Analyse syntaxique du français : des constituants aux dépendances

Marie Candito, Benoît Crabbé, Pascal Denis, François Guérin

Analyse syntaxique du français : des constituants aux dépendances

Marie Candito[1] Benoît Crabbé[1] Pascal Denis[2] François Guérin[2]
(1) Université Paris 7/INRIA (Alpage),
30 rue du Château des Rentiers, 75013 Paris
(2) INRIA (Alpage),
Domaine de Voluceau Rocquencourt - B.P. 105 78153 Le Chesnay
mcandito/bcrabbe@linguist.jussieu.fr, pascal.denis/francois.guerin@inria.fr

Résumé. Cet article présente une technique d'analyse syntaxique statistique à la fois en constituants et en dépendances. L'analyse procède en ajoutant des étiquettes fonctionnelles aux sorties d'un analyseur en constituants, entraîné sur le French Treebank, pour permettre l'extraction de dépendances typées. D'une part, nous spécifions d'un point de vue formel et linguistique les structures de dépendances à produire, ainsi que la procédure de conversion du corpus en constituants (le French Treebank) vers un corpus cible annoté en dépendances, et partiellement validé. D'autre part, nous décrivons l'approche algorithmique qui permet de réaliser automatiquement le typage des dépendances. En particulier, nous nous focalisons sur les méthodes d'apprentissage discriminantes d'étiquetage en fonctions grammaticales.

Abstract. This paper describes a technique for both constituent and dependency parsing. Parsing proceeds by adding functional labels to the output of a constituent parser trained on the French Treebank in order to further extract typed dependencies. On the one hand we specify on formal and linguistic grounds the nature of the dependencies to output as well as the conversion algorithm from the *French Treebank* to this dependency representation. On the other hand, we describe a class of algorithms that allows to perform the automatic labeling of the functions from the output of a constituent based parser. We specifically focus on discriminative learning methods for functional labelling.

Mots-clés : Analyseur syntaxique statistique, analyse en constituants/dépendances, étiquetage en fonctions grammaticales.

Keywords: Statistical parsing, constituent/dependency parsing, grammatical function labeling.

1 Problématique

Le problème que nous nous posons ici est de produire une analyse syntaxique statistique à la fois en constituants et en dépendances pour le Français. Si les constituants permettent d'exprimer des généralisations structurales évidentes, les dépendances ont l'avantage de permettre une extraction plus directe des structures argumentales. Elles constituent également un format linguistiquement plus neutre pour l'évaluation de la tâche d'analyse syntaxique, qu'il s'agisse

d'une évaluation entre analyseurs ou d'une évaluation intrinsèque : la mesure Parseval habituellement utilisée pour les constituants (on y compte rappel et précision sur les constituants, où un constituant est correct si son type et ses frontières sont correctes) est connue comme très sensible au schéma d'annotation et au ratio nombre de terminaux / nombre de non-terminaux (Lin, 1995). Une motivation supplémentaire pour obtenir à la fois des constituants et des dépendances est d'ordre pratique : dans la littérature de parsing statistique de l'anglais, l'extraction de dépendances non typées à partir d'arbres de constituants semble plus performante que la production directe d'arbres de dépendances (McDonald *et al.*, 2005). La situation est moins claire cela dit concernant la production d'arbres de dépendances typées.

L'approche pratique pour obtenir cette analyse statistique est de s'appuyer sur un analyseur statistique en constituants, décrit par (Crabbé & Candito, 2008), analyseur appris sur le French Treebank (Abeillé *et al.*, 2003) ci-après FTB. Un module supplémentaire prend comme entrée les sorties en constituants de cet analyseur et les convertit en dépendances typées comme illustré en Figure 1.

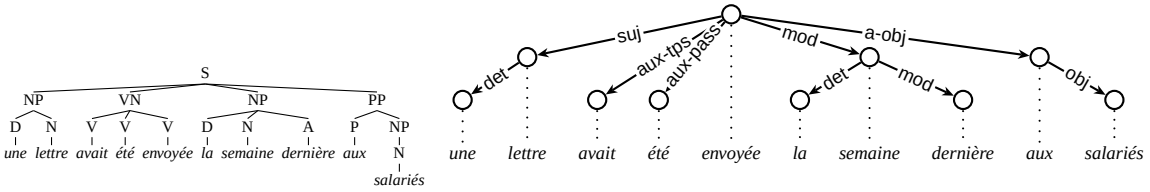

FIGURE 1 – Entrées et sorties de la tâche traitée

Les structures syntagmatiques fournies par l'analyseur statistique en constituants suivent le schéma d'annotation du FTB, moins les annotations fonctionnelles. Or, sans ces annotations, la fonction des dépendants n'est pas directement déductible de la forme des arbres (par exemple, dans la Figure 1, le NP postverbal *la semaine dernière* est un modifieur temporel mais pourrait structurellement être un objet). C'est bien pour cette raison que les auteurs du FTB ont encodé sur les constituants la fonction des dépendants de verbes, les autres cas étant considérés comme déductibles de la seule structure syntagmatique (Abeillé & Barrier, 2004).

Le propos de cet article est de montrer comment tirer parti de cette information supplémentaire pour extraire une analyse en dépendances à partir de l'analyse en constituants. Nous spécifions en section 2 la représentation en dépendances visées, et la procédure de conversion du FTB vers des dépendances. La section 3 détaille l'architecture d'analyse, et motive un processus en deux étapes. La section 4 présente plus particulièrement la tâche d'étiquetage fonctionnel. Nous terminons avec des comparaisons avec l'existant et des perspectives d'amélioration.

2 Annotation en dépendances

Il existe de multiples schémas d'annotation en dépendances, comme le schéma Easy (Paroubek *et al.*, 2005), ou des standards internationaux tels le schéma GR (Carroll *et al.*, 1998), défini comme un format plus adapté à l'évaluation de parsers, le schéma Stanford Dependencies (catherine De Marneffe *et al.*, 2006), issu d'une conversion automatique du Penn TreeBank, ou le schéma PARC 700 (King *et al.*, 2003), inspiré des structures fonctionnelles de LFG. La multiplicité de ces schémas d'annotation tient pour partie à des choix linguistiques et pratiques différents, notamment sur le caractère surfacique ou pas des dépendances. Ainsi par exemple les

relations de contrôle sont sensées être encodées pour les 4 formats cités, mais les dépendances sur mots sémantiquement vides n'apparaissent que pour GR et Stanford. Nous avons préféré définir un schéma d'annotation uniquement en dépendances, et en dépendances purement *de surface*, dont le format soit un pivot convertible vers les différents standards cités, et un pivot enrichissable ou convertible en dépendances plus profondes.

L'objectif pratique à court terme était de convertir vers ce format en dépendances les arbres syntagmatiques du FTB, et plus généralement les analyses fournies par les parsers appris sur le FTB, ce qui permet d'évaluer les dernières sur les premières. Nous décrivons ici le schéma d'annotation en dépendances choisi, la procédure de conversion du FTB vers ce schéma, et l'écart entre l'annotation visée et l'annotation obtenue automatiquement.

2.1 Caractéristiques linguistiques et formelles

Caractéristiques linguistiques générales : On entend ici spécifier des dépendances : la dépendance syntaxique représente le fait que la présence d'un mot [1] est légitimée par un autre mot, son gouverneur [2]. De plus il s'agit de dépendances "surfaciques", c'est-à-dire des relations entre formes fléchies, où toute forme fléchie, même sémantiquement vide, est représentée, et a un et un seul gouverneur, sauf pour la forme tête de la phrase [3]. Donc, par exemple, les relations de contrôle ne sont pas encodées dans ce schéma.

Caractéristiques formelles : Formellement, ces caractéristiques linguistiques impliquent que la structure de dépendance associée à une phrase est un arbre orienté - un graphe acyclique et connexe - dont les noeuds correspondent aux tokens de la phrase, et dont les arcs correspondent aux dépendances et sont étiquetés par une relation de dépendance. Les noeuds sont étiquetés par une forme fléchie, un lemme et sa catégorie syntaxique. Les catégories sont celles du FTB. L'ordre linéaire de la phrase est encodé par un identifiant sur les noeuds, de type entier, local à chaque phrase. A noter que le schéma d'annotation n'impose pas la projectivité : la projection d'un noeud [4] peut correspondre à un segment discontinu de la phrase. L'arbre de dépendances pour une phrase est encodé avec un format parenthésé, où les dépendances apparaissent sous la forme de triplets *relation* (*gouverneur~id linéaire* , *dépendant~id linéaire*).

Spécifications pour le français : Nous définissons un schéma d'annotation en dépendances de surface pour le français (ci-après l'annotation *deps*), qui suit largement l'annotation en constituants préconisée dans le FTB, avec toutefois quelques cas où l'annotation en constituants n'est pas assez précise, et des cas de divergences [5]. En comparaison du format standard français EASy (Paroubek *et al.*, 2005), le propos est d'obtenir un format entièrement en dépendances, et clairement de surface.

La liste des relations utilisées comprend les fonctions définies dans le FTB (où les fonctions ne sont annotées que pour les dépendants de verbes). Nous étendons l'utilisation de ces fonctions dans deux cas, non annotés dans le FTB : (i) les dépendants dans une participiale passée et (ii)

1. Ou plutôt du groupe de mots incluant celui-ci et tous ses dépendants.
2. (Kahane, 2001) pour un historique de la représentation en dépendances
3. Pour obtenir que tout mot a exactement un gouverneur, on ajoute un noeud *root* comme gouverneur de la tête.
4. La projection d'un noeud N est définie comme les noeuds du sous-arbre de N. Les mots correspondant à ces noeuds peuvent être ordonnés selon l'ordre linéaire de la phrase.
5. Les spécifications complètes sont disponibles à l'adresse : `http://docs.google.com/Doc?id=dhm896nk_2dcpgfkdw&hl=en`

les dépendants adverbiaux réduits à un seul mot comme dans *ils sont là*. On ajoute par ailleurs les relations *aux_tps, aux_pass, aux_caus* pour les auxiliaires et *aff* pour les clitiques figés.

Pour les autres catégories de gouverneurs, on utilise les relations *mod, mod_rel, coord, arg, arg_coord, arg_comp, arg_cons, det* et *ponct*. On ne gère pas la sous-catégorisation des têtes non verbales. Donc, dans le cas de dépendants prépositionnels, pour un gouverneur non verbal, on ne distingue pas les arguments et les ajouts, et on utilise la relation générique *dep* par défaut. On choisit pour la coordination un schéma où le premier conjoint gouverne le coordonnant, et le coordonnant gouverne la suite de la coordination (Exemple (d) en Figure 2). Concernant les prépositions, on choisit d'encoder la préposition comme gouverneur, que l'objet de la préposition soit nominal ou infinitival, et que la préposition soit sémantiquement pleine ou vide. A noter que le FTB encode comme préposition les cas de *de* ou *à* introduisant une infinitive directe, comme dans *Paul promet de partir*. Nous gardons la catégorie prep du FTB et codons *obj(promet~1,de~2), obj(de~2,partir~3)*. Pour les conjonctions introduisant une phrase, on uniformise également le traitement, que la conjonction soit sémantiquement pleine ou vide : la conjonction est gouverneur du verbe de la phrase qu'elle introduit (ou autre catégorie de syntagme en cas d'ellipse), avec une dépendance de type *obj*.

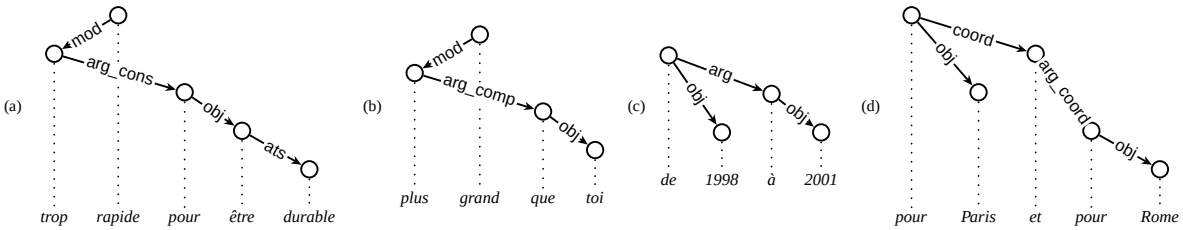

FIGURE 2 – Exemples d'arbres de dépendances

Cas de non-projectivité : Les cas de non projectivité recensés actuellement sont au nombre de quatre : *Extraction* : Dans *A ces cinq départs, trois autres sont susceptibles de s'ajouter*, le gouverneur de *départs* est *ajouter* ; *Extraction hors du SN (clitique* en*)* : Pour *afin d'en améliorer l'efficacité*, le gouverneur de *en* est *efficacité*. *Extraction hors du SN (relatif* dont*).* Il y a non projectivité dans le cas où le SN est objet dans la relative : dans *Lyonnaise Espana, dont le groupe français ne détiendra plus que 51%*, le gouverneur de *dont* est *%*. *Comparatives :* on choisit de faire dépendre la comparative de l'adverbe comparatif. Aussi on aura non projectivité dès une discontinuité entre l'adverbe et la comparative : Dans *La croissance est actuellement plus faible que ce qu'il prévoyait*, le gouverneur de *que* est *plus*. Même principe pour les infinitives consécutives comme dans *trop rapide pour être durable* (exemple (a) en figure 2).

2.2 Conversion du FTB

On décrit ici la conversion d'un arbre syntagmatique de type FTB (on parlera dans la suite de type *ftb-f*) vers le schéma *deps* défini ci-dessus. La conversion se fait en trois étapes, dont les deux premières sont automatiques et la troisième est manuelle. Pour faciliter la lecture, on nomme les formats intermédiaires, et on peut décrire la conversion par : *ftb-f* =auto⇒ *deps-ftb* =auto⇒ *deps-ftb-f*+ =manuel⇒ *deps*.

Isomorphie constituants/dépendances (*ftb-f* ⇒ *deps-ftb*) : Dans cette première étape on commence par un marquage automatique de la tête de chaque constituant, en utilisant une table de propagation de têtes (initialement proposée par (Magerman, 1995), nous utilisons une table

modifiée à partir de (Arun & Keller, 2005)). Cette table contient des règles du type “pour un constituant NP, la tête est le premier N en partant de la gauche s’il existe, sinon le premier A, sinon ...”. Ceci permet de remonter sur chaque noeud syntagmatique sa tête lexicale. Ensuite les dépendances peuvent être extraites ainsi : pour chaque constituant C, dont le fils tête est un noeud H, annoté avec la tête lexicale h, pour chaque noeud F, frère du noeud H, annoté avec la tête lexicale f, on extrait la dépendance *dep(h, f)*. Si en outre le noeud F porte une étiquette fonctionnelle func, alors on type la dépendance extraite : *func(h, f)*. Cette étape produit des dépendances partiellement sous-spécifiées (*deps-ftb*), i.e. où le type de dépendance n’est renseigné que pour les cas où une étiquette fonctionnelle est présente dans l’entrée syntagmatique.

Typage des dépendances non typées dans le FTB **(*deps-ftb* $\Rightarrow$ *deps-ftb-f*+)** : Une deuxième étape consiste à préciser les types de dépendances pour les cas non renseignés par l’étape précédente. Nous utilisons des heuristiques, initialement écrites par Mathieu Falco [6], utilisant simplement la structure syntagmatique et la structure de dépendances partielles.

Révision manuelle (*deps-ftb-f*+$\Rightarrow$ *deps*) : Les deux étapes précédentes ne peuvent, en l’état actuel des heuristiques, coder certaines spécifications de l’annotation *deps*, notamment pour les cas de non projectivité cités supra : la conversion ne crée jamais de non-projectivité. Pour évaluer la qualité des dépendances produites automatiquement, et l’écart par rapport à l’annotation visée, et juger de l’opportunité d’une phase de révision manuelle de tout le corpus, nous avons corrigé manuellement la conversion de 120 phrases consécutives du FTB. Ce sous-corpus (ci-après P7-120) comporte 2894 mots [7] sans compter la ponctuation.

Evaluation : L’évaluation consiste à calculer rappel, précision et F-score des dépendances produites automatiquement par rapport à celles révisées manuellement, pour le sous-corpus P7-120. On ignore les cas où la dépendance dans la référence est de type *ponct*. On obtient F_1=98.00% en dépendances typées [8], et F_1=98.78% en dépendances non typées (i.e. en ignorant le type de dépendances pour compter les dépendances correctes, mais toujours en écartant les dépendances de type *ponct*). Cela permet de conclure que le P7-120 ne contient pas plus de 1,22% de dépendances non projectives [9]. Les erreurs supplémentaires de typage sont dues au caractère trop grossier des heuristiques.

Le résultat de la conversion automatique du FTB, et le sous-corpus manuellement validés sont disponibles sous réserve de licence du FTB. Le corpus complet, correspond à une “pseudo-référence” en dépendances de surface, qui, si l’on projette l’évaluation du P7-120, contient environ 2% d’erreurs.

3 Analyse syntaxique automatique

Dans le cadre de l’analyse syntaxique automatique, on peut adapter la procédure de conversion de treebank donnée précédemment en faisant produire à l’analyseur automatique une sortie

6. Dans le cadre d’un stage de master 1.

7. Les mots composés sont codés comme un seul token, comme décrit dans (Crabbé & Candito, 2008).

8. Dans le cas présent avec exactement une dépendance par mot, rappel et précision se confondent, et valent le pourcentage de mots qui reçoivent le bon gouverneur, avec le bon type de dépendances.

9. Une analyse des 34 erreurs, i.e. des 34 mots qui ne reçoivent pas le bon gouverneur par la procédure automatique de conversion, donne 18 erreurs dues à des dépendances non projectives (*dont, en*, extraction non bornée), 8 erreurs dues à un manque de précision de la table de propagation des têtes, et 8 erreurs dans l’annotation en constituants initiale.

analogue à l'entrée de la procédure de conversion en dépendances sur le French Treebank. Pour cela, il faut pouvoir ajouter les étiquettes fonctionnelles sur les dépendants verbaux. Dans ce qui suit nous utilisons le protocole expérimental décrit par (Crabbé & Candito, 2008) qui divise le corpus en trois parties : entrainement : 80% , développement 10% et test 10%. L'analyseur en constituants utilisé est l'analyseur faiblement lexicalisé de Berkeley (décrit pour le Français par (Candito *et al.*, 2009) comme supérieur, en constituants et en dépendances non typées, aux analyseurs dits lexicalisés).

Analyse intégrée La manière la plus évidente de réaliser cela, c'est d'apprendre une grammaire dont les symboles ne sont pas uniquement des constituants (comme `NP`) mais des symboles intégrant l'information fonctionelle (comme `NP-SUJ`) pour les noeuds comportant cette annotation dans le treebank. Cette approche, bien que simple et directe, donne un mauvais résultat : le F-Score de l'analyse en constituants chute de 86.4 à 78.8. Cette dégradation de résultats est peut être due à deux facteurs : premièrement la division des symboles non terminaux comme `NP` en symboles plus raffinés (comme `NP-SUJ, NP-OBJ,NP-MOD...`) entraîne une démultiplication du nombre de règles et par conséquent un accroissement des effets de dispersion de données. La seconde explication est que les informations fonctionnelles attachées aux dépendants verbaux encodent la sous-catégorisation verbale. La tête verbale a un rôle prépondérant pour décider de la fonction du dépendant. Or l'approche intégrée ne modélise pas la sous-catégorisation, car la règle de grammaire supposée émettre les dépendants est émise indépendamment de la tête verbale, par définition de PCFG et par configuration des arbres du treebank.

Analyse séquentielle Pour remédier à ces deux problèmes, on opte ici pour une analyse en séquence : une première passe procède à l'analyse en constituants dépourvue d'étiquettes fonctionnelles en utilisant un modèle faiblement lexicalisé qui se révèle meilleur pour modéliser la constituance. Une seconde passe d'étiquetage fonctionnel procède à l'ajout des étiquettes de fonction en utilisant un modèle lexicalisé, c'est-à-dire un modèle où les fonctions sont émises en tenant compte (entre autres) de la tête verbale. L'analyse séquentielle permet de préserver un meilleur F-Score en constituants : ainsi le F-score obtenu avec le meilleur modèle séquentialisé décrit ci-dessous est de 83.3 comparé à 78.8 pour l'approche intégrée.

4 Modèles discriminants pour l'étiquetage fonctionnel

Cette section décrit un système automatique d'étiquetage en fonctions grammaticales qui opère en aval de l'analyse en constituants. Ce système repose sur des méthodes d'apprentissage supervisé, en particulier des modèles discriminants. Ces modèles sont plus expressifs que PCFG car ils permettent de prendre des décisions sur base de traits non locaux à une règle de grammaire. [10]

Notre système cible les relations de dépendance entre un prédicat verbal et les syntagmes qui apparaissent comme frères du noeud verbal dans le FTB [11]. On appellera *gouverneur* (ou g) le verbe et *dépendant* (ou d) le noeud syntagmatique relié au verbe. Le FTB distingue huit types distincts de fonctions grammaticales, à savoir : `SUJ`, `OBJ`, `A-OBJ`, `DE-OBJ`, `P-OBJ`, `MOD`, `ATS`, `ATO`.

De manière générale, la tâche d'étiquetage fonctionnel consiste à prédire une séquence de n

10. Leur utilisation lors de l'analyse est cepedendant difficile, car ils ne construisent pas de structure (ou alors de manière beaucoup trop coûteuse en temps) contrairement à un modèle génératif comme PCFG.

11. Nous suivons en cela le guide d'annotation du FTB, puisque seuls les noeuds frères de VN, à l'exclusion des noeuds COORD, sont en effet éligibles pour un étiquetage fonctionnel.

fonctions grammaticales $\{f_1, \ldots, f_n\}$ étant donné un gouverneur g et une séquence de n dépendants $\{d_1, \ldots, d_n\}$. Vu en termes probabilistes, le but de l'apprentissage revient à estimer la probabilité conditionnelle suivante : $p(\{f_1, \ldots, f_n\}|g, \{d_1, \ldots, d_n\})$. Il est bien entendu impossible d'estimer correctement de telles probabilités pour des problèmes évidents de dispersion des données. En pratique, on peut néanmoins simplifier le problème en faisant l'hypothèse d'*indépendance* suivante :

$$p(\{f_1, \ldots, f_n\}|g, \{d_1, \ldots, d_n\}) \approx \prod_{i=1}^{n} p(f_i|g, d_i) \quad (1)$$

Ce qui revient à supposer que les étiquetages de chaque fonction f_i grammaticale à un dépendant d_i se font indépendamment les unes des autres. La tâche d'étiquetage fonctionnel se réduit alors à un simple problème de *classification* (où chaque dépendant se voit assigner une fonction syntaxique unique). La tâche d'apprentissage revient, quant à elle, à apprendre une fonction de D dans F, où D représente l'ensemble des dépendants et F l'ensemble des 8 fonctions grammaticales mentionnées. Concrètement, chaque dépendant est représenté par un vecteur de traits, qui décrit le dépendant, son contexte, et sa relation au gouverneur. Les différents schémas de traits utilisés sont résumés dans le tableau 1. Le processus d'extraction de ces différents traits à partir des structures de consistuance est présenté dans la figure 3.

Trait	**Description**
C_N	catégorie du noeud à classifier
C_D	cat. synt. de la tête du noeud dépendant
C_H	cat. synt. de la tête
W_D	forme lexicale stemmée du dépendant
W_H	forme lexicale stemmée de la tête verbale
$dist$	distance en nombre de mots entre la tête du dépendant et la tête verbale
$span$	longueur en nombre de mots de la chaine dominée par le noeud dépendant
C_P	cat. synt. du noeud parent (noeud dominant le noeud à classifier)
LC_D	cat. synt. du noeud gauche immédiatement adjacent à D (`-STOP-` s'il n'existe pas)
RC_D	cat. synt. du noeud droit immédiatement adjacent à D (`-STOP-` s'il n'existe pas)
C_{CH}	cat. synt. de la co-tête (`-NONE-` si la co-tête n'existe pas)
W_{CH}	forme lexicale stemmée de la co-tête (`-NONE-` si la co-tête n'existe pas)
M_H	mode de la tête syntaxique de la phrase
$rank$	indice du dépendant d_i dans la séquence $d_0, \ldots, d_n$
wh	la phrase est une phrase interrogative (1 ou 0)
rel	la phrase est relative (1 ou 0)
$etre$	le verbe est conjugué avec *être* (1 ou 0)
inv	le verbe est construit avec une inversion clitique (1 ou 0)

TABLE 1 – Principaux traits utilisés par le classificateur

Les traits $W_D, W_H, C_D, C_H, C_{CH}, W_{CH}$ capturent des dépendances bilexicales entre le mot tête et le mot dépendant en incluant de la redondance, comme les catégories, pour obtenir des comptes de granularités variées analogues aux modèles de lissage de (Collins, 1999; Charniak, 2000) pour l'analyse en constituants de l'anglais reposant sur des modèles bayésiens. Ne disposant pas de lemmatiseur déterministe, nous avons procédé pour W_D et W_H à un stemmage brutal : à savoir, les 4 premiers caractères. Les hapax dans le corpus d'entraînement sont remplacés par un symbole unique, permettant ainsi de gérer les mots inconnus. Notons encore que les valeurs des traits $dist$ et $span$ ont été discrétisées également de manière à réduire la dispersion des données.

Les traits $C_P, LC_D, RC_D, dist, span, M_H, rank, wh, rel, etre, inv$ capturent, quant à eux, des informations configurationnelles. Par exemple plus un dépendant est éloigné de la tête, plus il a tendance à être un modifieur, les trait LC_D, RC_D vont par exemple permettre d'identifier si le dépendant est précédé ou suivi d'une ponctuation. Un dépendant séparé de la tête par une

ponctuation sera plutôt un modifieur. Le mode permet par exemple de pénaliser l'assignation de sujets à l'infinitif ou à l'impératif. L'auxiliaire être est une approximation grossière du passif, et doit favoriser l'étiquetage d'une fonction `P-OBJ` (fonction du complément d'agent), etc.

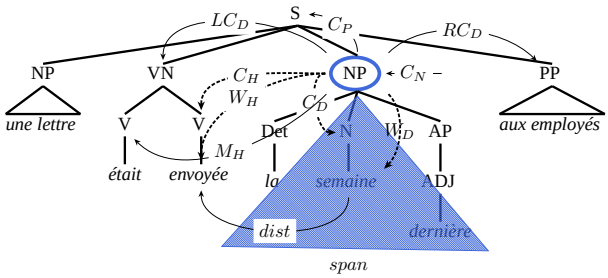

FIGURE 3 – Extraction des traits pour l'étiquetage fonctionnel

Pour l'apprentissage, nous avons eu recours à des modèles log-linéraires (encore appelés modèles par maximum d'entropie) (Berger *et al.*, 1996) [12]. Largement utilisé en TAL, ce type de modèles permet en effet d'incorporer de nombreux traits potentiellement inter-dépendants sans pour cela faire d'hypothèse d'indépendance. [13] Pour nos expériences, nous utilisons le découpage du FTB de (Crabbé & Candito, 2008) en trois parties, modulo le fait que le corpus de test n'est pas l'entière partie 3, mais seulement une sous-partie : le corpus P7-120 révisé manuellement pour les dépendances (section 2). Nous entraînons sur la partie (1), nous avons développé sur la partie (2) et nous testons sur P7-120. [14]

Evaluation du classifieur Une première évaluation consiste à évaluer l'étiquetage fonctionnel seul, en comptant parmi les noeuds éligibles pour un étiquetage fonctionnel, combien reçoivent la bonne étiquette. Nous obtenons une exactitude (angl. *accuracy*) de 87.9 sur la partie test (88.1% sur la partie développement). [15] Les scores de précision, rappel et F_1 pour les différentes fonctions sur le test sont fournis en table 2.

GF	Précision	Rappel	F1
SUJ	0.972	0.963	0.967
OBJ	0.898	0.911	0.904
A_OBJ	0.787	0.747	0.767
DE_OBJ	0.826	0.665	0.737
P_OBJ	0.816	0.213	0.338
ATS	0.852	0.876	0.864
ATO	0.7	0.206	0.318
MOD	0.818	0.9	0.857

TABLE 2 – Performances par fonction

Evaluation de la chaîne d'analyse : Une seconde évaluation concerne la chaîne intégrée : analyse en constituants, premier étiquetage fonctionnel des fonctions de dépendants verbaux, puis conversion en dépendances et étiquetage par heuristiques des autres fonctions (cf. section 2). L'évaluation se fait avec le protocole décrit section 2 pour l'évaluation de la procédure de conversion. Nous obtenons sur le P7-120 une précision F_1=86.56% en dépendances typées.

Evaluation EASy Enfin, une troisième évaluation consiste à évaluer l'analyseur sur le corpus EASy (Paroubek *et al.*, 2005). Pour cela, un module de conversion, réalisé par François Guérin, convertit les dépendances de surface décrites section 2, vers les relations et chunks EASy. Les résultats préliminaires de cette chaîne d'analyse donne un F_1 de 66.0% sur les relations de la

12. L'implémentation utilisée est Megam : `www.cs.utah.edu/~hal/megam`.

13. Notons néanmoins que ces modèles ne modélisent pas explicitement l'interaction entre traits. Aussi, il est souvent d'usage d'introduire de nouveaux traits complexes pour modéliser ces interactions —ce qui n'a pas encore été fait ici. De manière générale, les traits présentés ci-dessus restent encore très rudimentaires, tentant simplement de capturer nos principales intuitions linguistiques. Une exploration plus détaillée des traits pertinents fera l'objet de recherches futures. Nous n'avons pas non plus cherché à optimiser le paramètre de lissage gaussien.

14. Le nombre de dépendants/séquences dans ces sous-corpus sont, respectivement, de : 51865/20623, 6979/2850, 7119/2953.

15. Les scores de précision pour la séquence entière sont, respectivement, de 74.6% et 74.8%. Les longueurs moyennes des séquences est de 2.41/2.45 dépendants.

partie “Le Monde” du corpus EASY.

5 Travaux antérieurs et reliés

(Abeillé & Barrier, 2004) décrivent la tâche d’annotation fonctionnelle du FTB, pour laquelle ils ont utilisé une annotation automatique, faite par règles, préalable à une révision manuelle. Les auteurs annoncent un rappel et une précision de 89.69% et 89.27% [16], comparables aux résultats obtenus ici par approche statistique.

(Schluter & van Genabith, 2008) proposent une architecture de parsing statistique ’profond’ du français, avec un parser appris sur le ’modified French Treebank’ : un corpus comprenant environ la moitié des phrases du FTB, avec modification du schéma d’annotation, correction manuelle, et ajout de chemins fonctionnels encodant les dépendances non bornées. Le parser appris produit des couples structure-c/structure-f de type LFG. Les auteurs évaluent leur parser en rappel/précision/F_1 sur les traits des structures fonctionnelles produites. Ils relatent un F_1 de 86.73%, pour un parsing avec tagging parfait préalable. Ces résultats ne sont pas exactement comparables aux nôtres : d’un côté le tagging parfait augmentent leur score, d’un autre côté, les dépendances encodées dans les structures fonctionnelles sont moins surfaciques et donc constituent une tâche plus difficile.

Plus directement comparables sont les résultats de la campagne d’évaluation EASy (Paroubek *et al.*, 2005). Le meilleur analyseur, Syntex (Bourigault *et al.*, 2005) donne un F_1=66% [17] sur sous-corpus Le Monde compris dans EASy, score équivalent à ceux obtenus ici. Cependant, le parser statistique présenté ici a des performances nettement moins bonnes lorsque l’on change de domaine (par exemple F_1=61% sur le corpus médical, contre F_1=70% pour l’analyseur Syntex).

6 Conclusions et perspectives

Cet article présente une architecture pour l’analyse syntaxique statistique du français journalistique, qui repose sur un processus séquentiel : analyse en constituants d’abord, extraction de dépendances ensuite.

Du point de vue de la modélisation, l’architecture présentée ici fait deux hypothèses simplificatrices essentielles : le modèle génératif en constituants est vu comme indépendant du modèle discriminant d’analyse fonctionnelle et l’étiquetage fonctionnel assigne les fonctions indépendamment à chaque dépendant alors qu’on sait qu’il s’agit d’un problème de séquence, qui doit modéliser une notion de sous-catégorisation. Il reste à investiguer comment lever ces deux hypothèses. La première pourrait être levée en reformulant le modèle discriminant comme un modèle de reranking sur une analyse en constituants intégrée. La seconde demande de modéliser explicitement des séquences de dépendants (incluant arguments et modifieurs) de manière à modéliser des cadres de sous-catégorisation.

Du point de vue plus linguistique cette fois, il s’agira d’exploiter les dépendances de surface

16. Evaluation sur mille phrases du FTB.

17. Scores disponibles sous `http://w3.erss.univ-tlse2.fr:8080/index.jsp?perso=bourigault&subURL=syntex.html`.

(issues de la conversion du FTB ou bien du résultat du parser) pour obtenir des dépendances plus profondes (relations de contrôle, suppression des mots sémantiquement vides...).

Références

ABEILLÉ A. & BARRIER N. (2004). Enriching a french treebank. In *LREC 2004*, Lisbon.

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). *Building a treebank for French*, In *Treebanks*. Kluwer : Dordrecht.

ARUN A. & KELLER F. (2005). Lexicalization in crosslinguistic probabilistic parsing : The case of french. In *ACL 2005*, p. 306–313, Ann Arbor, MI.

BERGER A., PIETRA S. D. & PIETRA V. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**(1), 39–71.

BOURIGAULT D., FABRE C., FRÉROT C., JACQUES M.-P. & OZDOWSKA S. (2005). Syntex, un analyseur syntaxique de corpus. In *TALN 2005, Atelier EASy : campagne d'évaluation des analyseurs syntaxiques*, Dourdan.

CANDITO M., CRABBÉ B. & SEDDAH D. (2009). On statistical parsing of french with supervised and semi-supervised strategies. In *EACL 2009 Workshop Grammatical inference for Computational Linguistics*, Athens.

CARROLL J., BRISCOE T. & SANFILIPPO A. (1998). Parser evaluation : a survey and a new proposal. In *LREC 1998*, Granada.

CATHERINE DE MARNEFFE M., MACCARTNEY B. & MANNING C. D. (2006). Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

CHARNIAK E. (2000). A maximum entropy inspired parser. In *NAACL 2000*, p. 132–139, Seattle, WA.

COLLINS M. (1999). *Head driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania, Philadelphia.

CRABBÉ B. & CANDITO M. (2008). Expériences d'analyse syntaxique statistique du français. In *TALN 2008*, p. 45–54, Avignon.

KAHANE S. (2001). Grammaires de dépendances formelles et théorie sens-texte. In *TALN 2001*, Tours, France.

KING T., CROUCH R., RIEZLER S., DALRYMPLE M. & KAPLAN R. (2003). The parc 700 dependency bank. In *EACL workshop on Linguistically Interpreted Corpora*, Budapest.

LIN D. (1995). A dependency-based method for evaluating broad-coverage parsers. In *IJCAI 1995*, p. 1420–1425, Montreal.

MAGERMAN D. (1995). Statistical decision-tree models for parsing. In *ACL 1995*, p. 276–283, Morristown.

MCDONALD R., CRAMMER K. & PEREIRA F. (2005). *Spanning Tree Methods for Discriminative Training of Dependency Parsers*. UPenn CIS Technical Report MS-CIS-05-11, University of Pennsylvania.

PAROUBEK P., POUILLOT L.-G., ROBBA I. & VILNAT A. (2005). Easy : campagne d'évaluation des analyseurs syntaxiques. In *Actes de TALN'05, Atelier EASy : campagne d'évaluation des analyseurs syntaxiques*, Dourdan.

SCHLUTER N. & VAN GENABITH J. (2008). Treebank-based acquisition of LFG parsing resources for french. In *LREC 2008*, Marrakech, Morocco.