFATY R., NIVRE J. & ANDERSSON E. (2011). Evaluating dependency parsing : Robust and heuristics-free cross-annotation evaluation. In *Proceedings of the 8th International Conference on Empirical Methods in Natural Language Processing (EMNLP 11)*, Edinburgh, UK.

URIELI A. & TANGUY L. (2013). L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur Talismane. In *Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013)*, p. 188–201, Les Sables d'Olonne, France.

VILLEMONTE DE LA CLERGERIE É. (2013a). Exploring beam-based shift-reduce dependency parsing with DyALog : Results from the SPMRL 2013 shared task. In *4th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL'2013)*, Seattle, États-Unis.

VILLEMONTE DE LA CLERGERIE É. (2013b). Improving a symbolic parser through partially supervised learning. In *The 13th International Conference on Parsing Technologies (IWPT)*, Nara, Japon.