Stratégies pour l'étiquetage et l'analyse syntaxique statistique de phénomènes difficiles en français : études de cas avec Talismane

Assaf Urieli[*],[**]

[*] *CLLE-ERSS, CNRS, Université de Toulouse - assaf.urieli@univ-tlse2.fr*
[**] *Joliciel Informatique, 09000 Foix, France*

RÉSUMÉ. *Les outils statistiques robustes en TAL, tels que les étiqueteurs morphosyntaxiques et les analyseurs syntaxiques, utilisent souvent des descripteurs « pauvres », qui peuvent être appliqués facilement à n'importe quelle langue, mais ne prennent pas en compte les particularités de la langue. Dans cette étude, nous cherchons à améliorer l'analyse de deux phénomènes en français en injectant des connaissances plus riches : l'étiquetage morphosyntaxique du mot* que *et l'analyse syntaxique de la coordination. Nous comparons plusieurs techniques : la transformation automatique du corpus vers d'autres normes d'annotation avant l'entraînement, l'ajout de descripteurs ciblés et riches lors de l'entraînement, et l'ajout de règles symboliques qui contournent le modèle statistique lors de l'analyse. Nous atteignons une réduction du taux d'erreur de 55 % pour l'étiquetage de* que*, et de 37 % pour les structures coordonnées.*

ABSTRACT. *Robust statistical NLP tools, such as pos-taggers and syntax parsers, often use knowledge-poor features, which can easily be applied to any language, but which do not take into account language specifics. In this study, we attempt to improve analysis for two French phenomena by injecting richer knowledge: pos-tagging of the function word* que *and syntax parsing of coordinated structures. We compare several techniques: automatic transformation of the corpus to other annotation standards prior to training, addition of rich targeted features during training, and addition of symbolic rules during analysis. We attain 55% error reduction for the pos-tagging of* que*, and 37% for coordinated structure parsing.*

MOTS-CLÉS : *analyse syntaxique statistique en dépendances, étiquetage morphosyntaxique, descripteurs riches, normes d'annotation.*

KEYWORDS: *statistical dependency parsing, pos-tagging, rich features, annotation schema.*

1. Introduction

Les outils statistiques robustes en TAL sont relativement faciles à construire : il suffit de disposer d'un corpus annoté (*e.g.* le French Treebank), d'un classifieur robuste (*e.g.* SVM ou *support vector machine* linéaire), d'un algorithme d'analyse (*e.g.* le *parsing* par transitions pour l'analyse syntaxique) et de quelques descripteurs. La plupart de ces systèmes utilisent des descripteurs linguistiquement pauvres, limités aux bigrammes ou trigrammes des tokens ou des étiquettes morphosyntaxiques, à quelques informations de base tirées d'un lexique à large couverture, et, au niveau du *parsing*, à un examen superficiel de la tête ou du dépendant le plus à droite ou à gauche d'un token donné. Même les études qui parlent de descripteurs « riches » (Zhang et Nivre, 2011) se limitent à des descripteurs génériques, qui prennent en compte des informations de surface telles que la valence d'un token (nombre de dépendants) ou la distance entre deux tokens, mais ne cherchent pas à coder les phénomènes spécifiques d'une langue donnée. Cela présente l'avantage de pouvoir s'appliquer facilement à beaucoup de langues, mais empêche l'injection des connaissances linguistiques spécifiques, et limite donc les gains d'exactitude réalisables. Notre but principal ici est de trouver des moyens d'améliorer l'analyse des systèmes statistiques par l'introduction d'informations plus riches.

En outre, les normes d'annotation du corpus d'apprentissage ne sont pas choisies pour leur compatibilité avec un algorithme d'analyse donné, dans notre cas le *parsing* par transitions. La plupart des études ne remettent pas en question ces normes, et pourtant, il peut s'avérer très intéressant de les transformer automatiquement sans perte d'information vers une forme plus facile à analyser.

L'analyseur syntaxique Talismane[1] a été développé dans l'optique de permettre à l'utilisateur d'injecter le maximum d'informations linguistiques, dans un système qui reste statistique et robuste (Urieli, 2013). Il comprend quatre modules statistiques enchaînés : la segmentation en phrases, la segmentation en mots (*tokenisation*), l'étiquetage morphosyntaxique (*pos-tagging*) et l'analyse syntaxique (*parsing*).

Nous cherchons ici à améliorer l'analyse de certains phénomènes particulièrement difficiles et fréquents en français : l'étiquetage morphosyntaxique du mot fonctionnel *que* et l'analyse syntaxique des structures coordonnées, où le choix des conjoints est souvent ambigu. Ces phénomènes ont été choisis car ils nécessitent l'injection de connaissances linguistiques au délà des descripteurs « pauvres » pour résoudre l'ambiguïté. Pour l'étiquetage de *que*, décrit ici dans la section 4, nous utilisons deux méthodes complémentaires : l'ajout de descripteurs riches lors de l'entraînement, et, lors de l'analyse, l'ajout de règles symboliques, qui imposent ou interdisent des décisions locales, contournant ainsi le modèle statistique. Pour les structures coordonnées, présentées dans la section 5, nous comparons de nouveau deux méthodes : une qui introduit des descripteurs riches, cette fois-ci au niveau du parseur, et une autre qui

1. http://redac.univ-tlse2.fr/applications/talismane.html

compare différentes normes d'annotation équivalentes pour identifier celle qui est la plus compatible avec notre algorithme d'analyse.

2. Méthodes d'injection de connaissance

Dans cette étude, nous abordons l'analyse linguistique comme un problème de classification. Pour l'étiquetage morphosyntaxique cela va de soi : étant donné une liste d'étiquettes possibles, il faut attribuer une étiquette à chaque mot. Pour l'analyse syntaxique, on utilise le *parsing* en dépendances par transitions : étant donné une liste de transitions possibles, il faut choisir la transition à appliquer à chaque étape de l'analyse. Cet algorithme est décrit dans Kübler *et al.* (2009), et repris ici dans la section 5.1. La question ici est : comment améliorer l'exactitude d'un classifieur statistique (*e.g.* SVM linéaire) par l'injection de connaissances linguistiques spécifiques, sans faire exploser le temps d'analyse ni l'effort de maintenance à long terme. Nous explorons cette injection par le moyen de descripteurs et de règles.

Un **descripteur** (*feature* en anglais) spécifie l'information à extraire d'un contexte linguistique donné, qui pourra aider le classifieur à choisir la bonne classe (c'est-à-dire étiquette morphosyntaxique ou transition) de ce contexte. Dans Talismane, un descripteur est défini par une expression qui combine des informations de base, soit par concaténation (pour les chaînes de caractères), soit par des opérations mathématiques ou logiques. Une **règle** est une expression booléenne (de type vrai/faux) définie avec la même grammaire que les descripteurs. Si l'expression s'évalue à *vrai* dans un contexte donné, la règle peut soit imposer le choix d'une certaine classe pendant l'analyse, soit empêcher le système de choisir cette classe.

Par exemple, dans l'étiquetage morphosyntaxique du mot *que*, l'étiquette attribuée au mot précédant le mot *que* peut être utilisée comme descripteur. Si ce mot est un verbe à l'indicatif (`V`), *e.g. « il **faut** que… »*, alors on observe certaines tendances sur l'étiquetage du *que* : dans notre corpus d'apprentissage, sur 523 cas, 77 % sont des conjonctions de subordination (`CS`) et 23 % des adverbes (`ADV`). Un autre descripteur peut porter sur les étiquettes possibles du mot suivant le mot *que* dans un lexique externe de référence. Si ce mot est listé dans le lexique comme verbe à l'indicatif, *e.g. « l'exemple que **fournit** Dupont… »*, alors dans notre corpus d'apprentissage, sur 205 cas, 1 % des *que* sont des `ADV`, 20 % des `CS`, 72 % des `PROREL` (pronom relatif) et 7 % des `PROWH` (pronom interrogatif). Ces informations vont être combinées avec des dizaines d'autres descripteurs pour aider le modèle probabiliste à construire une distribution de probabilités des étiquettes pour un cas donné du mot *que*.

On peut aussi être amené à définir une règle déterministe : *e.g.* si on a une structure de type *« ne `V` que »*, alors on oblige le système à attribuer l'étiquette `ADV`. Cette règle prend priorité sur le modèle probabiliste qui ne sera même pas consulté. Un autre type de règle est la règle négative : *e.g.* si *que* est le premier mot d'une phrase, alors on empêche le système d'attribuer l'étiquette `PROREL`. Dans ce cas, le modèle probabiliste va utiliser tous les descripteurs pour définir une distribution de probabilités des

étiquettes, mais l'étiquette PROREL sera supprimée de cette distribution avant que le système ne choisisse l'étiquette la plus probable.

Un descripteur cherche, par nature, à capter des régularités dans le corpus d'apprentissage qui peuvent se généraliser à d'autres corpus. Il est donc limité aux régularités qui se trouvent dans ce corpus, même si elles peuvent être décrites à l'aide de ressources externes pour les rendre plus généralisables (*e.g.* un lexique qui remplace la forme lexicale par son lemme). Une règle, en revanche, cherche à traduire directement les connaissances linguistiques du concepteur du système. Elle permet donc au système statistique d'aller au-delà des informations qui lui sont directement accessibles. Puisque celle-ci est appliquée uniquement au moment de l'analyse, elle peut aussi traduire des connaissances spécifiques au corpus qu'on est en train d'analyser.

Les descripteurs serviront à alimenter le classifieur SVM, qui va appliquer sa « magie noire » statistique pour donner plus ou moins de poids à chaque descripteur pour chaque étiquette, selon les occurrences trouvées dans le corpus d'apprentissage. À la différence des systèmes précoces tels que Brill (1995), les descripteurs ici sont tous directement mis en concurrence en parallèle pendant l'entraînement, et leur poids relatif déterminé de façon itérative pour maximiser l'exactitude des classes attribuées dans le corpus d'apprentissage. Les descripteurs peuvent donc se contredire et se chevaucher. Ils décrivent des tendances : si X est *vrai*, alors l'étiquette sera plus probablement Y que Y'. Les règles, quant à elles, cherchent à décrire des vérités absolues : si X est *vrai*, alors l'étiquette doit être (ou ne peut pas être) Y. Par conséquent elles sont contraintes à viser des phénomènes très spécifiques et non ambigus. Néanmoins, tout corpus ne couvrant qu'une partie infime des possibilités d'une langue, beaucoup de constructions non ambiguës, telles que les locutions figées, sont sous-représentées ou bien absentes. Ce sont ces constructions que les règles vont cibler.

Reste la question de la facilité de maintenance du système. Pour les descripteurs, autant l'écriture et l'affinage des descripteurs peuvent s'avérer longs et complexes, autant la maintenance du système à long terme est simple, puisque le modèle probabiliste ajuste automatiquement le poids de chaque descripteur au fur et à mesure que d'autres descripteurs sont ajoutés, ou que d'autres données d'apprentissage deviennent disponibles. Pour les règles, la réponse est plus complexe. Il est important de fonder les règles uniquement sur des erreurs effectivement rencontrées dans le corpus de développement, et que l'on peut décrire de façon non ambiguë. Ceci réduit considérablement le nombre de règles, qui sont là uniquement pour compléter le système dans certains cas bien définis : la plupart du travail continue à être fait par les descripteurs.

3. Corpus

Pour cette étude, notre corpus d'apprentissage est la partie française du corpus SPMRL (Seddah *et al.*, 2013), un corpus disponible en dépendances et construit automatiquement à partir du French Treebank (FTB) (Abeillé *et al.*, 2003). Nous avons préservé le découpage en sous-corpus `train` (14 759 phrases, 412 879 tokens), `dev`

(1 235 phrases, 36 272 tokens) et `test` (2 541 phrases, 69 922 tokens). Pour le corpus d'évaluation de l'étiquetage morphosyntaxique, en plus des parties `dev` et `test` du corpus SPMRL, nous avons utilisé la version 4.0 des corpus Sequoia (Candito et Seddah, 2012) et un corpus de pages de discussion du Wikipédia français, FrWikiDisc[2]. Nous utilisons le jeu d'étiquettes décrit dans Crabbé et Candito (2008). Tous les scripts et ressources nécessaires pour effectuer les expériences de cette étude sont disponbiles en ligne[3].

4. Étiquetage morphosyntaxique

Dans Talismane, l'algorithme d'étiquetage morphosyntaxique fonctionne de gauche à droite. Ainsi, les descripteurs peuvent prendre en compte tous les tokens qui se trouvent à gauche et à droite du token à étiqueter, ainsi que les étiquettes déjà attribuées à sa gauche. Comme descripteurs de base, nous utilisons des descripteurs similaires à ceux décrits par Denis et Sagot (2012), faisant un usage massif d'un lexique, en l'occurrence le LeFFF (Sagot, 2010). En particulier, nous utilisons les descripteurs de base suivants : *W* la forme lexicale exacte, *P* l'étiquette attribuée au token (s'il se trouve avant le token actuel) ou les étiquettes trouvées dans le lexique pour ce token (s'il est le token actuel, ou se trouve après celui-ci), *L* le lemme de ce token pour une étiquette donnée, *U* si le token est inconnu dans le lexique, Sfx_{n} les n dernières lettres de la forme. Ces briques de base sont combinées en bigrammes et trigrammes pour les tokens à position – 2, – 1, 0, + 1, + 2 par rapport au token actuel. Au vu de ce jeu de descripteurs, un descripteur plus riche consiste à regarder plus loin que deux tokens à gauche ou à droite, ou à regrouper les tokens en classes d'équivalence à un niveau qui se trouve entre le lemme et l'étiquette morphosyntaxique. Dans la pratique, nous avons utilisé des descripteurs plus complexes qui mettent en œuvre diverses combinaisons logiques des informations de base, décrits ci-dessous.

Pour Talismane v2.4.2 avec une classifieur SVM linéaire de la bibliothèque liblinear (Fan *et al.*, 2008) en version Java[4], les paramètres $\epsilon = 0,01$ et $C = 0,5$, et un *cutoff* de 3 (nombre de fois qu'un descripteur doit apparaître pour être pris en compte), on observe une exactitude de 96,8 sur SPMRL-`test`.

4.1. *Le cas de* que

Les difficultés pour étiqueter le mot *que* ont déjà été explorées par Jacques (2005), qui décrit les différents contextes dans lesquels *que* est utilisé, et qui propose une méthode pour corriger l'étiquetage par un mélange de règles de surface et de corrections appliquées pendant l'analyse syntaxique. Mais, à la différence de cette étude, qui considère des systèmes uniquement à base de règles, nous mettons l'accent ici sur

2. https://github.com/urieli/talismane/tree/master/examples/french/corpus
3. Site Web de l'article : https://github.com/urieli/talismane/tree/master/examples/tal56-1
4. http://liblinear.bwaldvogel.de/

les descripteurs riches, qui alimentent directement un système statistique robuste. Les règles symboliques sont utilisées uniquement en complément des descripteurs, pour des cas très précis et non ambigus.

Pour résumer, il y a six usages principaux du token *que*, annotés selon les normes d'annotation du corpus FTB avec quatre étiquettes différentes, comme illustré par les exemples suivants :

1) Conjonction de subordination (CS) : *Je pense qu'il a trop bu.*

2) Pronom relatif (PROREL) : *Il boit le vin que j'ai acheté.*

3) Pronom interrogatif (PROWH) : *Que buvez-vous ?*

4) Adverbe négatif (ADV) : *Je n'ai bu que trois verres.*

5) Adverbe exclamatif (ADV) : *Qu'il est bon, ce vin !*

6) Construction comparative (CS) : *Il est plus bourré que moi.*

Bien qu'il pourrait être intéressant de remettre en cause l'assimilation des constructions comparatives aux conjonctions de subordination, ainsi que la combinaison des adverbes négatifs et exclamatifs, nous nous limitons dans cette étude aux étiquettes déjà utilisées dans le FTB.

	ADV	**CS**	**PROREL**	**PROWH**	**Total**	**Erreurs**
ADV	90	44	4	1	139	49
CS	37	1 097	61	0	1 195	98
PROREL	0	69	244	0	313	69
PROWH	0	4	2	23	29	6

Tableau 1. *Matrice de confusion de base pour* que

Avec le modèle de base décrit dans le paragraphe précédent, le tableau 1 montre la matrice de confusion pour le mot *que* dans l'ensemble des corpus d'évaluation. Les lignes représentent la bonne étiquette et les colonnes représentent l'étiquette devinée. Nous avons au total 222 erreurs pour 1 676 occurrences, donc une exactitude de 86,75 %. Il est à noter que la confusion se trouve principalement entre CS et ADV d'une part, et entre CS et PROREL d'autre part. Nous traiterons chacun de ces cas séparément.

4.2. Que *comme adverbe négatif*

En termes de descripteurs ciblés, nous traiterons d'abord le cas de *que* en tant qu'adverbe négatif. La première étape consiste à analyser les erreurs dans le corpus SPMRL-dev et concevoir des descripteurs utiles. Dans ce corpus, la plupart des erreurs ressemblent à l'exemple 4.1, où le mot *que* est étiqueté à tort comme CS. Dans ce cas, reconnaître le mot *que* comme adverbe négatif revient à chercher une occurrence du mot *ne* plus tôt dans la même phrase. Il n'y a pas de limitation inhérente de distance car, comme on voit dans l'exemple, plusieurs syntagmes prépositionnels

peuvent séparer les deux particules. En revanche, une autre particule négative peut compléter le *ne*, ce qui rend le *que* ambigu, comme dans les deux exemples 4.2 et 4.3.

Exemple 4.1 *Mais cela **ne** représente dans cette mouture, pour un couple avec deux enfants, **qu'**/ADV une prime maximale.*

Exemple 4.2 *Pour cela, il **n'**est **pas** question **que**/CS le zloty, la monnaie polonaise, soit « l'ancre de la stabilité » de l'économie polonaise.*

Exemple 4.3 *... qui, faute de volonté politique, **ne** fut **jamais que**/ADV la caricature du système français.*

La prochaine étape consiste à écrire ces descripteurs dans la syntaxe de Talismane, et les projeter sur le corpus `train`. Après affinage pour étendre la portée des descripteurs tout en éliminant des cas non voulus afin de maximiser le déséquilibre dans la distribution des étiquettes, nous avons défini une liste consultable sur le site Web de l'article. Voici quelques exemples.

Descripteur 4.1 Le *que* est précédé par un *ne* sans autre particule négative entre les deux. En plus, le *ne* n'est pas lui-même précédé par {*personne*, *rien*, *aucun/e*, *nul/le*}, afin d'exclure des phrases comme *« Personne ne sait que je mange ici. »*. Nous avons 345 cas en tout, dont 312 `ADV` : *« Ils **n'**en comprendront le sens **que**/ADV bien plus tard »* ; et 32 `CS`. Parmi les `CS`, on trouve beaucoup d'erreurs d'annotation. Les autres sont des phrases où la particule *ne* n'est pas complétée, dans des expressions de type *moins ADJ qu'on ne...* : *« L'Amérique, moins superficielle qu'on **ne** l'imagine parfois, a entrepris une réflexion sur son identité bien avant **que**/CS [...] »* ; ou en modifiant le verbe *pouvoir* : *« [...] **ne** peuvent ainsi éviter **que**/CS, en la matière, l'histoire ne se repète »*.

Descripteur 4.2 Il n'y a pas de *ne* précédent le *que*. Nous avons 2 608 cas en tout, dont 1 941 `CS`, 622 `PROREL`, 26 `PROWH` et 19 `ADV`. Parmi les adverbes, 10 sont des erreurs d'annotation ; 5 sont des adverbes exclamatifs : *« Mais pour parvenir à cela, **que**/ADV d'esprits à convaincre en France ! »* ; et 1 est une phrase « informelle » ou l'auteur a laissé tomber le *ne* : *« Il lui manque **que**/ADV le sac à main de Maggie »*.

4.3. Que *comme pronom relatif*

À la différence du *que* adverbe négatif, où la présence d'un *ne* précédent est un indicateur de surface fort, il n'y a pas d'indicateur de surface simple pour distinguer le *que* pronom relatif du *que* conjonction de subordination, étant donné le peu d'informations disponibles à l'étape de l'étiquetage morphosyntaxique. Suivant la méthodologie décrite dans le paragraphe précédent, nous analysons les erreurs du corpus `dev` pour identifier des descripteurs utiles.

Exemple 4.4 *[. . .] la Commission des opérations de bourse (COB) a annoncé le 14 janvier **qu'**/CS elle saisit la justice [. . .]*

L'exemple 4.4 est annoté PROREL plutôt que CS. Nous avons relevé les descripteurs suivants : d'abord *annoncer* est parmi les verbes qui sous-catégorisent un objet direct avec *que* (tels que *admettre*, *affirmer*, *ajouter*, . . .). De plus, le verbe transitif *saisir* a déjà un objet direct (*justice*), ce qui exclut généralement un pronom relatif. Pourtant, à ce stade de l'analyse, nous ne disposons pas d'informations syntaxiques sur les objets directs : à la place il faut faire une analyse superficielle des mots suivant le premier verbe après le mot *que*, qui, selon le lexique, sont potentiellement un déterminant et un nom. Finalement, noter que l'ambiguïté entre PROREL et CS existe uniquement quand il y a un nom qui peut servir d'antécédent entre le verbe précédent et le *que*, dans ce cas *janvier*. La nature de ce nom est un indicateur : les expressions de temps, dont les noms des mois, sont très souvent des circonstants. Ils remplissent rarement l'argument d'objet direct, et sont rarement modifiés par une proposition relative.

Exemple 4.5 *Le gouvernement va présenter dans un délai de trois mois les dispositions **qu'**/PROREL il entend retenir [. . .]*

L'exemple 4.5 est annoté CS plutôt que PROREL. C'est le cas contraire de l'exemple précédent : le verbe *présenter* a déjà un objet direct (*dispositions*) et ne sous-catégorise pas un objet direct avec *que*, alors que le verbe transitif *retenir* n'a pas d'objet direct qui le suit. Nous voyons ici l'importance de reconnaître les verbes qui sous-catégorisent avec *que*. Le corpus train contient 245 verbes différents qui répondent à ce critère. Nous avons choisi manuellement 152 de ces verbes qui nous semblaient les plus aptes à préférer cette sous-catégorisation.

Finalement, dans l'exemple 4.6 nous avons d'autres indicateurs : certains noms introduisent des propositions subordonnées (*e.g. fait*), et le subjonctif (*aient*) indique généralement qu'on a affaire à une subordonnée indépendante plutôt que relative.

Exemple 4.6 *Le fait **qu'**/CS ils aient accepté de reprendre les pourparlers est interprété de façon positive.*

Après projection des descripteurs sur le corpus train et affinage, nous avons retenu une liste assez longue, consultable sur le Web. Voici quelques exemples :

Descripteur 4.3 Verbe précédent sous-catégorise avec *que*. Si le verbe précédent sous-catégorise avec *que* (total 98 cas), les deux étiquettes sont donc distribuées de façon à peu près égale : 50 CS, *e.g. « Helmut Kohl a **annoncé** à l'automne **que**/CS des hausses d'impôts seraient nécessaires en 1994 »* ; et 48 PROREL, *e.g. « Il a toutefois **refusé** à Mr Vernay les 100 000 francs de dommages et intérêts **que**/PROREL celui-ci réclamait »*. Dans le cas contraire (total 126 cas), nous avons 113 PROREL, et uniquement 12 CS, dont 10 erreurs d'annotation.

Descripteur 4.4 Le verbe qui précède a un objet direct. Le verbe précédant le *que* est-il suivi directement d'un déterminant et d'un nom, en dehors des noms représentant les expression de temps (*e.g. la semaine dernière*) ? Sur 63 cas, nous avons 56 PROREL : *« En revanche, la CGT **dénonce un texte qu'**/PROREL elle juge décrédibilisé »* ; et 7 CS : *« Nous avons **obtenu l'assurance** du premier ministre **que**/CS la suppression du recours [...] »*, dont 6 sont des erreurs d'annotation.

Descripteur 4.5 Le verbe qui suit a un objet direct. Le verbe suivant le *que* est-il suivi directement d'un déterminant et d'un nom, en dehors des noms représentant les expressions de temps ? Résultats : 93 CS.

Descripteur 4.6 *Que* suivi d'un verbe subjonctif. Le *que* est-il suivi d'un verbe d'une forme clairement subjonctive ? Étant à droite du token actuel, c'est le lexique qui doit reconnaître les tokens qui peuvent représenter des verbes. Il fallait éliminer des cas où le token avait aussi une étiquette non verbale dans le lexique, comme le nom *émissions* (imparfait du subjonctif du verbe *émettre*). Résultat : 80 CS : *« Faut-il encore **que**/CS l'ambiance non seulement le **permette** mais aussi le favorise »* ; et 1 PROREL : *« Faut-il en conclure que le mieux **qu'**/PROREL on **puisse** attendre, c'est le chacun-pour-soi ? »*

4.4. *Résultats pour les descripteurs ciblés*

	ADV	CS	PROREL	PROWH	Total	Erreurs
ADV	133 (+43)	6 (−38)	0 (−4)	0 (−1)	139	6 (−43)
CS	10 (−27)	1 135 (+38)	50 (−11)	0 (−1)	1 195	60 (−38)
PROREL	0	52 (−17)	261 (+17)	0	313	52 (−17)
PROWH	0	0 (−4)	4 (+2)	25 (+2)	29	4 (−2)

Tableau 2. *Matrice de confusion pour* que *avec les descripteurs ciblés*

Le tableau 2 montre la matrice de confusion pour *que* après l'ajout des descripteurs riches. Les résultats sont considérablement améliorés pour toutes les catégories, mais plus particulièrement pour la confusion entre ADV et CS. En tout, nous avons supprimé 45 % des erreurs, passant d'une exactitude de 86,75 % à 92,72 %. Les résultats sont hautement significatifs, avec 139 nouvelles corrections pour 29 nouvelles erreurs (test de McNemar, p-valeur $< 0{,}001$). Pourtant, ces gains ont un prix : la vitesse d'analyse. Dans la version de base, on étiquette 1 million de mots en 6 min 48 s. Avec les descripteurs ciblés, cela prend 1,5 fois plus de temps : 10 min 09 s.

4.5. *Les règles*

Dans le paragraphe précédent, certains cas n'ont pas été corrigés même quand les descripteurs riches ajoutaient du poids à la bonne étiquette : les descripteurs riches

semblaient noyés dans un océan de descripteurs plus pauvres et génériques, ce qui les empêchait de faire pencher la balance en faveur de la bonne étiquette. En analysant les erreurs restantes, nous avons identifié certaines règles qui nous semblaient généralisables. Vu la rareté des phénomènes qui peuvent être ciblés par des règles, nous avons examiné ici les erreurs dans tous les corpus sauf SPMRL-`test` et EMEA-`test`, à la différence de l'expérience avec les descripteurs, où seulement les erreurs de SPMRL-`dev` avaient été examinées. La liste complète des règles est consultable sur le Web. Voici quelques exemples.

Règle 4.1 Étiqueter `CS` dans les locutions de type *attendre / veiller / tenir à ce que*, *n'empêche que*, *dommage que*, *avoir honte à ce que*, *le / du / au fait que*, *une fois que*.

Règle 4.2 Étiqueter `PROREL` dans *ceux / celui / celle / celles / quoi / qui / quel / quelle / quels / quelles / où que* et dans l'expression *tout ce que*.

Règle 4.3 Ne pas étiqueter `PROREL` si *que* est le premier mot de la phrase, s'il suit un verbe directement, ou s'il est séparé du verbe uniquement par un commentaire entouré par des virgules. Pour les besoins de cette étude, la notion de commentaire est définie simplement comme une suite de tokens de longueur courte, entourée par des virgules, précédée directement par un verbe, et suivie directement par le mot *que*. Bien que cette définition ne soit pas linguistiquement rigoureuse, elle corrige quelques erreurs dans le corpus `dev`, et n'en introduit aucune.

Règle 4.4 Ne pas étiqueter `ADV` sauf s'il y a un *ne* plus tôt dans la phrase, ou si *que* est le premier mot de la phrase.

	ADV	**CS**	**PROREL**	**PROWH**	**Total**	**Erreurs**
ADV	134 (+1)	5 (–1)	0	0	139	5 (–1)
CS	10	1 149 (+14)	36 (–14)	0	1 195	46 (–14)
PROREL	0	48 (–4)	265 (+4)	0	313	48 (–4)
PROWH	0	0	2 (–2)	27 (+2)	29	2 (–2)

Tableau 3. *Matrice de confusion pour* que *avec les règles*

Le tableau 3 montre les résultats après l'ajout des règles pour tous les corpus `dev` et `test`, avec, entre parenthèses, les gains par rapport au modèle des descripteurs riches. Sans surprise, les résultats sont positifs, puisqu'on a visé des erreurs trouvées dans ces mêmes corpus `dev`. Il nous reste 101 sur 222 erreurs, soit une réduction de 55 % du taux d'erreur, avec une exactitude de 93,97 %. Au niveau de la significativité, nous avons 21 nouvelles corrections pour 0 nouvelles erreurs. Pourtant, il y a un risque de suradéquation des règles aux corpus évalués, les corpus `test` étant trop petits pour mesurer l'impact plus global des règles. De ce fait, nous avons testé les mêmes règles sur des corpus non annotés, comparant les différences entre les analyses avec et sans règles. Nous avons donc analysé, avec et sans règles, 200 000 mots de chacun des

corpus *Est Républicain*[5], Leximedia[6], Frantext[7] et Revues.org. Au total, il y a uniquement une différence tous les 8 500 mots, mais avec un bilan très positif : 46 corrections pour 5 erreurs dans les 51 premières différences. Les corrections les plus intéressantes concernent les commentaires qui séparent le *que* du verbe qui le gouverne. La règle corrige l'étiquette `PROREL` en `CS`, comme dans la phrase suivante.

Exemple 4.7 *Je **conteste**, en tant que père de famille, **que**/CS l'on vienne me dire que l'argent est le corollaire du succès.*

5. Analyse syntaxique des structures coordonnées

Dans cette deuxième étude de cas, nous nous tournons vers les structures coordonnées (SC), qui coordonnent plusieurs éléments conjoints (appelés simplement « conjoints » ci-dessous), et qui représentent une source d'erreur reconnue pour les analyseurs syntaxiques automatiques. Elles posent un défi particulièrement difficile pour les parseurs par transitions, qui analysent la phrase de façon séquentielle de gauche à droite : en effet, même dans le cas d'une SC simple, il est pratiquement impossible d'identifier le premier conjoint sans analyser la suite de la phrase. Considérez les trois phrases suivantes, identiques jusqu'à la conjonction de coordination.

Exemple 5.1 – Je mange une pomme rouge et mûre.
– Je mange une pomme rouge et une orange.
– Je mange une pomme rouge et Georges boit du thé.

Dans les phrases ci-dessus, il est possible d'identifier le premier conjoint par un examen superficiel des étiquettes morphosyntaxiques immédiatement après la conjonction, sauf dans la dernière phrase, où il faut décider si Georges sera mangé ou non. Néanmoins, aucun élément avant la conjonction ne peut nous aider à prendre la décision. Souvent la situation est bien plus complexe, avec l'insertion de circonstants entre la conjonction et le deuxième conjoint, sans parler des différentes formes elliptiques, des SC avec trois conjoints et plus, ou des modifieurs partagés par plusieurs conjoints. Nous cherchons ici à améliorer l'analyse des SC par l'ajout de descripteurs riches, et par une transformation des normes d'annotation du corpus.

5.1. *Annotation et mécanismes d'analyse*

Considérons la phrase suivante, contenant une SC à trois conjoints : *« Je vois Jean, Paul et Marie. »* La figure 1 montre l'annotation en dépendances de cette phrase

5. http://www.cnrtl.fr/corpus/estrepublicain/
6. http://redac.univ-tlse2.fr/applications/leximedia2007.html
7. http://www.frantext.fr

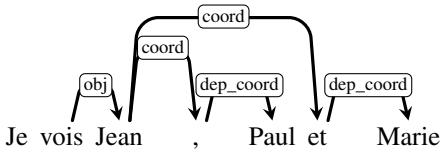


Figure 1. *L'annotation de la coordination dans le SPMRL français*

selon la norme du SPRML français : tous les conjoints sont gouvernés par le premier conjoint, *via* la virgule ou la conjonction précédente. Quelles sont alors les étapes pour effectuer l'analyse syntaxique de cette phrase dans le *parsing* par transitions ? Selon Kübler *et al.* (2009), cet algorithme utilise la définition suivante d'une **configuration de *parsing*** :

– σ : une **pile** (*stack*) – une séquence de tokens qui ont été partiellement traités ;

– β : un **buffer** – une séquence de tokens qui n'ont pas encore été traités ;

– Δ : un ensemble d'arcs de dépendance de la forme *étiquette(gouverneur, dépendant)* déjà ajoutés ;

– τ : une séquence de transitions permettant d'atteindre la configuration actuelle à partir de la configuration initiale.

Nous utilisons σ_0 pour indiquer le token actuellement en haut de la pile, et $\sigma_{1..n}$ pour les tokens plus en profondeur de la pile. De façon similaire, β_0 indique le prochain token à traiter dans le buffer, et $\beta_{1..n}$ les tokens plus loin dans le buffer. L'analyse commence avec un artefact *root* sur la pile, et tous les autres tokens dans le buffer. À chaque étape d'analyse, l'algorithme doit déterminer la transition à appliquer, en décidant s'il existe ou non une dépendance entre σ_0 et β_0. L'analyse se termine quand le buffer est vide. Notre étude utilise le système des transitions *arc-eager* (Nivre, 2008), qui définit les quatre transitions présentées dans le tableau 4 permettant de passer d'une configuration à l'autre.

Transition	**Effet**
`left-arc`$_{\text{étiquette}}$	Créer l'arc de dépendance *étiquette(β_0,σ_0)* et enlever le token en haut de la pile
`right-arc`$_{\text{étiquette}}$	Créer l'arc de dépendance *étiquette(σ_0,β_0)*, et mettre le token en tête du buffer en haut de la pile
`reduce`	Enlever le token en haut de la pile
`shift`	Mettre le token en tête du buffer en haut de la pile

Tableau 4. *Le système des transitions* arc-eager

Il est bien connu que les parseurs par transitions tendent à favoriser les dépendances à plus courte distance (McDonald et Nivre, 2007 ; Candito *et al.*, 2010), puisqu'ils comparent toujours deux tokens plus proches l'un de l'autre avant de com-

parer deux tokens plus éloignés, et la décision concernant les deux tokens les plus proches est prise indépendamment, ne faisant usage que de l'information disponible à ce moment-là. Ainsi, un token plus proche n'est jamais directement comparé à un token plus éloigné quand on prend la décision de rattachement. Cette tendance peut être atténuée en appliquant une recherche par faisceau (Urieli et Tanguy, 2013).

La figure 2 montre la séquence exacte de transitions nécessaires pour parser la troisième phrase de l'exemple 5.1, à partir du moment où la conjonction de coordination se trouve pour la première fois à β_0, jusqu'au moment ou la SC a été complètement analysée. Parmi ces transitions, les deux transitions `reduce` en début d'analyse sont particulièrement difficiles, car elles doivent chercher le deuxième conjoint le plus probable dans le buffer, étant donné qu'on ne peut pas réduire un token tant qu'il reste des dépendants à rattacher. La transition `shift` vers la fin de l'analyse est bien plus simple, puisqu'on sait déjà que le premier conjoint est un verbe. Les autres décisions sont assez triviales.

5.2. *Typologie des erreurs initiales*

Comme pour l'étiquetage morphosyntaxique, nous avons utilisé un modèle de SVM linéaire, mais cette fois avec les paramètres $C = 0,25$ et $\epsilon = 0,01$ et un *cutoff* de 5. Sauf pour la toute dernière expérience, nous utilisons les étiquettes morphosyntaxiques « *gold* » du Treebank, afin de faire abstraction des erreurs d'étiquetage et se concentrer sur l'analyse syntaxique. Le LAS de base en dehors de la ponctuation est de 89,57 % (`dev`) et 89,45 % (`test`). La f-mesure de base pour les SC, calculée au niveau des arcs de dépendance de la coordination, est de 84,35 % (`dev`) et 85,16 % (`test`).

Nous avons démarré cette étude en analysant les erreurs de coordination faites par Talismane dans le corpus `dev`. Sur 240 erreurs analysées, 24 % étaient des erreurs d'annotation dans le corpus (dont 60 % ont été correctement annotées par Talismane), 14 % étaient des artéfacts des normes d'annotation (Talismane a attaché le troisième conjoint directement au deuxième, au lieu de l'attacher au premier), et 30 % étaient des erreurs où Talismane a coordonné des tokens avec des étiquettes morphosyntaxiques différentes, alors qu'il aurait dû coordonner deux tokens ayant la même étiquette. Si l'on regroupe ces cas avec d'autres cas de parallélisme simple (*e.g.* des cas où Talismane a coordonné deux prépositions différentes au lieu de coordonner la même préposition), on arrive à 38 %. Les 24 % qui restent sont des cas difficiles, dont des structures elliptiques. Seuls 12 % des cas sont des erreurs entre des tokens ayant les mêmes étiquettes (ou les mêmes prépositions), pour lesquelles seulement une information sémantique pourrait nous aider à arbitrer.

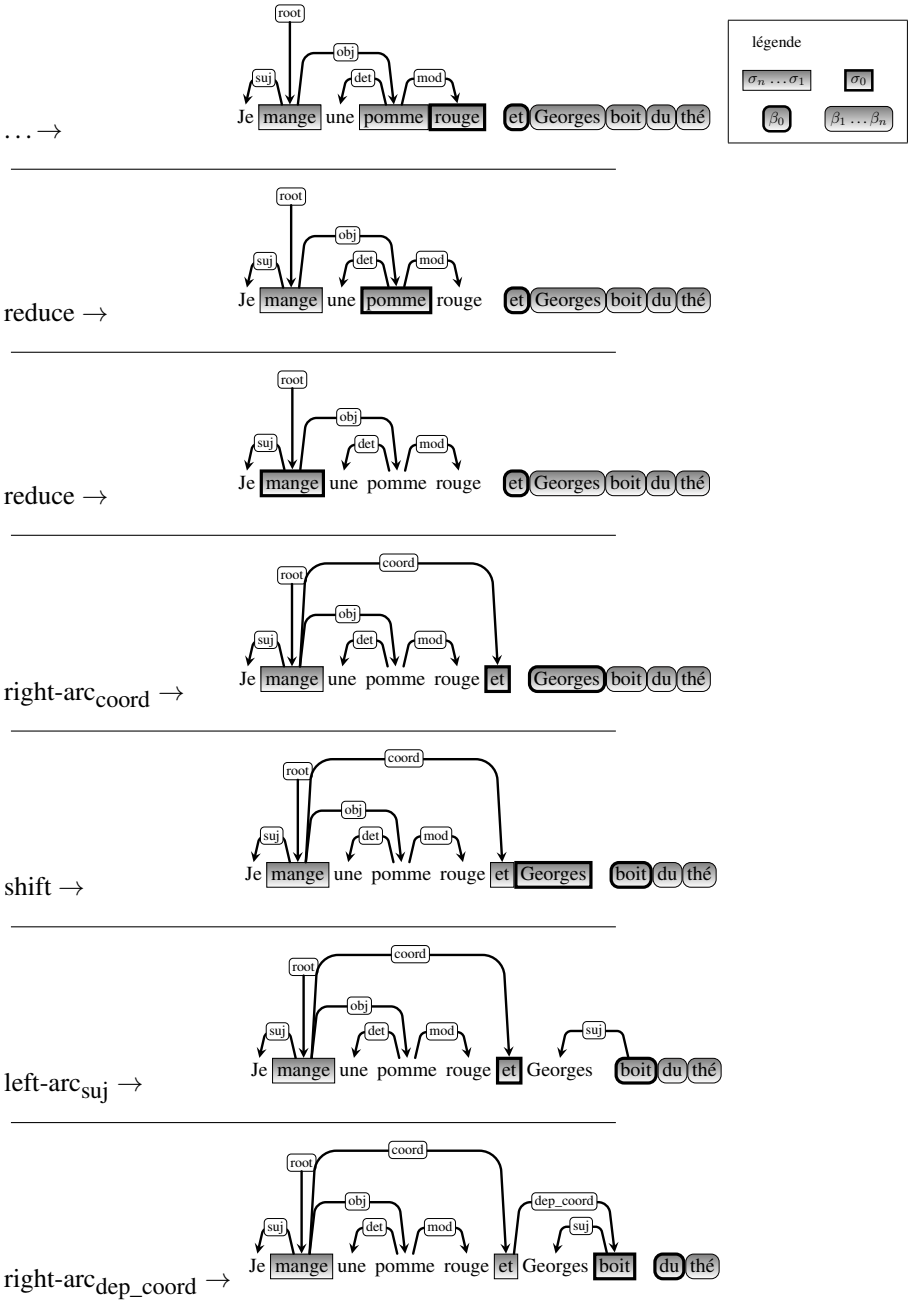


Figure 2. *Séquence des transitions* arc-eager *pour la coordination*

5.3. *Utiliser les descripteurs riches*

Dans le *parsing* par transitions, un descripteur s'applique à une configuration, c'est-à-dire qu'il puise des informations dans la pile, le buffer ou les arcs de dépendance déjà ajoutés, et doit aider le système à sélectionner la bonne transition parmi les transitions possibles. Pour notre première expérience sur la coordination, nous avons décidé de cibler les 38 % des erreurs concernant un parallèlisme simple – soit deux étiquettes morphosyntaxiques mal assorties, soit deux prépositions mal assorties – à l'aide de descripteurs riches.

5.3.1. *Travaux similaires*

Hogan (2007) a utilisé des descripteurs riches pour améliorer la coordination des syntagmes nominaux (SN) en anglais de façon significative, dans le contexte d'un parseur en constituants à base d'historique, en introduisant des descripteurs pour la similarité sémantique entre les têtes des SN, ce qui exige une ressource sémantique fiable. Plutôt que d'utiliser la similarité sémantique, Shimbo et Hara (2007) ajoutent des descripteurs qui donnent une similarité syntaxique entre deux structures à coordonner. Leur travail se rapproche du nôtre par l'utilisation des descripteurs riches qui visent spécifiquement le parallélisme des éléments conjoints. En revanche, ces deux études s'appliquent à des parseurs en constituants avec une complexité de base plus élevée que notre parseur par transitions à complexité linéaire, et leur descripteurs visent soit la sémantique, soit la structure syntaxique, alors que nous nous sommes intéressés au parallélisme simple (étiquettes ou prépositions mal assorties).

D'autres études ont tenté d'introduire des descripteurs plus riches mais génériques, sans viser précisément le parallélisme des structures coordonnées. Kübler *et al.* (2009) proposent une méthode où les n-meilleures analyses syntaxiques PCFG sont reclassées, afin d'améliorer l'analyse des SC en allemand. Leurs descripteurs sont génériques et peuvent s'appliquer à des arbres syntaxiques entiers puisqu'ils sont appliqués lors du reclassement post-traitement. Notre étude diffère de la leur en appliquant les descripteurs riches pendant l'analyse du parseur, plutôt que de nécessiter un grand faisceau des n-meilleures solutions.

Zhang et Nivre (2011 ; 2012) ont démontré l'utilité des descripteurs riches génériques tels que la valence (nombre de dépendants) d'un token, la distance entre deux tokens, etc. Ils démontrent que ces descripteurs sont surtout utiles dans le cas des parseurs à apprentissage global (*global learning*) avec un faisceau très large (64). Ceci s'accompagne malheureusement de certains inconvénients pratiques car la vitesse d'analyse est corrélée linéairement à la largeur du faisceau. Ici nous n'étudions pas des largeurs de faisceau au-delà de deux et n'appliquons pas d'apprentissage global, mais démontrons néanmoins que des descripteurs ciblés très spécifiques peuvent apporter un gain considérable.

De la Clergerie (2014) introduit des descripteurs riches symboliques dans un parseur par transitions indirectement, en analysant chaque phrase d'abord par FRMG, un parseur TAG, et en injectant des descripteurs fondés sur l'analyse FRMG dans le par-

seur par transitions. Il atteint d'excellents résultats pour le français (LAS = 90,25 pour le corpus SPMRL-`test` étiqueté automatiquement). En revanche, la mise en place du système est compliquée et le système à deux parseurs n'est pas directement comparable à un parseur par transitions à complexité linéaire.

5.3.2. *Définition et application des descripteurs*

Dans la définition des descripteurs riches que nous utilisons dans cette étude, vu l'importance d'identifier le deuxième conjoint avant de pouvoir identifier le premier, nous avons d'abord créé le descripteur suivant.

Descripteur 5.1 Identification du deuxième conjoint. Ce descripteur étant un composant de tous les descripteurs ci-dessous pour identifier le premier conjoint, il doit deviner le deuxième conjoint sans connaître le premier. Il ne peut utiliser que les informations disponibles au moment où le candidat pour le premier conjoint est à σ_0 et la conjonction à β_0 (étapes 1, 2 et 3 dans la figure 2). Il a été construit par essais et erreurs comme une suite de règles superficielles *ad hoc*, jusqu'à ce qu'on atteigne une exactitude d'environ 97 %.

Les cas les plus difficiles pour ce descripteur sont les verbes, car il est difficile de distinguer entre les coordinations verbales et les incises, particulièrement nombreuses dans les textes journalistiques, et marquées (en dehors de la sémantique) uniquement par la ponctuation, l'ordre des mots, et les choix lexicaux. Les deux exemples ci-dessous (verbes en italique, conjoints soulignés) illustrent ces deux cas.

Exemple 5.2 Coordination verbale. Il s'*agit* ici d'un jour normal **et** un inventaire scrupuleux *exigerait* que l'on prenne en compte l'offre accrue du mercredi.

Exemple 5.3 Incise. À Lourdes, nous *signale* notre correspondant Jean-Jacques Rollat, la venue **et** la circulation des pèlerins *ont été* très *perturbées*.

Par la suite, nous avons utilisé ce descripteur pour construire divers descripteurs qui cherchent à reconnaître du parallélisme dans les SC dans le contexte du *parsing* par transitions. La plupart des descripteurs comparent le token actuellement à σ_0 au deuxième conjoint deviné par le descripteur 5.1, et contrôlent qu'il n'existe pas un meilleur candidat plus loin dans la pile. Dans les descripteurs ci-dessous, les tokens à σ_0 et β_0 sont soulignés, et les tokens comparés sont en gras.

Descripteur 5.2 Étiquettes mal assorties : N **ADJ** et **N**. Si le candidat pour le premier conjoint à σ_0 n'a pas la même étiquette morphosyntaxique que l'étiquette λ du deuxième conjoint identifié, existe-t-il un candidat plus loin dans la pile avec l'étiquette λ ?

Descripteur 5.3 Prépositions mal assorties : de N **à** N et **de** N. Si le candidat pour le premier conjoint à σ_0 n'est pas la même préposition que celle identifiée comme deuxième conjoint, cette dernière préposition existe-t-elle plus loin dans la pile ?

Descripteur 5.4 Étiquettes bien assorties : V **N** et **N**. Si le candidat pour le premier conjoint à σ_0 a la même étiquette morphosyntaxique que le deuxième conjoint identifié, y a-t-il d'autres candidats avec cette étiquette plus loin dans la pile ?

Descripteur 5.5 Parallélisme à trois conjoints : **N**, **N** et **N**. Quand deux tokens avec la même étiquette se trouvent à σ_0 et à β_0 séparés par une virgule, le second token est-il suivi d'une conjonction de coordination dont le dernier conjoint identifié a la même étiquette que les deux premiers ?

Descripteur 5.6 Parenthèses : mot_1 (mot_2) **et** mot_3. Le candidat pour le premier conjoint à σ_0 se trouve-t-il à l'intérieur des parenthèses qui ne contiennent pas la conjonction ?

Quand nous avons appliqué ces descripteurs aux corpus `test`, la f-mesure de la coordination (`coord` et `dep_coord` combinés) est passée de 85,16 % à 86,97 %, donc avec une réduction du taux d'erreur de 12,20 %. En termes de significativité, le test de McNemar donne une p-valeur $< 0,001$ pour les arcs de coordination. Il existe bien sûr des cas dans le corpus d'entraînement avec des structures non parallèles valides, telles que : *« Au mieux, la reprise sera lente/ADJ **et** de/P faible ampleur. »* Néanmoins, celles-ci sont très peu nombreuses en comparaison du grand nombre d'erreurs pour les étiquettes mal assorties. Nous examinerons quelques erreurs introduites au niveau des SC non parallèles par l'ajout de ces descripteurs dans le paragraphe 5.5.

5.3.3. *Facilitation de la correction manuelle des corpus*

Afin de définir les descripteurs ci-dessus, plusieurs itérations ont été nécessaires, pendant lesquelles les descripteurs ont été projetés sur le corpus d'entraînement et tous les résultats inattendus ont été analysés. Parmi ceux-ci se trouvaient un très grand nombre d'erreurs d'annotation. Étant donné que 24 % des erreurs initiales dans le corpus `dev` concernaient des erreurs d'annotation, et ayant sous la main une méthode très efficace pour identifier et corriger ces erreurs par la projection de descripteurs ciblés, nous avons décidé d'appliquer ces corrections manuelles au corpus SPMRL français entier (`train`, `dev` et `test`). Notez que les erreurs d'annotation peuvent provenir en partie de la conversion automatique du corpus FTB en dépendances, mais suite à notre examen manuel des erreurs, celles-ci ne semblaient couvrir qu'une minorité des cas.

En tout, nous avons appliqué 1 313 corrections à `train` (sur 21 285 arcs de coordination = 6,2 %), 88 pour `dev` (sur 1 758 arcs de coordination = 5,0 %) et 183 pour `test` (sur 3 468 arcs de coordination = 5,3 %). L'évaluation a montré que corriger les erreurs dans le corpus d'entraînement n'est utile que lorsque les erreurs équivalentes ont été corrigées dans le corpus d'évaluation. On est passé d'une f-mesure de coordination de 85,16 à 86,77 pour `test`. Le reste de cette étude utilise les corpus manuellement corrigés comme baseline. Bien que ceci ne soit pas satisfaisant en termes de comparaison avec d'autres études, nous avons été contraints de le faire, car la conversion automatique entre différentes normes d'annotation ne fonctionnait pas tant que l'an-

notation initiale n'était pas cohérente et propre. Afin de simplifier les comparaisons, nous avons généré un fichier des différences à appliquer au SPMRL d'origine[8].

5.4. *Normes d'annotation pour la coordination*

Comme nous l'avons vu dans le paragraphe 5.2, plus de 14 % des erreurs initiales étaient des artéfacts de la norme d'annotation pour une SC avec plus de deux conjoints, où Talismane attachait l'élément conjoint systématiquement au conjoint précédent, alors que la norme d'annotation du SPMRL l'attachait au premier conjoint. Nous avons décidé de tester quatre normes différentes d'annotation des SC.

5.4.1. *Travaux similaires*

Plusieurs études ont exploré les normes d'annotation de coordination en anglais. Malheureusement, le Penn Treebank (PTB) d'origine n'annote qu'une partie des SC simples de façon implicite, c'est-à-dire en les regroupant dans un seul syntagme. Maier *et al.* (2012) présentent une norme d'annotation pour l'anglais qui inclut la ponctuation, puis entraînent un classifieur (Maier et Kübler, 2013) pour différencier les virgules coordonnantes et non coordonnantes, atteignant une f-mesure de 89,22 pour les virgules coordonnantes. La plupart des phénomènes qu'ils veulent désambiguïser dans les corpus en constituants en annotant la ponctuation, sont désambiguïsés plus simplement dans les corpus en dépendances en utilisant un jeu plus précis d'étiquettes de dépendances, *e.g.* pour différencier l'apposition de la coordination. C'est l'approche de cette étude, qui montre l'intérêt de supprimer l'annotation de la ponctuation, afin de concentrer les décisions du système sur les conjoints plutôt que sur les coordonnants.

Ivanova *et al.* (2013) mesurent la performance de trois normes d'annotation différentes pour l'anglais, toutes couvertes dans cette étude. Concernant l'annotation, ils arrivent à des conclusions similaires pour l'anglais que celles que nous obtenons pour le français, mais ils mettent l'accent sur la norme des SC « gouvernée par la conjonction » qui pourtant produit les résultats les moins bons, du fait que le parseur grammatical dont ils sont spécialistes prône cette norme.

Popel *et al.* (2013) comparent une multitude de normes d'annotation pour la coordination et développent un outil pour passer d'une annotation à l'autre sans perte d'information. Ils décrivent en détail les diverses difficultés rencontrées lors de l'annotation de la coordination, y compris le rôle de la ponctuation.

Schwartz *et al.* (2012) comparent l'« apprenabilité » (*learnability*) de diverses normes d'annotation pour six structures syntaxiques du PTB, dont la coordination, où l'annotation la plus « apprenable » est celle qui donne l'exactitude la plus élevée ainsi que celle atteinte avec le nombre le plus réduit d'exemples d'entraînement. Ils comparent deux normes d'annotation pour la coordination, et trouvent, comme nous, qu'il est bien mieux d'utiliser le conjoint comme tête plutôt que la conjonction, pour

8. https://github.com/urieli/talismane/tree/master/examples/tal56-1/spmrl2013patches

tout type de parseur. Puisque le PTB n'annote pas les SC avec plus de deux conjoints, ils explorent moins de possibilités d'annotation que dans la présente étude.

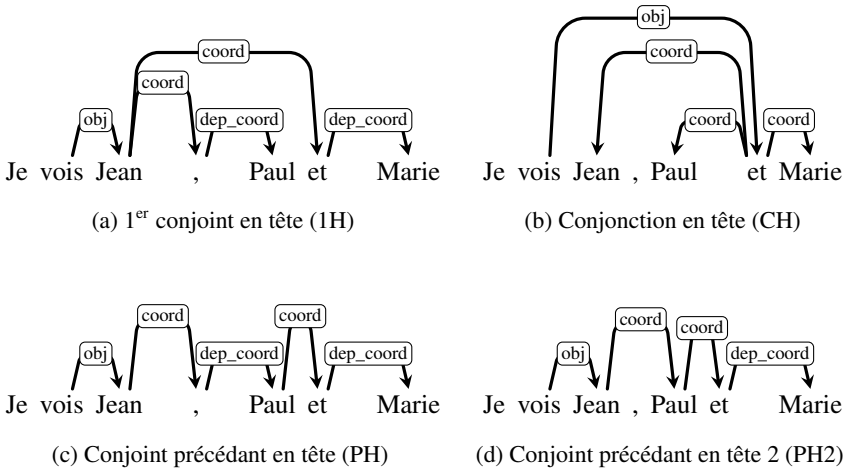
Notre étude étend les travaux précédents de la façon suivante : (a) elle applique des expériences similaires au français, et consolide certaines conclusions, tout en se concentrant sur les cas de trois conjoints ou plus ; (b) elle met en avant l'importance d'une annotation systématique de la ponctuation, uniquement possible quand l'annotation d'une SC ignore les virgules en reliant les conjoints directement entre eux ; et (c) elle compare les gains dus à un changement de norme d'annotation à ceux dus à des descripteurs riches ciblés ou à l'utilisation d'un faisceau plus large.

5.4.2. *Comparer les normes d'annotation*

Les quatre normes d'annotation des SC testées sont présentées dans la figure 3. La sous-figure 3a présente la norme **1H** (premier conjoint en tête) du corpus SPMRL. Le premier conjoint est la tête de la SC, qui gouverne les virgules coordonnantes et la conjonction *via* l'étiquette `coord`, qui, elles, gouvernent les autres conjoints *via* l'étiquette `dep_coord`. La sous-figure 3b présente la norme **CH** (conjonction en tête), prônée par beaucoup de grammaires : la conjonction gouverne tous les conjoints *via* l'étiquette `coord`. La sous-figure 3c présente la norme **PH** (conjoint précédent en tête), où chaque conjoint gouverne le coordonnant suivant (qu'il soit une virgule ou une conjonction) *via* l'étiquette `coord`, et le coordonnant gouverne le conjoint suivant *via* l'étiquette `dep_coord`. Enfin, la sous-figure 3d présente la norme **PH2**, identique à la norme PH sauf que la virgule est sautée (ou, plus précisement, rattachée au token précédant par l'étiquette `ponct`, ce qui l'exclut de l'annotation de la SC proprement dite), et que tout token coordonné par une virgule est gouverné directement par le conjoint précédent *via* l'étiquette `coord`. Dans le cas d'une simple SC avec deux conjoints, les normes PH, PH2 et 1H sont identiques.

Ces annotations ne sont pas pour autant équivalentes au niveau de l'information. Dans le cas des SC imbriquées (*e.g. « un film noir et blanc et pas trop nul »*), la portée de la coordination est ambiguë dans la norme 1H, mais pas dans les autres. Les modifieurs post-positionnés partagés (*e.g. « Jean, Paul et Marie Dupont »*, où les trois sont membres de la famille Dupont) ne sont clairement identifiés que par la norme CH, qui semble donc supérieure au niveau linguistique. Pour les autres normes, on peut indiquer un modifieur post-positionné partagé en l'annotant comme dépendant du premier conjoint. Cette convention devient non projective (avec des arcs de dépendance croisés) dès que le modifieur s'applique aux dépendants des conjoints, *e.g.* aux objets de la préposition dans *« Je parle de Jean, de Paul, et de Marie Dupont »*. Finalement, aucune des normes proposées ne fournit une solution pour les coordinations elliptiques, *e.g. « J'ai vu Jean et Paul hier, et Marie aujourd'hui »*. Malgré ces différences, dans la vaste majorité des cas rencontrés dans nos corpus, les quatre normes sont strictement équivalentes. Ainsi, notre étude se concentre sur les performances d'un analyseur automatique plutôt que sur la pertinence linguistique des diverses annotations.

Par ailleurs, le gouverneur de la ponctuation dans le FTB semble être choisi de façon arbitraire, sans motivation linguistique, en dehors des virgules coordonnantes.



(a) 1er conjoint en tête (1H)

(b) Conjonction en tête (CH)

(c) Conjoint précédant en tête (PH)

(d) Conjoint précédant en tête 2 (PH2)

Figure 3. *Normes d'annotation pour la coordination*

Nous avons donc testé une autre règle d'annotation, dans le but unique d'améliorer les performances automatiques du parseur : la ponctuation (à l'exception des virgules coordonnantes dans les normes 1H et PH) est systématiquement attachée au token précédent le plus proche. Dans les normes CH et PH2, cette règle est appliquée d'office afin d'éviter un grand nombre d'arcs non projectifs. Dans ces deux normes, où les virgules de coordination ne sont pas utilisées pour annoter les SC, cette option se traduit par une annotation homogène de toute la ponctuation du corpus, et donc une application systématique des transitions $\texttt{right-arc}_{\text{ponct}}$ et `reduce` chaque fois qu'on trouve un token de ponctuation à β_0. Cette technique permet au parseur de concentrer la question de la coordination autour d'une seule décision : y a-t-il ou non une relation entre deux conjoints potentiels ?

Nous avons fait deux hypothèses : d'abord, les parseurs par transitions devraient favoriser les normes d'annotation qui utilisent des attachements à plus courte distance, à savoir PH et PH2. Ensuite, l'annotation systématique des virgules (PH2) devrait améliorer les performances en enlevant une source inutile d'ambiguïté.

Le tableau 5 montre les résultats des six normes d'annotation (où +P indique qu'on a appliqué l'annotation systématique de la ponctuation) : 1H, 1H+P, CH+P, PH, PH+P, PH2+P. Nous présentons le LAS et UAS avec et sans prise en compte de la ponctuation. Sans surprise, dans les normes sans l'option +P, nous perdons systématiquement 2,5 % avec ponctuation, alors que dans les normes avec l'option +P, nous gagnons 1,5 %. Pour la coordination, on considère les étiquettes `coord` et `dep_coord`, car la proportion de chaque étiquette est différente selon la norme utilisée. La précision est très élevée grâce à la présence de conjonctions. Le rappel est bien plus bas, vu la difficulté d'identifier le premier conjoint. La norme CH+P, exigeant beaucoup plus de

Annotation	1H	1H+P	CH+P	PH	PH+P	PH2+P
Coord F1	86,77	87,06	74,19	88,30	88,38	90,17
Coord préc.	99,70	99,41	99,54	99,75	99,32	99,63
Coord rappel	76,81	77,44	59,13	79,21	79,62	82,35
LAS hors ponct.	89,79	89,88	87,34	89,91	90,02	90,19
UAS hors ponct.	91,77	91,87	89,32	91,88	92,02	92,16
LAS	87,36	91,18	89,06	87,40	91,30	91,52
UAS	89,07	92,92	90,77	89,11	93,06	93,23

Tableau 5. *Comparaison des normes d'annotation (corpus* `test`*)*

descripteurs anticipatifs, est de loin la perdante. Les normes où le conjoint est gouverné par le conjoint précédent (PH, PH+P, PH2+P) donnent de meilleurs résultats que celles où il est gouverné par le premier conjoint (1H, 1H+P) : 1,5 % de plus pour la f-mesure de la coordination, ce qui valide notre première hypothèse. Finalement, la norme PH2+P, où toute l'ambiguïté d'attachement est transposée de la ponctuation aux conjoints, donne les meilleurs résultats, avec un gain de 2 % par rapport à la norme PH+P, ce qui valide notre deuxième hypothèse. La réduction du taux d'erreur dans la f-mesure de la coordination entre la norme originale 1H et PH2+P est de 25,70 %. En termes de significativité (test de McNemar appliqué aux conjoints individuels), les différences entre 1H, 1H+P, PH et PH+P ne sont pas significatives (p-valeur $>$ 0,05), alors que les différences entre CH+P ou PH2+P et toutes les autres normes sont très significatives (p-valeur $<$ 0,001).

5.5. *Combiner les normes d'annotation et les descripteurs ciblés*

Notre dernière expérience vise à combiner la norme PH2+P avec les descripteurs ciblés présentés dans le paragraphe 5.3.2, afin de voir si les gains sont cumulatifs. Nous avons aussi testé des largeurs de faisceau de 1 et 2, afin de mesurer le gain supplémentaire possible avec un faisceau plus large, où un faisceau permet de garder les *k* solutions les plus probables localement à chaque étape d'analyse, pour éviter de prendre des décisions définitives trop tôt, comme décrit dans Urieli et Tanguy (2013).

Le tableau 6 montre les résultats avec des faisceaux de largeur 1 et 2 pour la norme originale 1H et pour la meilleure norme PH2+P, avec (+) ou sans (∅) les descripteurs ciblés. Les gains sont clairement centrés sur le rappel de la coordination. Le tableau 7 montre la même information en termes de réduction du taux d'erreur de la f-mesure de la coordination, par rapport à la configuration de base (norme 1H, descripteurs de base, faisceau 1), avec une réduction maximale de 37,04 %. Les trois paramètres testés sont assez cumulatifs au niveau du gain. Changer la norme d'annotation permet le gain le plus important, suivi par l'ajout de descripteurs ciblés et finalement par la largeur du faisceau. Toutes les combinaisons sont significatives (p-valeur $<$ 0,05, test de McNemar appliqué aux conjoints individuels).

Faisceau	**Faisceau 1**				**Faisceau 2**			
Annotation	**1H**		**PH2+P**		**1H**		**PH2+P**	
Descripteurs	∅	+	∅	+	∅	+	∅	+
Coord F1	86,8	88,7	90,2	91,3	88,0	89,1	90,7	91,7
Coord préc.	99,7	99,6	99,6	99,6	99,7	99,6	99,6	99,6
Coord rappel	76,8	80,0	82,4	84,3	78,8	80,7	83,2	84,9
LAS hors ponct.	89,8	90,0	90,2	90,3	90,2	90,5	90,4	90,6
UAS hors ponct.	91,8	92,0	92,2	92,3	92,2	92,5	92,4	92,5
LAS	87,4	87,6	91,5	91,6	87,9	88,2	91,7	91,8
UAS	89,1	89,3	93,2	93,3	89,6	89,9	93,4	93,5

Tableau 6. *Combinaison annotation / descripteurs / faisceau – corpus* `test`

	Aucun	**Descripteurs**	**Annotation**	**Les deux**
Faisceau 1	0,00	14,74	25,70	34,32
Faisceau 2	9,45	17,84	29,55	37,04

Tableau 7. *Réduction du taux d'erreur par rapport à la f-mesure de base pour la coordination (86,77) avec 1H, descripteurs de base et faisceau 1 – corpus* `test`

En termes de vitesse d'analyse, ces paramètres ont un coût très varié. La configuration de base prend 171 secondes pour parser le corpus `test` auxquelles s'ajoutent 133 s pour le chargement du modèle et du lexique, ou environ 400 tokens/seconde. Changer la norme d'annotation rend l'analyse un peu plus rapide ($\times\ 0,93$). Changer la largeur du faisceau ralentit l'analyse de façon linéaire ($\times\ 2$ pour un faisceau de 2). Finalement, les descripteurs ciblés ralentissent l'analyse énormément ($\times\ 22$).

Il est difficile d'estimer l'effet des nouveaux descripteurs sur la complexité théorique d'analyse. Un analyseur par transitions ne garde sa complexité linéaire que si le calcul des descripteurs se fait en temps constant. Dans notre cas, les descripteurs riches sont enclenchés pour chaque conjonction de subordination, mais la recherche du deuxième conjoint est limité à k tokens, ce qui le rend théoriquement constant par rapport à la longueur n de la phrase. Néanmoins, au niveau empirique, nous notons une petite corrélation entre la longueur de la phrase et la durée d'analyse moyenne par token, même pour les descripteurs de base (prenant 1,9 fois plus de temps en moyenne par token pour les 50 % de phrases plus longues que la médiane), mais moins accentuée pour les descripteurs riches (1,5 fois plus de temps en moyenne par token).

Nous avons refait une analyse des erreurs du corpus `dev`, pour la norme PH2+P avec des descripteurs ciblés et un faisceau de largeur 1. Bien que le nombre d'erreurs de coordination ait diminué de 241 à 151, le pourcentage concernant des erreurs de parallélisme simple reste stable, diminuant de 38 % à 36 %. Les cas compliqués, en revanche, ont bien augmenté : les ellipses de 5 % à 13 %, et les cas où seule une analyse sémantique peut nous aider de 12 % à 23 %. Ces derniers résultats nous encouragent à

envisager des descripteurs fondés sur des ressources sémantiques. Il y a quelques cas de SC non parallèles, où les nouveaux descripteurs ont introduit des erreurs, comme dans les exemples ci-dessous (erreurs d'analyse en italique, vrais conjoints soulignés, conjonction en gras). On a trouvé deux cas d'étiquettes non parallèles coordonnées (les exemples 5.4 et 5.5). Les autres concernent des cas qui auraient dû être parallèles, mais qui sont mal annotés à cause d'erreurs d'orthographe ou d'étiquetage dans le corpus *gold*, comme dans l'exemple 5.6, où le deuxième *baisser* est à l'infinitif plutôt qu'au participe passé, ou l'exemple 5.7, avec une mauvaise étiquetage du mot *généraux*.

Exemple 5.4 [...] celle *d'/P* une part significative des programmes et des productions réalisées/VPP **ou** *en cours de/P* réalisation.

Exemple 5.5 Ce n'*est/V* pas forcément la plus économiquement souhaitable/ADJ, **mais** celle/PRO qui fera le moins de vagues, *entendait/V*-on [...]

Exemple 5.6 Quant au dollar, il a monté/V quand on croyait qu'il allait *baisser/VINF* [...] **et** *baisser/VINF* derechef quand le marché commençait à se convaincre...

Exemple 5.7 [...] à l'ensemble des *présidents/NC* des conseils régionaux/ADJ et *généraux/NC*.

	LAS hors ponct.	UAS hors ponct.	LAS	UAS
Baseline	86,1	89,1	83,9	86,5
Meilleure	88,5	90,9	90,1	92,1

Tableau 8. *Évaluation complète (étiquetage + analyse syntaxique), pour la configuration de base (1H, descripteurs de base, faisceau 1) et la meilleure configuration (PH2+P, descripteurs ciblés* que *et coordination, faisceau 2) – corpus* `test`

Finalement, le tableau 8 montre une évaluation plus réaliste qui combine les étapes d'étiquetage morphosyntaxique et d'analyse syntaxique, pour la configuration de base et la meilleure configuration. Bien que l'on travaille sur un corpus corrigé, et donc non directement comparable avec d'autres études du français, un LAS de 90,1 et un UAS de 92,1 sont parmi les meilleurs pour le français. Il serait intéressant maintenant d'évaluer d'autres systèmes sur le corpus SPMRL corrigé et transformé à la norme PH2+P, afin de pouvoir effectuer une comparaison fiable.

6. Conclusion et perspectives

Dans cette étude, nous avons tenté d'améliorer l'analyse de certains phénomènes particulièrement difficiles en français : l'étiquetage morphosyntaxique du mot *que* et l'analyse syntaxique des structures coordonnées, avec l'aide de l'analyseur Talismane,

outil qui permet la configuration de descripteurs riches et de règles d'analyse avec une syntaxe très expressive. Pour l'étiquetage de *que*, nous avons appliqué des descripteurs ciblés et des règles et atteint une réduction de 55 % du taux d'erreur. Pour les structures coordonnées, nous avons appliqué des descripteurs ciblés, des transformations de la norme d'annotation, et un faisceau plus large et atteint une réduction maximale de 37 % du taux d'erreur.

Au niveau de l'étiquetage, il reste maintenant à généraliser la méthode à d'autres mots fonctionnels ambigus, tels que *de* ou *soit*. De même, nous pourrons généraliser la méthode pour l'analyse syntaxique à d'autres phénomènes difficiles, tels que l'attachement des syntagmes prépositionnels, tout en travaillant sur la vitesse d'analyse des descripteurs. Il serait aussi intéressant d'ajouter des descripteurs qui comparent la similarité sémantique des différents candidats à la coordination.

Bien que les scores globaux du système soient très compétitifs (LAS de 90,1 et UAS de 92,1), l'amélioration obtenue dépend autant des normes d'annotation et du faisceau que des descripteurs riches. On pourrait se demander si le travail minutieux nécessaire pour développer ces descripteurs, et qui semble très difficile à automatiser, est justifié compte tenu du gain d'exactitude. La réponse dépend de la personne qui demande : si elle s'intéresse uniquement au score global du parseur, l'effort manuel semble trop important pour peu de gain. Si, en revanche, elle s'intéresse à l'amélioration des constructions spécifiques, ou bien à la typologie des erreurs d'annotation manuelle ou automatique, l'effort semble justifié, d'autant plus qu'elle dispose après d'un corpus amélioré et de descripteurs en grande partie réutilisables par d'autres parseurs.

7. Bibliographie

Abeillé A., Clément L., Toussenel F., « Building a treebank for French », *in* A. Abeillé (ed.), *Treebanks*, Kluwer, 2003.

Brill E., « Transformation-based error-driven learning and natural language processing : A case study in POS-tagging », *Computational linguistics*, vol. 21, n° 4, p. 543-565, 1995.

Candito M., Nivre J., Denis P., Anguiano E. H., « Benchmarking of statistical dependency parsers for French », *Proceedings of the 23rd International Conference on Computational Linguistics : Posters*, ACL, p. 108-116, 2010.

Candito M., Seddah D., « Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical », *TALN*, ATALA, 2012.

Crabbé B., Candito M., « Expériences d'analyses syntaxique statistique du français », *TALN*, ATALA, 2008.

De la Clergerie E. V., « Jouer avec des analyseurs syntaxiques », *TALN*, ATALA, Marseille, France, p. 67-78, 2014.

Denis P., Sagot B., « Coupling an annotated corpus and a lexicon for state-of-the-art POS tagging », *Language Resources and Evaluation*, vol. 46, n° 4, p. 721-736, 2012.

Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J., « LIBLINEAR : A library for large linear classification », *Journal of Machine Learning Research*, vol. 9, p. 1871-1874, 2008.

Hogan D., « Coordinate noun phrase disambiguation in a generative parsing model », *Annual Meeting - ACL*, vol. 45, p. 680, 2007.

Ivanova A., Oepen S., Øvrelid L., « Survey on parsing three dependency representations for English », *ACL*p. 31, 2013.

Jacques M.-P., « Que : la valse des étiquettes », *TALN*, ATALA, Dourdan, France, p. 133-142, 2005.

Kübler S., Maier W., Hinrichs E., Klett E., « Parsing coordinations », *EACL*, p. 406-414, 2009.

Kübler S., McDonald R., Nivre J., *Dependency parsing*, Morgan & Claypool Publishers, 2009.

Maier W., Hinrichs E., Kübler S., Krivanek J., « Annotating coordination in the Penn Treebank », *6th Linguistic Annotation Workshop*, ACL, p. 166-174, 2012.

Maier W., Kübler S., « Are All Commas Equal ? Detecting Coordination in the Penn Treebank », *TLT12*, p. 121, 2013.

McDonald R. T., Nivre J., « Characterizing the Errors of Data-Driven Dependency Parsing Models. », *EMNLP-CoNLL*, p. 122-131, 2007.

Nivre J., « Algorithms for deterministic incremental dependency parsing », *Computational Linguistics*, vol. 34, n° 4, p. 513-553, 2008.

Popel M., Marecek D., Štepánek J., Zeman D., Žabokrtskỳ Z., « Coordination structures in dependency treebanks », *Annual Meeting - ACL*, 2013.

Sagot B., « The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French », *LREC*, 2010.

Schwartz R., Abend O., Rappoport A., « Learnability-Based Syntactic Annotation Design. », *COLING*, p. 2405-2422, 2012.

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 Clérgerie E., « Overview of the SPMRL 2013 Shared Task : A Cross-Framework Evaluation of Parsing Morphologically Rich Languages », *SPMRL*, Seattle, WA, 2013.

Shimbo M., Hara K., « A Discriminative Learning Model for Coordinate Conjunctions. », *EMNLP-CoNLL*, p. 610-619, 2007.

Urieli A., Robust French syntax analysis : reconciling statistical methods and linguistic knowledge in the Talismane toolkit, PhD thesis, Université de Toulouse, 2013.

Urieli A., Tanguy L., « L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions », *TALN*, ATALA, Les Sables d'Olonne, France, p. 188-201, 2013.

Zhang Y., Nivre J., « Transition-based Dependency Parsing with Rich Non-local Features. », *ACL (Short Papers)*, p. 188-193, 2011.

Zhang Y., Nivre J., « Analyzing the Effect of Global Learning and Beam-Search on Transition-Based Dependency Parsing. », *COLING (Posters)*, p. 1391-1400, 2012.