Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITE DE TOULOUSE

Délivré par :

Université Toulouse-Jean Jaurès

Présentée et soutenue par :
Aleksandra MILETIC

le mercredi 20 juin 2018

Titre :

Un treebank pour le serbe : constitution et exploitations

Tome 1

École doctorale et discipline ou spécialité :
ED CLESCO : Sciences du langage

Unité de recherche :
CLLE (UMR 5263)

Directeurs de Thèse :
Cécile FABRE, Université Toulouse-Jean Jaurès

Dejan STOSIC, Université Toulouse-Jean Jaurès

Jury :

SYLVAIN KAHANE	Professeur, Université Paris Nanterre	Rapporteur
PAOLA MERLO	Professeure, Université de Genève	Rapporteure
MARIE CANDITO	Maître de conférences, Université Paris Diderot	Examinatrice
VERAN STANOJEVIĆ	Professeur, Université de Belgrade	Examinateur
CECILE FABRE	Professeure, Université Toulouse-Jean Jaurès	Directrice
DEJAN STOSIC	Maître de conférences, Université Toulouse-Jean Jaurès	Directeur

Remerciements

Je tiens à exprimer ici ma reconnaissance à mes directeurs de thèse Cécile Fabre et Dejan Stosic. Merci à Dejan pour cette collaboration qui dure depuis 2011, quand j'ai participé au projet ParCoLab pour la première fois. Merci à Cécile d'avoir accepté de m'encadrer sans me connaître et de découvrir le serbe et la Serbie. Mon travail doit beaucoup à leur rigueur scientifique, leur bienveillance et leur implication. Cette interaction m'a apporté plus que je ne saurais l'écrire ici.

Je remercie Sylvain Kahane et Paola Merlo d'avoir accepté d'être les rapporteurs de cette thèse, ainsi que Marie Candito et Veran Stanojević d'avoir accepté d'être membres du jury. Ce travail a beaucoup bénéficié de mes échanges avec Marie Candito dans le cadre de mon comité de suivi de thèse. Et si je me suis orientée vers la recherche, c'est en partie grâce à Veran Stanojević, qui a encadré mon premier travail en linguistique durant ma licence à l'Université de Belgrade.

Je tiens également à remercier quatre personnes avec lesquelles j'ai beaucoup collaboré dans le cadre de cette thèse. Merci à Assaf Urieli pour nos nombreuses interactions autour du parser Talismane. C'est grâce à son enthousiasme scientifique et à sa disponibilité personnelle que j'ai pu approfondir cet aspect de mon travail. Merci à Saša Marjanović, pour son aide avec l'organisation des campagnes d'annotation, mais aussi pour nos discussions sur divers sujets et pour son soutien moral. Merci à Juliette Thuilier d'avoir participé à mon comité de suivi de thèse et d'avoir toujours accepté de discuter avec moi sur des questions syntaxiques. Le chapitre sur la position de l'adjectif en serbe lui doit beaucoup. Merci également à Franck Sajous d'avoir partagé avec moi son expérience de travail sur le Wiktionnaire. Cela m'a beaucoup aidée dans la constitution des ressources lexicales.

Si le corpus ParCoTrain-Synt a pu être complété dans les délais impartis, c'est grâce à la compétence et à l'énergie des annotateurs qui ont travaillé sur son enrichissement : Igor Ilić, Milena Janjić, Marijana Kaličanin, Jovana Marković, Jovana Milovanović, Irena Stanković, Andrijana Stojanović, Dušica Terzić et Nataša Živanović. Hvala svima !

Le laboratoire CLLE et l'équipe ERSS ont fourni une ambiance parfaite pour le déroulement de ce travail. Je tiens à remercier mes deux axes de recherche CARTEL et S'caladis pour leur soutien tout au long de ces quatre années. Merci également aux membres perma-

nents avec qui j'ai collaboré sur des projets ou autour des cours : Myriam Bras, Lydia-Mai Ho-Dac, Josette Rebeyrolle et Ludovic Tanguy. Nos échanges et interactions ont beaucoup contribué à mon développement professionnel. Un merci spécial à Ludovic pour toutes nos discussions autour d'un café.

Merci aussi à tous les doctorants, et particulièrement à Maxime Warnier et à Karla Orihuela. Avec Maxime j'ai partagé le bureau, mais aussi la charge du représentant des doctorants et de nombreuses pauses déjeuner. Karla, merci pour ton énergie positive inépuisable, et merci de m'avoir souvent hébergée durant les derniers mois de ma thèse.

Sans certaines personnes, rien de tout cela ne serait arrivé. Merci à mes parents Slađana et Borivoj pour la confiance inébranlable qu'ils ont en moi. Vous êtes les racines de tous mes arbres. Merci à mon frère Filip de m'avoir accompagnée à travers mes questionnements linguistiques et d'avoir fourni sans broncher des jugements de grammaticalité à des heures improbables. Tu es ma personne exemplaire. Merci à mon mari Alain, pour sa patience, sa constance et sa compréhension. Tu es mon point d'équilibre asymptotiquement stable.

Votre soutien m'a portée jusqu'au bout. Merci.

Table des matières

Liste des principaux sigles

BCMS	Bosniaque-croate-monténégrin-serbe
BSNLP	*Balto-Slavic Natural Language Processing*
CRF	*Conditional Random Fields*
DLM	*Dependency Length Minimization*
FTB	*French Treebank*
FTBDep	*French Treebank* en dépendances
GA	groupe/syntagme adjectival
GN	groupe/syntagme nominal
GP	groupe/syntagme prépositionnel
GPSG	*Generalized Phrase Structure Grammar* (Gazdar et al., 1985)
GRACE	*Grammars and Resources for Analysers of Corpora and their Evaluation* (Adda et al., 1995)
HMM	*Hidden Markov Models*
HOBS	*Hrvatska Ovisnosna Banka Stabala* (Treebank croate)
HPSG	*Head-Driven Phrase Structure Grammar* (Pollard & Sag, 1994)
LAS	*Labelled Attachment Score*, score de rattachement étiqueté
MEMM	*Maximum Entropy Markov Models*
MTT	*Meaning-Text Theory* (Mel'čuk, 1988)
OSV	Objet-sujet-verbe
OVS	Objet-verbe-sujet
PDT	*Prague Dependency Treebank*
POS tagging	*Part-of-Speech Tagging*

RSS	Relation de syntaxe de surface (Mel'čuk, 1988)
SDT	*Slovene Dependency Treebank*
SOV	Sujet-objet-verbe
SPMRL	*Syntactical Parsing of Morphologically Rich Languages*
SVM	*Support Vector Machines*
SVO	Sujet-verbe-objet
TAL	Traitement automatique du langage
TST	Théorie Sens-Texte ; traduction française du terme *Meaning-Text Theory*
UAS	*Unlabelled Attachement Score*, score du rattachement non étiqueté
UD	*Universal Dependencies Project*
VOS	Verbe-objet-sujet

Introduction

Au début de cette thèse, aucun corpus annoté syntaxiquement (treebank) n'était disponible pour le serbe. Or, les treebanks annotés manuellement sont une condition *sine qua non* du développement (entraînement et évaluation) d'outils statistiques dédiés à l'annotation syntaxique automatique (parsers). L'existence des parsers performants permet à son tour l'annotation syntaxique de corpus plus larges, qui peuvent ensuite être exploités pour alimenter des recherches en linguistique théorique. De fait, l'absence de ces ressources pour le serbe freine le développement des recherches sur cette langue dans ces deux directions, et plus généralement les efforts visant l'informatisation et la valorisation du serbe. Notre objectif est de combler cette lacune.

Dans ce travail, nous avons cherché à mettre en oeuvre des solutions équilibrées entre plusieurs exigences. Tout d'abord, entre la vitesse d'annotation manuelle et sa qualité. En effet, la constitution d'un treebank est un processus complexe et exigeant, qui fait typiquement l'objet de projets pluriannuels, habituellement confiés à des équipes scientifiques entières. Dans notre cas, la durée maximale du projet était déterminée par la durée d'une thèse, et l'équipe qui y travaillait activement était le plus souvent réduite à l'auteure de ce travail. Pour respecter les délais impartis, il fallait donc mettre en place des moyens d'accélérer l'annotation du corpus, sans pour autant en compromettre la qualité.

Deuxièmement, dans le cadre de la constitution des corpus annotés, le TAL et la linguistique théorique sont souvent vus comme deux domaines dont les objectifs ne sont pas toujours faciles à concilier. Pour qu'un corpus soit adapté aux recherches descriptives, il doit idéalement être doté d'annotations fines et détaillées. En revanche, dans le cadre du TAL, un autre impératif prévaut : il faut garantir dans le corpus des conditions propices à l'apprentissage des outils statistiques. Cela se traduit souvent par le besoin de limiter la granularité des informations encodées : si l'annotation utilisée est trop détaillée par rapport à la taille du corpus, différents types d'informations apparaîtront trop peu de fois pour qu'un outil automatique les maîtrise. Si l'on cherche à créer une ressource utile dans cette double perspective, comme c'est le cas dans cette thèse, la définition des annotations à apporter devient une question délicate à négocier.

Enfin, en créant un treebank du serbe, nous modélisons son fonctionnement syntaxique.

Il était donc logique de faire appel aux ouvrages existants en syntaxe théorique de cette langue pour identifier les structures et leur représentation à encoder dans nos données. Cependant, certains traitements traditionnels n'étaient pas adaptés à une implémentation en corpus : il s'agit notamment de ceux qui se basaient sur des critères sémantiques, insaisissables pour un parser. Pour ne pas compromettre l'utilité du corpus pour le TAL, il a donc été nécessaire d'abandonner certaines distinctions, pourtant reconnues et utilisées dans les travaux théoriques. Néanmoins, pour que ce corpus soit utilisé par la communauté des linguistes travaillant sur le serbe, il fallait s'assurer que les compromis faits soient acceptables.

Dans la suite de cette introduction, nous abordons tour à tour le contexte scientifique de cette thèse, ses objectifs et les principales propriétés de la méthode adoptée, pour préciser enfin son cadre pratique et l'organisation de ce manuscrit.

Positionnement scientifique

De par la nature de son sujet, cette thèse se situe à l'intersection de plusieurs domaines scientifiques. Tout d'abord, elle relève du TAL (traitement automatique du langage), aussi bien dans son versant consacré à la constitution de ressources que dans celui dédié à l'entraînement et à l'évaluation d'outils. Plus particulièrement, ce travail s'inscrit dans le paysage du TAL serbe. Ce contexte précis est défini par deux caractéristiques principales : premièrement, il existe relativement peu de ressources et d'outils librement disponibles, et deuxièmement, les performances des outils disponibles sont généralement en-dessous de l'état de l'art. À titre d'illustration, les seules ressources dédiées à cette langue qui sont librement diffusées comprennent un corpus adapté à l'étiquetage morphosyntaxique (Krstev et al., 2004b), un corpus issu du web doté d'annotations automatiques et par conséquent inadapté à l'entraînement des outils statistiques (Ljubešić & Klubička, 2014), un outil capable d'étiquetage morphosyntaxique et de lemmatisation (Gesmundo & Samardžić, 2012) et un lexique morphosyntaxique (Krstev et al., 2004b). Cependant, de nombreuses autres ressources sont citées dans les travaux existants sans être librement diffusées (cf. Vitas & Krstev, 2006 ; Krstev & Vitas, 2011, 2005 ; Jakovljević et al., 2014 ; Vitas & Krstev, 2004 ; Pavlović-Lažetić et al., 2004 ; Krstev, 2008). Plus concrètement, au niveau du parsing, la seule tentative d'entraînement d'un parser sur un corpus serbe a donné des résultats largement en dessous de l'état de l'art (cf. Jakovljević et al., 2014), dus le plus probablement à la taille très limitée du corpus utilisé. À notre connaissance, ni le corpus d'entraînement ni les modèles de parsing développés n'ont été diffusés. De ce fait, et contrairement à ce qui pourrait être déduit à partir d'une revue des travaux existants, le serbe reste une langue peu dotée dans le domaine du TAL.

Notre thèse rejoint donc deux cadres du TAL particuliers, mais fortement liés : celui du

traitement automatique des langues à morphologie flexionnelle riche et celui des langues peu ou sous-dotées. En effet, les langues à morphologie flexionnelle riche – dont le serbe fait partie – posent des défis spécifiques au TAL. Leurs systèmes flexionnels sont le plus souvent couplés à une flexibilité importante au niveau syntaxique. Cette richesse en formes fléchies, en propriétés morphosyntaxiques et en structures syntaxiques fait que ces langues sont plus difficiles à traiter que les langues comme l'anglais. Elles sont également victimes d'un paradoxe : pour obtenir une bonne couverture des différents phénomènes qu'elles exhibent, il est nécessaire de disposer de corpus plus larges que pour des langues à morphologie réduite. Or, la complexité de l'annotation a un effet rédhibitoire sur la constitution de ressources et elles sont souvent relativement mal dotées en corpus annotés.

Pour toutes ces raisons, ce type de langues suscite un intérêt particulier en TAL : des campagnes d'évaluation, des ateliers et des groupes d'intérêt leur sont consacrés. On peut citer SPMRL (Syntactical Parsing of Morphologically Rich Languages) [1], BSNLP (Balto-Slavic Natural Language Processing) [2], CCURL (Collaboration and Computing for Under-Resourced Languages) [3], SIGUL (Special Interest Group on Under-resourced Languages) [4], ou encore SIGSLAV (Special Interest Group on Slavic Natural Language Processing) [5]. Grâce à ces initiatives, la problématique du traitement de ces langues est désormais bien circonscrite, et des méthodes spécifiques pour l'aborder ont été identifiées, développées et évaluées (cf. Seddah et al., 2010 ; Candito & Seddah, 2010 ; Le Roux et al., 2012 ; Goldberg & Tsarfaty, 2008 ; Green et al., 2013 ; Fraser et al., 2013 ; Ling et al., 2015). Certaines d'entre elles ont été appliquées dans ce travail.

Cette thèse s'inscrit également dans la tradition de la création des treebanks. La constitution de ce type de corpus a débuté avec PennTreebank (Marcus et al., 1993), et les treebanks se multiplient depuis (cf. Skut et al., 1997 ; Hajič, 1998 ; Boguslavsky et al., 2002b ; Brants et al., 2002 ; Abeillé et al., 2003 ; Maamouri et al., 2004 ; Tonelli et al., 2008 ; Mille et al., 2013). Cette tendance continue aujourd'hui encore : un exemple en est le projet collaboratif Universal Dependencies, qui recueille plus de 100 treebanks différents [6]. L'expérience combinée de ces projets a permis de décrire différents aspects de la création de ce type de corpus. Notre travail puise certains de ses principes dans ces travaux, notamment en ce qui concerne la définition des annotations à apporter, l'organisation des campagnes d'annotation du corpus, les moyens de faciliter le travail manuel, etc.

Par le niveau d'analyse linguistique auquel il s'intéresse, ce travail relève également de la syntaxe théorique. Pour définir les principes de l'annotation syntaxique de notre corpus,

1. http://www.spmrl.org/
2. http://bsnlp-2017.cs.helsinki.fi/
3. http://www.ilc.cnr.it/ccurl2018/
4. http://www.elra.info/en/sig/sigul/
5. http://sigslav.cs.helsinki.fi/
6. http://universaldependencies.org/

nous faisons à la fois appel aux travaux en syntaxe du serbe (cf. Stanojčić & Popović, 2012 ; Ivić, 2005 ; Mrazović, 2009) et à des théories syntaxiques générales, notamment à la Théorie Sens-Texte (Mel'čuk, 1995) et aux travaux directement inspirés d'elle (cf. Iordanskaja & Mel'čuk, 2009 ; Burga et al., 2011).

Objectifs

Les objectifs de cette thèse sont multiples et se répartissent sur trois axes principaux.

Concernant la constitution des ressources du TAL, notre but est de doter le serbe des instruments nécessaires au parsing. Cela comprend la confection d'un treebank et de toute autre ressource auxiliaire qui faciliterait le traitement syntaxique de cette langue. Étant donné la relative pénurie d'outils de traitement automatique du serbe, nous envisageons également de nous servir des ressources créées pour entraîner des outils statistiques et obtenir ainsi des modèles de traitement réutilisables. Une importance particulière sera accordée à la diffusion des ressources constituées. Aussi, en outillant le serbe, nous espérons fournir un cadre propice aux recherches sur cette langue, aussi bien en TAL qu'en linguistique sur corpus.

Concernant la méthodologie de création de corpus, nous cherchons à identifier la démarche optimale pour entreprendre la confection d'un treebank pour une langue peu dotée. Notre objectif est de faciliter au maximum la tâche des annotateurs humains afin d'accélérer l'annotation du corpus tout en garantissant sa qualité. En le faisant, nous nous détachons du cadre concret de notre travail : au-delà de l'objectif de constitution d'un corpus et d'outils d'annotation précis, nous visons également la définition d'une méthode générale, applicable à d'autres langues, focalisée sur l'optimisation des ressources disponibles, aussi bien matérielles (corpus et outils) qu'humaines (processus d'annotation). Une telle méthode a le potentiel de faciliter la création de treebanks pour les langues qui n'en disposent pas. Dans le but de garantir la possibilité d'implémenter notre approche dans un nouveau contexte, nous fournissons toutes les informations nécessaires pour assurer une prise en main aisée des procédés décrits ici. Ce document peut donc également se lire comme un mode d'emploi pour la constitution d'un treebank d'une langue peu dotée.

Enfin, un troisième objectif de cette thèse porte sur l'exploitation des ressources constituées : nous souhaitons les mettre à l'épreuve et évaluer leur utilité dans différents types d'applications, en TAL comme en linguistique. Par cette démarche, nous cherchons à répondre à la question suivante : les mêmes types d'annotation peuvent-ils satisfaire les deux cadres applicatifs des corpus annotés ? Autrement dit, est-il possible d'avoir une annotation suffisamment détaillée pour permettre des recherches linguistiques fructueuses tout en assurant un terrain propice à l'entraînement des outils du TAL ?

Contexte pratique : projet ParCoLab

Avant de poursuivre, nous souhaitons présenter le contexte pratique dans lequel cette thèse s'est déroulée. En effet, notre travail a été effectué dans le cadre du projet ParCoLab (`http://parcolab.univ-tlse2.fr/`). Ce projet, lancé en 2010 par Dejan Stosic (équipe CLLE-ERSS, Université Toulouse - Jean Jaurès) a pour objectif la constitution d'un corpus parallèle serbe-français-anglais. Après les phases initiales de récolte, numérisation et parallélisation des textes, le corpus est consultable en ligne depuis 2015. À présent, il contient environ 11,1 millions de tokens provenant très majoritairement d'ouvrages littéraires, mais aussi de textes juridiques, de sous-titres de films et de contenu de sites institutionnels bilingues (cf. Miletic et al., 2017). Pour les trois langues, le corpus dispose d'un noyau de textes originaux alignés avec leurs traductions dans une ou deux langues (cf. tableau 1). Le corpus est parallélisé au niveau du paragraphe et de la phrase et il dispose d'un moteur de recherche permettant des requêtes combinant des expressions régulières et des opérateurs booléens. Il n'est en revanche pas doté d'annotations linguistiques.

Volet	Textes originaux	Traductions	Total
Serbe	1 100 137	2 686 547	3 786 684
Français	2 700 884	2 211 348	4 912 232
Anglais	979 027	1 451 346	2 430 373
Total			11 129 289

Table 1 – Distribution des tokens par volet de ParCoLab

La portée de cette thèse a été en partie déterminée par la volonté d'optimiser les conditions d'exploitation de ParCoLab. Du fait de sa nature parallèle, ce corpus a un grand intérêt aussi bien pour le TAL que pour la linguistique contrastive, la didactique des langues ou la traduction automatique. Cependant, ces exploitations potentielles sont conditionnées par la présence d'annotations linguistiques. Comme il existe déjà de nombreux outils d'annotation pour l'anglais et le français, le besoin de doter le serbe de ressources et outils d'annotation automatique a été identifié comme prioritaire. Ce fait a motivé la mise en place de cette thèse, l'idée étant que les résultats de ce travail pourront être exploités pour enrichir le volet serbe de ParCoLab d'informations linguistiques.

Ainsi, cette thèse a été soutenue financièrement par le projet ParCoLab. Plus particulièrement, les camapgnes d'annotation décrites dans ce document ont été financées à travers le Projet Campus France bilatéral franco-serbe PHC « Pavle Savic » dont ParCoLab a bénéficié en 2016-2017.

Nous précisons cependant que l'orientation scientifique de cette thèse n'a pas été déterminée par la nature parallèle du corpus : notre objectif était simplement d'identifier les traitements les mieux adaptés au serbe, de constituer des ressources d'une qualité aussi

élevée que possible et d'identifier la méthode optimale pour le faire. Nous visions ainsi la création de ressources qui pourront être réutilisées par la communauté scientifique serbe ou internationale.

Organisation du document

Ce manuscrit est organisé en deux tomes. Le premier décrit la démarche et les résultats scientifiques, alors que le deuxième contient les annexes. Le présent tome est divisé en trois parties. La partie I est consacrée à un état de l'art qui permet de définir le contexte scientifique de ce travail : nous illustrons ici les contraintes que le serbe pose dans le cadre du TAL et de la constitution des ressources (cf. chapitre 1) ; nous traçons ensuite les principes que nous adoptons par rapport aux différents aspects de la création des treebanks (cf. chapitre 2) ; enfin, nous présentons les outils sélectionnés pour réaliser nos objectifs (cf. chapitre 3).

Dans la partie II, nous abordons le travail pratique dans tous ses aspects. Nous détaillons d'abord les étapes mises en œuvre en amont de la constitution du treebank proprement dite : la mise en place de la méthode globale (cf. chapitre 4), l'élaboration des principes d'annotation (cf. chapitre 5), la constitution des ressources lexicales externes (cf. chapitre 6) et la mise en œuvre de la méthode adoptée (cf. chapitre 7). Ensuite, nous présentons les campagnes d'annotation et leurs résultats (cf. chapitre 8), ainsi qu'une première mise en pratique des ressources obtenues dans le cadre du parsing (cf. chapitre 9).

La partie III est dédiée à une évaluation de l'utilité de nos ressources au-delà du cadre global du parsing. Ceci est fait à travers deux études ciblées. La première d'entre elles, en syntaxe théorique, examine la position du groupe adjectival en serbe (cf. chapitre 10), alors que la deuxième aborde une question d'un double intérêt pour la syntaxe théorique et pour le TAL : les structures non projectives (cf. chapitre 11).

Enfin, pour clore ce document, nous proposons un bilan des résultats obtenus, des conclusions générales de ce travail et des perspectives pour sa continuation.

Première partie

État de l'art

Présentation de la partie I

Cette partie est consacrée à une considération approfondie de l'ensemble des principes théoriques et méthodologiques qui ont guidé notre travail. Dans le chapitre 1, nous examinons les contraintes et les exigences posées par le serbe en tant qu'une langue à morphologie flexionnelle riche, peu dotée en ressources et outils du TAL. Le chapitre 2 est dédié à une revue des principes et pratiques relatifs à la constitution des treebanks, à partir de laquelle nous posons les bases de notre méthode. Enfin, dans le chapitre 3, nous passons en revue certains outils disponibles pour le traitement automatique à différents niveaux, en accordant une attention spéciale aux résultats existants sur le serbe. Ainsi, nous identifions les outils dont nous nous servons dans la suite de ce travail.

Chapitre 1

Le serbe : une langue peu dotée ?

Comme il a été rapidement indiqué dans l'introduction, relativement peu de ressources adaptées au traitement automatique du serbe étaient librement disponibles au moment où ce travail de thèse a démarré. On peut attribuer cet état de faits à deux raisons principales. Tout d'abord, le serbe est une langue slave méridionale, à morphologie flexionnelle riche et à ordre de mots flexible. Du fait de leur diversité aux niveaux lexical, morphosyntaxique et syntaxique, ces langues posent des défis particuliers au TAL. En témoignent de nombreux travaux des journées d'étude SPMRL (*Statistical Parsing of Morphologically Rich Languages*), qui montrent les difficultés du traitement automatique de ces langues, qu'il s'agisse du parsing ou d'autres niveaux d'analyse et d'annotation.

Une deuxième raison du manque de ressources pour cette langue relève des pratiques de la communauté du TAL serbe. En effet, une partie importante des ressources auxquelles font référence les travaux publiés ne sont pas diffusées ou le sont sous des licences restrictives. Ceci a un impact négatif sur l'échange et la réutilisation des ressources et des outils, ainsi que sur la comparaison des résultats, défavorisant ainsi le développement du traitement automatique de cette langue.

La suite de ce chapitre est dédiée à l'analyse de la situation du serbe dans le cadre du TAL. Pour mieux cerner les défis liés au traitement automatique de cette langue, nous entamons ce chapitre par un rapide profil linguistique du serbe, en soulignant ses propriétés pertinentes pour le TAL (section 1.1). Nous dressons ensuite l'inventaire des outils et ressources disponibles pour le traitement automatique de cette langue (section 1.2) et nous cherchons à relier les observations de ces deux premières sections avec les spécificités du traitement automatique des langues à morphologie flexionnelle riche en général (section 1.3). Enfin, nous résumons les contraintes et les exigences identifiées au cours du chapitre (section 1.4).

1.1 Profil linguistique du serbe

Le serbe est une langue slave méridionale, parlée majoritairement en Serbie et dans les pays de l'ex-Yougoslavie par environ 8,7 millions de locuteurs (Keith, 2006). La langue est digraphique, utilisant l'alphabet latin et l'alphabet cyrillique de manière équivalente. Les deux systèmes d'écriture sont phonétiques, avec une correspondance parfaite entre les graphèmes et les phonèmes.

Du point de vue typologique, le serbe exhibe toutes les propriétés phares de la famille slave : il dispose d'un système de déclinaisons relativement complexe, l'ordre des constituants est flexible, il n'y a pas d'articles, le système de l'aspect verbal est particulièrement bien développé, et la réalisation du sujet dans la phrase n'est pas obligatoire (il s'agit d'une langue *pro-drop*). Ces caractéristiques – et notamment la richesse de la morphologie flexionnelle et la flexibilité syntaxique – soulèvent des questions particulières dans le cadre du traitement automatique. Afin de nous assurer que les lecteurs qui ne sont pas familiers avec les langues slaves puissent apprécier les enjeux du traitement automatique de ces langues, nous présentons leurs propriétés principales dans la suite de cette section, tout en soulignant leur impact sur le traitement automatique.

1.1.1 Morphologie flexionnelle riche

Si le système de la flexion verbale du serbe est comparable à celui du français et d'autres langues romanes, son domaine nominal est beaucoup plus riche en formes fléchies. Le serbe dispose d'un système de déclinaisons à sept cas (nominatif, génitif, datif, accusatif, vocatif, instrumental et locatif) et les noms portent également des marques du nombre (singulier ou pluriel). Par conséquent, on considère typiquement qu'un paradigme nominal contient 14 formes, même si on constate un degré de syncrétisme important (cf. tableau 1.1).

Cas	Singulier	Pluriel
Nominatif	grad	grado**vi**
Génitif	grad**a**	grado**va**
Datif	grad**u**	grado**vima**
Accusatif	grad	grado**ve**
Vocatif	grad**e**	grado**vi**
Instrumental	grad**om**	grado**vima**
Locatif	grad**u**	grado**vima**

TABLE 1.1 – Table de déclinaison du nom masculin *grad* 'ville'

Les cas ont un rôle important dans le fonctionnement syntaxique du serbe. À titre d'illustration, le sujet est exprimé au nominatif, l'objet direct prototypique à l'accusatif,

et l'objet indirect prototypique au datif. Toutes les prépositions imposent également des cas spécifiques à leur complément. Ce système permet un degré de liberté important dans l'organisation linéaire de la phrase (cf. section 1.1.2).

En plus du cas et du nombre, les adjectifs portent également des marques du genre, qui prend trois valeurs en serbe : masculin, féminin et neutre. Par ailleurs, des adjectifs peuvent également exprimer la définitude, ce qui se traduit par l'existence de deux paradigmes (défini et indéfini). Plus de détails sur ce point seront donnés dans la section 1.1.4. Pour le moment, retenons qu'un paradigme adjectival contient 84 formes, certaines d'entre elles pouvant être ambiguës. Comme on peut le voir dans le tableau 1.2, dans le cas des formes définies, le syncrétisme est particulièrement prononcé au pluriel, où les trois genres se distinguent seulement au nominatif et au vocatif.

	Cas	Masculin	Féminin	Neutre
Singulier	Nominatif	lep**i**	lep**a**	lep**o**
	Génitif	lep**og**	lep**e**	lep**og**
	Datif	lep**om**	lep**oj**	lep**om**
	Accusatif	lep**og**	lep**u**	lep**o**
	Vocatif	lep**i**	lep**a**	lep**o**
	Instrumental	lep**im**	lep**om**	lep**im**
	Locatif	lep**om**	lep**oj**	lep**om**
Pluriel	Nominatif	lep**i**	lep**e**	lep**a**
	Génitif	lep**ih**	lep**ih**	lep**ih**
	Datif	lep**im**	lep**im**	lep**im**
	Accusatif	lep**e**	lep**e**	lep**e**
	Vocatif	lep**i**	lep**e**	lep**a**
	Instrumental	lep**im**	lep**im**	lep**im**
	Locatif	lep**im**	lep**im**	lep**im**

TABLE 1.2 – Table de déclinaison des formes définies de l'adjectif *lep* 'beau'

Quant à la conjugaison, le serbe dispose des formes finies suivantes : présent (*ja radim* 'je fais'), parfait (*ja sam uradio* 'j'ai fait'), futur (*ja ću raditi* 'je ferai'), aoriste (*ja uradih* 'je fis'), imparfait (*ja radih* 'je faisais'), plus-que-parfait (*ja sam bio uradio/ja bejah uradio* 'j'avais fait'), futur antérieur (*ja budem radio* 'j'aurai fait'), conditionnel (*ja bih radio* 'je ferais') et impératif (*radi* 'fais'). Ces formes portent des marques de la personne (première, deuxième ou troisième) et du nombre (singulier ou pluriel). À titre d'illustration, le tableau 1.3 montre les formes fléchies du présent du verbe *raditi* 'faire, travailler'. Notons également qu'il s'agit d'une langue *pro-drop* qui n'exige pas la réalisation du sujet dans la phrase : la phrase serbe *Radim.* correspond à la phrase française *Je travaille.*

Parmi les formes non finies, on trouve l'infinitif, forme canonique du verbe serbe (*raditi*

Personne	Singulier	Pluriel
1ère	radi**m**	radi**mo**
2e	radi**š**	radi**te**
3e	radi	rad**e**

TABLE 1.3 – Formes du présent du verbe *raditi* 'faire'

'faire'), et quatre participes différents, dont deux sont invariables (*radeći* '(en) faisant' et *uradivši* 'ayant fait'), alors que les deux autres portent des marques du genre et du nombre (*radio* 'fait' à sens actif, et *rađen* 'fait' à sens passif). Les participes invariables sont typiquement utilisés dans des propositions participiales, alors que les participes variables sont surtout utilisés pour construire des formes verbales composées. Le subjonctif n'existe pas en serbe, et le passif connaît une utilisation moins répandue que, par exemple, en anglais ou en français. Si tous ces facteurs sont pris en compte, un verbe serbe peut avoir au-delà de 120 formes fléchies différentes.

Le serbe présente donc un nombre élevé de formes fléchies, ainsi que de nombreux traits morphosyntaxiques. Par ailleurs, nous avons également constaté qu'il existe un degré élevé d'ambiguïté dans les paradigmes flexionnels, et notamment dans celui des adjectifs. Le fait qu'une forme fléchie puisse correspondre à plusieurs interprétations morphosyntaxiques complexifie le traitement automatique. Ce type de langues posent donc des problèmes particuliers dans le cadre du TAL et exigent souvent la mise en place de méthodes spécialisées afin de pallier ces difficultés, ce qui sera discuté en détail dans la section 1.3.

1.1.2 Ordre de constituants flexible

Le fait que le système casuel prend en charge l'encodage des fonctions syntaxiques permet une grande variabilité dans la linéarisation de la phrase. Même si l'ordre des constituants canonique en serbe est SVO, les 5 autres variations sont grammaticales (cf. l'exemple 1, où le *sujet* est présenté en italiques et l'**objet direct** en gras).

(1) a. *Filip* predstavlja **Anu**
b. *Filip* **Anu** predstavlja
c. Predstavlja *Filip* **Anu**
d. Predstavlja **Anu** *Filip*
e. **Anu** predstavlja *Filip*
f. **Anu** *Filip* predstavlja
'*Filip* présente **Ana**'

Toutes ces configurations ne sont pas aussi fréquentes les unes que les autres, mais elles

sont toutes utilisées et servent à exprimer des focalisations différentes de la phrase (cf. Stanojčić & Popović, 2012, p. 367-368).

En outre, l'objet indirect prototypique dispose d'un degré de liberté comparable à celui du sujet et de l'objet direct. Une phrase relativement simple, contenant ces trois constituants de base, peut donc connaître de nombreuses variations. Dans l'exemple 2, nous donnons à titre d'illustration quelques configurations possibles, en indiquant l'objet indirect en souligné. Notons que les phrases proposées n'incluent pas de variations dans la position du sujet. Il est également important de remarquer que les trois fonctions syntaxiques illustrées sont portées par le même type d'élément – un groupe nominal. Par conséquent, les désinences casuelles sont le seul indice permettant la distinction de ces fonctions syntaxiques.

(2) a. *Filip* predstavlja **Anu** Alanu
b. *Filip* predstavlja Alanu **Anu**
c. *Filip* **Anu** predstavlja Alanu
d. *Filip* Alanu predstavlja **Anu**
'*Filip* présente **Ana** à Alain'

Un autre trait typique du serbe est la possibilité d'avoir des constituants discontinus. Dans l'exemple 3, nous voyons que l'adjectif *lepu* (accusatif singulier de la forme 'belle') est séparé des deux noms qui figurent dans la phrase. Pour déterminer lequel des deux est le gouverneur de cet adjectif, il est nécessaire de faire appel aux règles d'accord : ici, les valeurs des traits du genre, du nombre et du cas de l'adjectif correspondent à celles du nom *knjigu* 'livre', et non pas à celles de *Filip*, ce qui permet de dire que c'est la forme *knjigu* qui est le gouverneur de l'adjectif en question.

(3)

Lep-u	je	Filip	knjig-u	kupio.
beau-ACC.SG.F	est	Filip.NOM.SG	livre-ACC.SG.F	acheté

'C'est un beau livre que Filip a acheté.'

Les exemples parcourus illustrent le fait que les arbres syntaxiques en serbe disposent d'une flexibilité notable. Cela signifie qu'un parser dédié au traitement du serbe doit gérer un degré de variabilité plus important que dans des langues à ordre des mots plus rigide, comme l'anglais ou le français. On remarque également l'importance des traits morphosyntaxiques pour l'analyse syntaxique : la seule indication de la partie du discours et l'ordre des mots dans la phrase ne permettent pas d'identifier les fonctions syntaxiques de différents éléments. Dans le cadre du parsing, il est donc important de disposer de traits morphosyntaxiques plus fins pour faciliter la reconnaissance de différentes fonctions. Par ailleurs, nous avons observé que le serbe admet des structures syntaxiques discontinues.

Ce fait pose des contraintes à deux niveaux : tout d'abord, l'annotation syntaxique de cette langue doit être faite dans un cadre qui permet la représentation de ce type de structures, et deuxièmement, le parsing du serbe exige des outils capables de les traiter. Nous reviendrons plus en détail sur ces questions respectivement dans les chapitres 2 et 3.

1.1.3 Question de déterminants

Traditionnellement, on considère que la catégorie des déterminants n'existe pas en serbe (cf. Stanojčić & Popović, 2012 ; Ivić, 2005). Effectivement, cette langue ne dispose pas d'articles : les phrases *Filip achète un pull* et *Filip achète le pull* se traduisent toutes les deux par *Filip kupuje džemper* (lit. 'Filip achète pull'). Elle dispose néanmoins des formes suivantes : ***moj*** *primer* '**mon** exemple', ***taj*** *primer* '**cet** exemple', ***neki*** *primer* '**un (certain)** exemple' et ***kakav*** *primer* '**quel** exemple'. On remarque facilement la correspondance avec les classes des déterminants en français ou en anglais. Or, la tradition grammaticale serbe traite ces formes comme une sous-classe des pronoms, dits *pridevske zamenice* 'pronoms adjectivaux'.

En effet, la majorité des formes citées ci-dessus connaissent deux types de comportement syntaxique : elles peuvent apparaître antéposées à un nom, comme dans les exemples donnés, ou bien fonctionner de manière indépendante, comme dans la phrase *Neću* ***taj,*** *hoću* ***moj*** 'Je ne veux pas celui-ci, je veux le mien'. Dans ce cas, elles se comportent effectivement en tant que pronoms : elles occupent la place d'un groupe nominal dans la phrase. En revanche, dans le premier cas de figure, les formes en question se trouvent à l'intérieur d'un groupe nominal et prennent la place typique d'un adjectif (à gauche du nom). Dans le cas de la présence d'un adjectif qualificatif, ces formes se positionnent en tête du groupe (cf. ***taj/neki*** *plavi džemper* '**ce/un certain** pull bleu').

Malgré ces propriétés qui rappellent les déterminants dans d'autres langues, le seul travail théorique qui reconnaît ces formes comme appartenant à cette catégorie est celui de Mrazović (2009), alors que des travaux existants en traitement automatique du serbe suivent l'approche traditionnelle et leur accordent le statut de pronoms (cf. Krstev et al., 2004b). Ce traitement mérite cependant d'être questionné : comme ces formes connaissent un mode de comportement fondamentalement différent de celui des autres pronoms, le fait de leur accorder le même label peut introduire de la confusion dans l'apprentissage des outils automatiques. Le traitement de ces formes constitue donc une question qui doit être résolue lors de la définition des principes d'annotation morphosyntaxique pour cette langue.

1.1.4 Marquage de la définitude sur les adjectifs

Si le serbe ne dispose pas à proprement parler d'articles, comme nous venons de le voir, des résidus d'un système de marquage de la définitude existent. Il s'agit du trait morpho-syntaxique de la définitude des adjectifs, nommé en serbe *pridevski vid* 'aspect adjectival'. La définitude des adjectifs s'exprime par l'existence de deux paradigmes parallèles, l'un contenant des formes définies, et l'autre des formes indéfinies. Grâce à ce fait, il est possible de marquer la définitude des groupes nominaux dotés d'un adjectif qui distingue les deux paradigmes. La phrase *Filip achète un pull bleu* se traduit donc comme *Filip kupuje plav džemper*, alors que la phrase *Filip achète* ***le*** *pull bleu* correspond à *Filip kupuje plav**i** džemper*. Les formes du masculin défini et indéfini sont illustrées dans le tableau 1.4.

	Cas	Défini	Indéfini
Singulier	Nominatif	lep**i**	lep
	Génitif	lep**og**	lep**a**
	Datif	lep**om**	lep**u**
	Accusatif	lep**og**	lep**a**
	Vocatif	lep**i**	-
	Instrumental	lep**im**	lep**im**
	Locatif	lep**om**	lep**u**
Pluriel	Nominatif	lep**i**	lep**i**
	Génitif	lep**ih**	lep**ih**
	Datif	lep**im**	lep**im**
	Accusatif	lep**e**	lep**e**
	Vocatif	lep**i**	-
	Instrumental	lep**im**	lep**im**
	Locatif	lep**im**	lep**im**

TABLE 1.4 – Table de déclinaison des formes du masculin de l'adjectif *lep* 'beau'

Cette distinction est morphologiquement marquée uniquement au singulier des genres masculin et neutre, alors que le singulier du féminin et le pluriel des trois genres marquent l'opposition par le seul moyen de l'accent [1]. Par ailleurs, le paradigme de l'aspect indéfini des genres masculin et neutre semble disparaître de l'usage actif : la distinction est préservée au nominatif et à l'accusatif, alors que pour les autres cas on n'utilise que les formes de l'aspect défini. La saillance de ce trait au niveau morphosyntaxique est donc relativement faible et son apport au niveau syntaxique n'est pas net. Il est néanmoins encodé dans le corpus serbe MultextEast (cf. Krstev et al., 2004b), décrit en détail dans la section 1.2.1.

1. Le serbe dispose d'un accent qui est à la fois un accent d'intensité et un accent tonique. Il existe quatre accents différents : long descendant *pîvo* 'bière', long ascendant *písati* 'écrire', court descendant *vetar* 'vent', et court ascendant *òtac* 'père'. Cependant, les accents sont rarement marqués dans les textes écrits.

Or, ce fait complexifie le traitement automatique des adjectifs, alors que son utilité peut être remise en question. Il s'agit donc d'un deuxième trait morphosyntaxique qui mérite également d'être considéré lors de la constitution des ressources du TAL pour le serbe.

1.1.5 Aspect verbal

Le serbe dispose d'un système de marquage aspectuel très développé, basé en premier lieu sur des procédés dérivationnels. Les verbes peuvent être perfectifs, imperfectifs ou bi-aspectuels. La grande majorité des verbes imperfectifs disposent de correspondants perfectifs, dérivés souvent par préfixation : *jesti* 'manger' *vs pojesti* 'avoir mangé', *učiti* 'apprendre' *vs naučiti* 'avoir appris'. Certains verbes imperfectifs disposent même des séries entières de correspondants perfectifs qui apportent des nuances sémantiques spécifiques, aspectuelles ou autres. Considérons *čitati* 'lire' et la série *pročitati* 'avoir lu', *dočitati* 'finir de lire', *iščitati* 'lire qch en détail dans sa totalité'.

Pour des analyses plus extensives du système aspectuel serbe en français, nous renvoyons aux travaux de P.-L. Thomas, notamment (Thomas, 1993) et (Thomas, 1998). Notons ici que le conflit entre l'instruction aspectuelle lexicale d'un verbe et celle liée à un temps peut bloquer la création de certaines formes pour certaines classes des verbes. Par exemple, les verbes perfectifs n'ont typiquement pas de formes de l'imparfait (temps intrinsèquement imperfectif), alors que les verbes imperfectifs ne se conjuguent pas à l'aoriste (temps intrinsèquement perfectif).

L'aspect verbal est également annoté dans le corpus serbe MultextEast (cf. Krstev et al., 2004b). Son intérêt pour des études linguistiques est évident, notamment dans une perspective contrastive. Or, ce trait n'a pas d'incidence sur le fonctionnement syntaxique de la phrase, alors que son encodage complexifie de manière importante la représentation des verbes en corpus. Le statut à accorder à ce trait dans les treebanks et son utilité pour le parsing représentent donc également un sujet de discussion.

1.1.6 Rapport entre le serbe et le croate (et le bosniaque et le monténégrin)

Avant de poursuivre, il faut indiquer qu'il existe une langue très proche du serbe qui est mieux dotée du point de vue du TAL : le croate. Comme nous avons exploité certaines ressources construites pour cette langue dans le cadre de cette thèse, nous précisons ici le rapport entre le croate et le serbe.

Avant la décomposition de l'ex-Yougoslavie, le serbe et le croate (ainsi que le bosniaque et le monténégrin) étaient considérés comme une seule langue, typiquement désignée par le nom *serbo-croate* ou *croato-serbe*. La création des états indépendants a mené à la constitution des langues nationales. Leur statut est débattu depuis lors. Sans entrer dans des

considérations socio-politiques complexes et sensibles, on peut résumer le rapport entre ces langues, à la suite de (Thomas, 1994), en disant que le serbe, le croate, le bosniaque et le monténégrin sont quasiment identiques aux niveaux phonologique, morphologique et syntaxique. Par ailleurs, les différences existantes à ces niveaux sont largement régulières et prévisibles. Des différences plus importantes existent au niveau lexical, mais elles sont comparables à celles entre deux variétés diatopiques de la même langue et n'empêchent pas une compréhensibilité mutuelle élevée des locuteurs sur le terrain. Ces langues sont par ailleurs souvent désignées par un nouveau nom commun : *bosniaque-croate-monténégrin-serbe* ou BCMS.

Dans le cadre de cette thèse, nous tirons profit de cette situation particulière. En effet, parmi les quatre langues citées, le croate est le mieux doté du point de vue du TAL (cf. Agić & Ljubešić, 2014 ; Agić et al., 2014 ; Agić & Merkler, 2013 ; Agić et al., 2013a,b ; Berović et al., 2012 ; Ljubešić & Klubička, 2014 ; Ljubešić et al., 2016 ; Merkler et al., 2013 ; Tadić, 2007). Qui plus est, cette communauté pratique la libre diffusion de ressources et données. Nous explorons donc les travaux effectués sur cette langue comme une source d'indication de méthodes efficaces et comme un échelon de comparaison pour nos propres expériences. Nous exploitons également certaines ressources initialement développées pour cette langue. Ce sujet sera abordé plus en détail dans le chapitre 3.

1.2 Ressources et outils disponibles pour le traitement automatique du serbe

Les premiers travaux en parsing du serbe et la première tentative de création d'un treebank pour cette langue sont très récents, (cf. Jakovljević et al., 2014). Par conséquent, le serbe a été absent de la campagne d'évaluation CoNLL dédiée au parsing multilingue en 2006 (Buchholz & Marsi, 2006) et il ne figure pas non plus parmi les langues abordées dans le cadre des journées d'étude SPMRL (cf. Tsarfaty et al., 2010 ; Seddah et al., 2013, 2014). En revanche, cette langue a fait partie du projet MultextEast (Erjavec, 2012), ce qui a permis la création d'un corpus doté d'annotations en lemmes et en informations morphosyntaxiques, ainsi que la confection d'un premier lexique morphosyntaxique (Krstev et al., 2004b). Par ailleurs, un ensemble de travaux assez important signale l'existence d'un dictionnaire INTEX (Vitas & Krstev, 2004), d'un WordNet (Krstev et al., 2004a), de corpus annotés en lemmes et informations morphosyntaxiques (Krstev & Vitas, 2005 ; Jakovljević et al., 2014), ou encore d'un étiqueteur basé sur des règles construites manuellement (Sečujski, 2009). Il est donc tout à fait justifié de se demander si le serbe mérite d'être qualifié de langue peu dotée du point de vue du TAL. Or, comme il a été remarqué par Agić et al. (2013b), la communauté qui travaille sur le traitement automatique du serbe ne semble pas avoir adopté la culture du libre partage et de l'échange des données.

Par conséquent, un grand nombre des ressources citées ci-dessus sont indisponibles ; ou alors, si elles sont diffusées, elles sont soumises à des licences restrictives, ne permettant pas la modification ou la rediffusion des données. Un bilan plus détaillé est donné dans la suite.

1.2.1 Les corpus du serbe

Aujourd'hui, il existe plusieurs corpus en serbe. Ils sont aussi bien monolingues (Ljubešić & Klubička, 2014 ; Krstev & Vitas, 2005) que parallèles (Vitas & Krstev, 2006 ; Krstev & Vitas, 2011 ; Tiedemann, 2009 ; von Waldenfels, 2006 ; Čermák & Rosen, 2012), et certains d'entre eux sont également annotés à différents niveaux. Par exemple, le corpus du serbe contemporain SrpKor (Krstev & Vitas, 2005) est lemmatisé et étiqueté en parties du discours (Utvić, 2011), et srWac, le corpus web du serbe (Ljubešić & Klubička, 2014) dispose également d'annotations syntaxiques. Cependant, dans les deux cas, l'annotation a été faite de manière entièrement automatique, sans validation manuelle ultérieure. Par conséquent, ces corpus ne représentent pas une base idéale pour l'apprentissage et l'évaluation des outils automatiques.

En effet, les corpus serbes adaptés à l'évaluation des outils du TAL ne sont pas nombreux. Le plus connu d'entre eux est celui du projet MultextEast (Krstev et al., 2004b). La ressource contient environ 104 000 tokens, elle a été lemmatisée et dotée d'une annotation morphosyntaxique détaillée. C'est ce corpus qui est majoritairement utilisé dans les expériences de TAL sur le serbe (Popović, 2010 ; Gesmundo & Samardžić, 2012). Il est librement disponible à des fins non lucratives [2]. Cependant, sa pertinence peut être remise en cause, vu qu'il s'agit d'une traduction et non d'un texte original serbe : le corpus est entièrement basé sur la traduction serbe du roman *1984* de G. Orwell.

Nous avons constitué, dans le cadre d'une recherche antérieure, un corpus étiqueté et lemmatisé manuellement (Miletic, 2013). Il s'agit de ParCoTrain, qui contient environ 150 000 tokens provenant de trois ouvrages littéraires serbes du 20^{e} siècle [3]. Le corpus est diffusé à des fins non lucratives [4].

Quant aux treebanks, un premier effort de constitution d'un tel corpus pour le serbe est signalé par Jakovljević et al. (2014). Cependant, après la création d'un échantillon initial de 7 000 tokens utilisé dans ce travail, le projet ne semble pas avoir abouti à la

2. Téléchargeable à partir de l'adresse suivante : `https://www.clarin.si/repository/xmlui/handle/11356/1043` sous la licence CC BY-NC-SA 4.0 (`https://creativecommons.org/licenses/by-nc-sa/4.0/`)

3. Kiš, Danilo. *Enciklopedija mrtvih*, 2000. Beograd : BIGZ.
Kiš, Danilo. *Bašta, pepeo*, 2010. Podgorica : Narodna knjiga.
Stevanović, Vidosav. *Testament*, 1986. Beograd : SKZ.

4. Téléchargeable à partir de l'adresse `http://parcolab.univ-tlse2.fr/en/about/resources/` ou `http://redac.univ-tlse2.fr/corpus/parcotrain_fr.html` sous la licence CC BY-NC-SA 3.0 (`https://creativecommons.org/licenses/by-nc-sa/3.0/`).

création d'un treebank complet.

Très récemment (automne 2017), un treebank serbe a été diffusé dans le cadre du projet Universal Dependencies [5]. Le corpus contient 86 000 tokens provenant de journaux et de Wikipédia, et il est annoté en lemmes, traits morphosyntaxiques fins et fonctions syntaxiques. Une description initiale du projet (alors en cours) a été proposée dans (Samardžić et al., 2017). Le corpus peut être téléchargé à partir du site du projet [6].

1.2.2 Les lexiques constitués pour le serbe

Les travaux existants référencent plusieurs ressources lexicales pour le serbe. Parmi elles, on trouve le dictionnaire morphologique AlfaNum, évoqué par Jakovljević et al. (2014). Ce dictionnaire contient 3,8 millions de formes fléchies provenant de 100 000 lexèmes différents. Malheureusement, le travail n'indique aucune modalité de diffusion, et nos propres recherches n'ont pas permis de le repérer.

Un dictionnaire INTEX pour le serbe existe également (Vitas & Krstev, 2004). Il comprend 980 000 formes fléchies provenant de 70 000 lemmes, qui correspondent à 80 000 entrées environ (Pavlović-Lažetić et al., 2004). Malgré son utilisation relativement répandue dans d'autres travaux de la même équipe (cf. Krstev et al., 2004b,a), à notre connaissance, cette ressource n'est pas librement diffusée. Il en est de même pour le dictionnaire SrpMD (Krstev, 2008), qui contient 85 000 lemmes : il est indexé sur le site Meta-Share, mais n'est pas disponible en téléchargement [7].

Le volet serbe du projet MultextEast a abouti à la création d'un lexique morphosyntaxique librement diffusé [8] sous la même licence que le corpus MultextEast. Cependant, il contient seulement 20 000 entrées, correspondant à environ 17 000 formes fléchies et 8 000 lemmes (Krstev et al., 2004b). C'est très peu pour une langue à morphologie flexionnelle riche (cf. la taille des paradigmes flexionnels serbes décrits dans la section 1.1.1).

Depuis le début de cette thèse, deux autres lexiques morphosyntaxiques du serbe ont été diffusés. Dans le cadre de nos propres travaux, nous avons constitué Wikimorph-sr à partir du Wiktionnaire pour le serbo-croate (Miletic, 2017). Il contient 3,1 millions d'entrées uniques (1,2 millions de formes fléchies et 117 000 lemmes) [9]. SrLex a été construit par Ljubešić et al. (2016) dans le cadre d'une campagne de création manuelle basée sur des listes de fréquences. Il comprend 5,3 millions d'entrées uniques (1,4 millions de formes

5. Pour plus de détails sur le projet, voir la section 2.1

6. Téléchargeable à partir de l'adresse suivante : `http://universaldependencies.org/`, sous la licence CC BY-SA 4.0 (`https://creativecommons.org/licenses/by-sa/4.0/`).

7. `http://metashare.ilsp.gr:8080/repository/browse/serbian-morphological-dictionary-multext-east/dad2b9848be011e29ebd001517144592d5a00254a9fd45bb9383caa72801461a/`

8. `https://www.clarin.si/repository/xmlui/handle/11356/1042`

9. Téléchargeable à l'adresse suivante : `http://redac.univ-tlse2.fr/lexiques/wikimorph-sr_fr.html`, sous la licence CC BY-SA 3.0 (`https://creativecommons.org/licenses/by-nc-sa/3.0/`).

fléchies et 105 000 lemmes) [10]. Comme nous avons exploité ces deux lexiques dans le cadre de cette thèse, ils sont présentés en détail dans les sections 6.1 et 6.2.

1.2.3 Outils et modèles de traitement automatique pour le serbe

Étant donné le faible nombre de corpus et lexiques et leur disponibilité souvent limitée, il n'est pas surprenant de constater qu'il existe peu d'outils et de modèles de traitement automatique du serbe, notamment dans le domaine de l'apprentissage automatique supervisé [11]. À notre connaissance, le seul étiqueteur et lemmatiseur par apprentissage automatique construit spécifiquement pour cette langue est BTagger de Gesmundo & Samardžić (2012). Un étiqueteur morphosyntaxique basé sur des règles construites manuellement est également cité par Jakovljević et al. (2014), mais à la différence de BTagger, il ne semble pas librement diffusé [12]. Une méthode de parsing symbolique basée sur le formalisme LTAG a été constituée récemment (Đorđević, 2017). Cependant, cette grammaire traite un ensemble de phénomènes réduit et n'a pas encore été soumise à une évaluation quantitative sur des données réelles (*ibid*, p. 204 207).

Quant aux modèles de traitement développés par des outils initialement conçus pour d'autres langues, la situation est légèrement meilleure. Il s'agit notamment de modèles entraînés sur des ressources croates. Grâce aux travaux de Agić et al. (2013a) et de Ljubešić et al. (2016), des modèles pour l'étiquetage morphosyntaxique avec un étiqueteur HMM [13] (Halácsy et al., 2007) et avec un étiqueteur CRF [14] sont disponibles. Des modèles de parsing avec MST parser (McDonald et al., 2006) et Mate parser (Bohnet, 2010) ont été créés respectivement par Agić et al. (2013b) et Agić & Ljubešić (2015). Tous ces modèles sont librement diffusés [15].

Une analyse plus poussée des outils, des modèles et de leurs performances est disponible dans les sections dédiées à l'étiquetage (cf. section 3.2.7), à la lemmatisation (cf. section 3.3.4) et au parsing du serbe (cf. section 3.4.3).

10. La version la plus récente est disponible à l'adresse suivante : `https://www.clarin.si/repository/xmlui/handle/11356/1073`. SrLex est soumis à la licence GNU General Public Licence, v3 (`https://opensource.org/licenses/GPL-3.0`).
11. Le processus d'apprentissage automatique global, ainsi que les tâches concrètes d'étiquetage morphosyntaxique, de lemmatisation et de parsing, sont décrits dans la section 3.1.
12. Il s'agit de l'étiqueteur AlfaNum Tagger de Sečujski (2009).
13. Algorithme d'apprentissage automatique Hidden Markov Model. Voir la section 3.2.1.
14. Algorithme d'apprentissage automatique Conditional Random Fields. Voir la section 3.2.5.
15. `http://nlp.ffzg.hr/`

1.3 Traitement automatique des langues à morphologie flexionnelle riche

Depuis les années 2000, le développement des outils d'analyse syntaxique automatique prend son essor, notamment en ce qui concerne le parsing de l'anglais (cf. Charniak, 2000 ; Collins, 2003 ; McDonald et al., 2005a ; Petrov et al., 2006 ; Nivre et al., 2007b ; Chen & Manning, 2014 ; Dyer et al., 2015 ; Kiperwasser & Goldberg, 2016 ; Andor et al., 2016). L'efficacité des parsers a évolué avec le temps : par exemple, MST parser de McDonald et al. (2005a) et Malt parser de Nivre et al. (2007b) obtiennent respectivement une exactitude de 90,9 % et 88,1 % sur l'anglais, alors que le système d'Andor et al. (2016) atteint une exactitude de 94,6 % sur la même langue. Or, ces mêmes outils atteignent systématiquement des résultats beaucoup moins élevés sur des langues à morphologie flexionnelle riche : sur le tchèque, les parsers cités obtiennent respectivement une exactitude de 83,3 %, 80,1 % et 88,94 %[16]. Ils marquent donc tous une chute de performances de plus de 5 % par rapport à l'anglais. Cette tendance est confirmée par les résultats des campagnes d'évaluation CoNLL de 2006 (Buchholz & Marsi, 2006) et 2007 (Nivre et al., 2007a).

Cette difficulté de traitement est en premier lieu due, comme nous venons de le voir, à la nature même de ces langues. Par ailleurs, les langues à morphologie flexionnelle riche disposent le plus souvent de corpus beaucoup plus petits que ceux utilisés pour l'anglais : parmi les échantillons de données utilisés dans le cadre de la campagne d'évaluation SPMRL 2013 (Seddah et al., 2013), les seules langues à disposer d'échantillons d'entraînement supérieurs à 500 000 tokens étaient l'arabe et l'allemand, alors que pour les autres langues les tailles variaient entre 76 000 (suédois) et 443 000 tokens (français). La taille du corpus standard utilisé pour le parsing de l'anglais (PennTreebank) est de 1 million de tokens (Marcus et al., 1993). Même si cette situation tend à s'améliorer ces dernières années (cf. l'initiative Universal Dependencies, section 2.1), les langues à morphologie flexionnelle riche souffrent encore chroniquement du manque de données annotées en grandes quantités.

Ce fait est essentiel étant donné les caractéristiques intrinsèques de ces langues évoquées ci-dessus : ces corpus de tailles relativement limitées sont typiquement insuffisants pour assurer une bonne couverture de la multitude de formes fléchies, de traits morphosyntaxiques et de structures syntaxiques dont ces langues disposent. Au niveau de l'apprentissage automatique, ce fait se traduit par une dispersion des données : le nombre d'occurrences de différentes catégories reste bas, et par conséquent les outils automatiques ne sont pas capables de les maîtriser.

16. Nous rapportons ici les scores pour l'identification du gouverneur sans étiquetage de la relation syntaxique (*Unlabelled Attachment Score*), étant donné que c'est le seul score présenté pour MST dans le travail cité. Pour plus de détails sur les métriques du parsing, v. section 3.4.

Comme nous l'avons vu dans la section 1.1, le serbe exhibe les mêmes propriétés linguistiques. Il fait également partie des langues qui ne disposent pas de grands corpus adaptés à l'entraînement des outils automatiques : la taille des corpus existants de ce type ne dépasse pas 150 000 tokens (cf. section 1.2.1). Il est donc tout à fait justifié de s'attendre à ce que cette langue pose le même type de difficultés dans le cadre du TAL. Dans la suite de cette section, nous nous intéressons donc de plus près aux origines de la « dispersion des données » dans les langues à morphologie flexionnelle riche, ainsi qu'aux méthodes proposées pour pallier ce problème. Au cours du temps, différentes stratégies ont émergé, comme l'annotation morphosyntaxique et syntaxique jointe (Goldberg & Tsarfaty, 2008), l'utilisation des lexiques factorisés (Green et al., 2013), l'exploitation de données bilingues (Fraser et al., 2013), ou encore l'utilisation des *word embeddings* (Ling et al., 2015 ; Müller & Schütze, 2015). Cependant, l'objectif de cette thèse étant de fournir pour le serbe des outils de base, qui pourront être exploités dans un maximum de tâches différentes du TAL, nous focalisons notre attention sur deux méthodes traditionnelles pour diminuer la dispersion des données : l'utilisation des lexiques externes et la lemmatisation.

1.3.1 Facteurs à l'origine de la dispersion des données

Un système de morphologie flexionnelle riche induit une dispersion des données dans la mesure où le nombre d'occurrences des unités individuelles observées en corpus (que ce soit des formes fléchies ou des étiquettes morphosyntaxiques) est plus bas que pour une langue à morphologie flexionnelle réduite. Quant à la dispersion des données au niveau des formes fléchies, il a été observé qu'un corpus hongrois de 250 000 tokens contenait deux fois plus de formes fléchies différentes qu'un corpus d'anglais de taille comparable (Oravecz & Dienes, 2002). Pour le turc, le ratio est de 4 pour 1 par rapport à l'anglais (Hakkani-Tür et al., 2002). Un examen rapide d'un échantillon de texte serbe de 101 000 tokens [17] permet de voir que ce ratio est d'environ 3 à 1 par rapport à l'anglais (cf. tableau 1.5) si l'on se base sur les informations fournies pour l'anglais dans (Oravecz & Dienes, 2002).

Langue	No. tokens	No. formes fléchies	F.fléchie/tok
anglais	245 714	19 021	0,08
serbe	101 425	22 770	**0,22**

TABLE 1.5 – Rapport forme fléchie/token en anglais et en serbe

Cela se traduit par deux effets différents : par une dispersion des données au niveau des formes fléchies dans le corpus d'entraînement, mais aussi par un nombre élevé de formes inconnues de l'outil (absentes de son corpus d'entraînement) lors du traitement

17. Il s'agit de l'échantillon à la base du corpus constitué dans le cadre de cette thèse (cf.section 7.1).

de nouveaux textes. Deux méthodes pour limiter ces effets sont discutées dans la section suivante.

La dispersion des données se manifeste également au niveau de l'annotation morphosyntaxique, suivant le même principe : les langues à morphologie flexionnelle riche disposent de plus de traits morphosyntaxiques que les langues à morphologie flexionnelle réduite. Si tous ces traits sont encodés en corpus, la majorité d'entre eux auront peu d'occurrences dans un corpus de taille standard, ce qui les rend difficiles à apprendre. Comme ce phénomène est intrinsèquement lié à la structure du jeu d'étiquettes morphosyntaxiques, nous y revenons dans la section dédiée à ce sujet (cf. section 2.3.1).

1.3.2 Moyens de neutralisation de la dispersion des données au niveau lexical

L'un des mécanismes les plus communs pour contrer la dispersion des données au niveau des formes fléchies réside dans l'utilisation de lexiques morphosyntaxiques externes. Un lexique prend typiquement la forme d'une liste de formes fléchies d'une langue accompagnées de leur lemme et de leurs interprétations morphosyntaxiques possibles. Ces ressources peuvent être exploitées de deux manières principales : elles peuvent fournir des contraintes au moment de l'annotation, par exemple en proposant les étiquettes morphosyntaxiques valides pour une forme et en confiant la désambiguïsation à l'étiqueteur (Kim et al., 1999 ; Hajič, 2000), ou bien elles peuvent apporter des informations complémentaires au moment de l'apprentissage, par exemple en transformant les informations morphosyntaxiques présentes dans le lexique en des traits d'apprentissage additionnels (Denis & Sagot, 2012 ; Sagot, 2016 ; Goldberg et al., 2009).

Le travail de Hajič (2000) montre clairement l'impact de l'utilisation d'un lexique externe sur l'annotation morphosyntaxique. Les auteurs comparent les performances en étiquetage morphosyntaxique détaillé de 6 langues selon plusieurs scénarios : avec ou sans lexique externe, et sur des corpus d'apprentissage de tailles différentes. Dans le tableau 1.6, nous reprenons les résultats obtenus sans lexique externe sur les corpus complets, dont la taille varie entre 81 000 et 104 000 tokens en fonction de la langue (scénario 1) et ceux obtenus sur des échantillons d'apprentissage réduits de 2 000 tokens avec un lexique externe (scénario 2). Les résultats représentent le taux d'erreur (le pourcentage de tokens mal annotés).

Nous constatons que l'utilisation d'un échantillon d'entraînement minimal de 2 000 tokens complétée par l'exploitation d'un lexique externe permet d'atteindre des résultats comparables à l'utilisation d'un corpus d'entraînement 40 à 50 fois plus grand. Ces résultats suggèrent donc que le fait d'investir un effort dans la création d'un lexique morphosyntaxique externe peut être aussi bénéfique à l'étiquetage morphosyntaxique d'une

Langue	Taux d'erreur (tokens hors ponctuations) Scénario 1	Scénario 2
Anglais	9,18 %	7,64 %
Tchèque	18,83 %	18,07 %
Estonien	13,95 %	11,95 %
Hongrois	8,16 %	5,35 %
Roumain	7,76 %	9,47 %
Slovène	16,26 %	19,17 %

TABLE 1.6 – Effet du lexique *vs* effet du corpus d'apprentissage dans (Hajič, 2000)

langue à morphologie riche que le développement d'un corpus d'apprentissage étendu. Il faut cependant préciser que la constitution d'un lexique à couverture large peut être tout aussi coûteuse que la confection d'un corpus, notamment si le processus est manuel (cf. Ljubešić et al., 2016).

Une autre façon de diminuer la dispersion des données au niveau lexical consiste à effectuer la lemmatisation. Autrement dit, on attribue à chaque forme fléchie sa forme canonique. Grâce à cela, on regroupe les occurrences liées au même lemme, ce qui permet d'augmenter le nombre d'occurrences de chaque catégorie observée. Par conséquent, l'apprentissage automatique est facilité.

Le travail de Seddah et al. (2010) montre que la lemmatisation d'un corpus français apporte une amélioration des scores en parsing, bien que légère. En revanche, le même procédé sur un corpus anglais de taille comparable n'a aucun effet. Ces observations ont également été confirmées sur l'espagnol par Le Roux et al. (2012), où la lemmatisation a un impact plus net.

À cette fin, il est également possible d'exploiter le *clustering* des unités utilisées pour l'apprentissage du parsing. En particulier, Candito & Seddah (2010) remplacent les tokens par des clusters de formes fléchies et par des clusters de paires lemme - étiquette morphosyntaxique. Les deux méthodes permettent d'améliorer le parsing, et le gain est le plus prononcé dans le traitement des mots inconnus de l'outil ou peu fréquents dans le corpus d'apprentissage.

Plus récemment, on a également commencé à exploiter des informations lexicales non supervisées sous forme de *word embeddings*, aussi bien en étiquetage qu'un parsing (Bengio et al., 2003 ; Collobert & Weston, 2008 ; Ling et al., 2015 ; Müller & Schütze, 2015). Cette technique consiste à représenter différents mots sous la forme de vecteurs à partir des observations faites sur des corpus larges non annotés et de capter les similarités de fonctionnement en comparant ces vecteurs. Des travaux récents indiquent que les informations fournies par les *word embeddings* et celles provenant des lexiques morphosyntaxiques sont en effet complémentaires (cf. Sagot & Alonso, 2017).

1.4 Exigences posées par le serbe

Les faits présentés dans ce chapitre permettent tout d'abord d'expliquer l'orientation de notre travail : comme le montre la section 1.2, en début de cette thèse, le serbe ne disposait pas de treebank, et cette langue nécessitait également des ressources lexicales librement disponibles. Bien que des outils entraînés sur le croate aient été utilisés avec succès sur le serbe, les modèles entraînés directement sur les données en serbe étaient rares. C'est en fonction de ce bilan que nous avons identifié nos objectifs de constitution de ressources : la création d'un treebank serbe, et l'entraînement et la diffusion des modèles de traitement automatique à partir de ce corpus.

Cet état des lieux permet également d'identifier certaines exigences vis-à-vis des ressources envisagées qui découlent des caractéristiques du serbe. Nous les résumons ici.

Intérêt d'une annotation morphosyntaxique fine. Nous avons vu dans la section 1.1.2 que le décodage du fonctionnement syntaxique du serbe s'appuie fortement sur les traits morphosyntaxiques fins comme le cas, le genre, le nombre ou la personne. Un parser doit donc disposer de ce type d'information pour obtenir des performances solides sur cette langue. Par conséquent, un treebank doit en être doté.

Besoin de représenter des structures syntaxiques discontinues. Dans la section 1.1.2, nous avons illustré le fait que le serbe autorise des structures discontinues. L'annotation syntaxique d'un treebank serbe exige donc un cadre qui permet de représenter ce type de constructions. Nous revenons sur cette question dans la section 2.2, puis dans le chapitre 11.

Nécessité de la lemmatisation et utilité d'un lexique. La richesse des paradigmes flexionnels du serbe et le nombre de formes fléchies indiquent que le degré de dispersion des données dans cette langue est important. Cela signifie qu'un treebank pour cette langue doit de préférence être annoté en lemmes, et qu'il est utile de disposer d'un lexique morphosyntaxique.

Nos observations de la section 1.1 sur le statut de différents traits morphosyntaxiques du serbe dans le cadre du TAL ont également nourri des décisions plus précises, sur lesquelles nous revenons dans le cadre de la définition des principes d'annotation morphosyntaxique (cf. section 5.1).

Après avoir identifié ces points de départ pour notre campagne de constitution des ressources pour le serbe, nous abordons plus précisément la question de la création d'un treebank. Les éléments et les principes de ce processus sont présentés dans le chapitre suivant.

Chapitre 2

Constitution des treebanks

Les débuts du développement des corpus remontent aux années 1960 avec le corpus Brown (Francis & Kučera, 1979), le premier grand corpus textuel qui a connu une diffusion très large. Ce sont cependant les années 1990 qui apportent une véritable expansion de ces bases textuelles avec la création des ressources comme le British National Corpus (BNC) [1], le Czech National Corpus (CNC) [2] ou encore Frantext [3]. Un processus d'expansion et de diversification des corpus est en cours depuis : au-delà des premières ressources généralistes monolingues principalement basées sur la langue écrite, des corpus plurilingues (Ide & Véronis, 1994 ; Erjavec & Ide, 1998 ; Steinberger et al., 2006 ; Koehn, 2005), spécialisés (Qi-bo, 1989 ; Kim et al., 2003) ou oraux (Svartvik, 1990 ; Godfrey et al., 1992 ; Cresti & Moneglia, 2005) sont créés.

L'importance de l'annotation linguistique des corpus a été reconnue dès les débuts mêmes de leur développement : déjà les premiers corpus nationaux mentionnés ci-dessus étaient dotés d'un étiquetage morphosyntaxique. Des annotations syntaxiques (Marcus et al., 1993 ; Hajič, 1998), sémantiques (Baker et al., 1998 ; Palmer et al., 2005) ou discursives (Prasad et al., 2005 ; Afantenos et al., 2012) ont été mises au point depuis. Dans leur travail présentant PennTreebank, le premier corpus annoté syntaxiquement, Marcus et al. expriment l'avis suivant par rapport à l'utilité de ce type de ressources :

> « Annotated corpora promise to be valuable for enterprises as diverse as the automatic construction of statistical models for the grammar of the written and the colloquial spoken language, the development of explicit formal theories of the differing grammars of writing and speech, the investigation of prosodic phenomena in speech, and the evaluation and comparison of the adequacy of parsing models. » (Marcus et al., 1993, p. 313).

1. `http://www.natcorp.ox.ac.uk/corpus/index.xml`
2. `https://www.korpus.cz/`
3. `http://www.frantext.fr/`

Effectivement, l'existence des corpus enrichis a permis de nombreuses avancées dans différents cadres applicatifs. À titre illustratif, de grands corpus généralistes ont été exploités pour la création de dictionnaires et de grammaires (cf. Kjellmer, 1994 ; Sinclair, 1987 ; Quirk et al., 1985) ; des corpus oraux ont été utilisés dans les domaines de la reconnaissance et de la génération de la parole (cf. Godfrey et al., 1992 ; Black & Tokuda, 2005 ; Ellbogen et al., 2004) ; des applications en génération du langage naturel bénéficient des corpus annotés en relations du discours (cf. Prasad et al., 2005) ; des corpus parallèles ont servi pour la mise en place et l'évaluation de méthodes de traduction automatique statistique (cf. Koehn, 2005) ; des corpus spécialisés sont utilisés en recherche d'information dans leurs domaines respectifs (cf. Kim et al., 2003). Enfin, cette thèse s'intéresse particulièrement à un dernier mode d'exploitation des corpus annotés : leur utilisation en tant que corpus d'apprentissage pour les outils d'analyse linguistique basés sur l'apprentissage automatique supervisé.

Les outils développés dans ce paradigme sont basés sur des algorithmes statistiques qui, en parcourant un texte annoté, dérivent les probabilités pour les différents comportements des catégories observées. L'ensemble de ces probabilités constitue un modèle de traitement que l'outil exploite par la suite pour traiter des textes inconnus. Ces algorithmes probabilistes peuvent être adaptés à différents niveaux d'annotation et ont été appliqués avec succès à l'étiquetage en parties du discours, au parsing, mais aussi à la lemmatisation (voir le chapitre 3). L'un de leurs points forts principaux réside dans le fait qu'ils sont indépendants de la langue : le même algorithme peut apprendre à traiter toute langue, sous condition de disposer d'un corpus d'apprentissage adapté. Il est donc peu surprenant de constater que les méthodes par apprentissage automatique supervisé dominent les travaux en TAL et qu'ils ont écarté les méthodes symboliques, basées sur des règles écrites par des experts humains.

Cette prépondérance des outils probabilistes a été rendue possible par la généralisation des corpus annotés eux-mêmes, et notamment par le développement des treebanks. Les treebanks (banques d'arbres) sont des corpus de textes annotés en structures et fonctions syntaxiques, mais aussi en lemmes et en informations morphosyntaxiques. Il s'agit donc de ressources polyvalentes, adaptées au développement d'outils pour différents niveaux d'analyse.

Or, la constitution d'une telle ressource est un processus complexe et coûteux. Il est en grande partie déterminé par la double nature des treebanks : ils sont à la fois destinés à l'exploration linguistique et à l'apprentissage automatique. Les exigences liées à ces deux applications sont difficiles à concilier, et un effort de conception est nécessaire pour assurer un équilibre entre elles. Afin de mieux en rendre compte, la suite de ce chapitre sera consacrée à l'analyse des principes qui guident la création d'un treebank, ainsi qu'aux contraintes auxquelles elle est soumise. Après un bref historique du développement

des treebanks (section 2.1), nous discutons les différents cadres théoriques disponibles (section 2.2) et abordons l'importance des jeux d'étiquettes utilisés pour l'annotation (section 2.3). Nous évoquons ensuite la question de la qualité de l'annotation en corpus (section 2.4), les méthodes qui permettent d'accélérer le processus en exploitant des outils du TAL existants (section 2.5) et l'organisation globale des campagnes d'annotation (section 2.6). Enfin, nous résumons les principes retenus pour la suite de notre travail (section 2.7).

2.1 Bref historique des treebanks

Les treebanks sont apparus plus tardivement et se sont répandus plus lentement que les corpus dotés d'une annotation morphosyntaxique. Le premier grand treebank publié a été PennTreebank, développé par Marcus et al. (1993) à l'Université de Pennsylvanie à partir de 1989. La version du corpus décrite dans (Marcus et al., 1993) contenait 4,5 millions tokens, dont 2,8 millions étaient dotés d'une annotation syntaxique en constituants. Il a ensuite fallu attendre la fin du 20^{e} et le début du 21^{e} siècle pour que des treebanks d'autres langues deviennent disponibles.

L'un des premiers treebanks à voir le jour ensuite est le treebank NEGRA pour l'allemand (Skut et al., 1997). Il s'agit d'un corpus journalistique, annoté en constituants enrichis de fonctions syntaxiques. La version actuelle contient 355 000 tokens annotés et elle est librement disponible pour des exploitations en recherche [4]. Les schémas d'annotation mis en place dans ce corpus ont servi de point de départ pour la constitution des treebanks allemands ultérieurs (Brants et al., 2002).

Un autre projet de longue date est celui de Prague Dependency Treebank (dorénavant PDT) (Hajič, 1998). Depuis 1996, ce projet vise la création d'un corpus tchèque annoté à plusieurs niveaux : morphologie, syntaxe de surface (dite *analytical level* dans le cadre du projet) et syntaxe profonde (*tectogrammatical level*). Le projet fait appel à l'analyse syntaxique en dépendances (cf. section 2.2). Depuis la version 2.0, diffusée en 2006, le corpus contient 2 millions de tokens annotés en informations morphologiques, 1,5 million de tokens annotés au niveau de la syntaxe de surface, et 0,8 million de tokens annotés en syntaxe profonde. La version suivante a visé l'amélioration des annotations plutôt que l'extension du corpus [5].

La constitution du corpus French Treebank (dorénavant FTB) a été lancée en 1997 [6]. La première version était annotée en constituants portant des labels des fonctions syntaxiques (Abeillé et al., 2003), mais la ressource a également connu une conversion vers une

4. http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html
5. https://ufal.mff.cuni.cz/pdt3.0/documentation
6. http://ftb.linguist.univ-paris-diderot.fr/

représentation en dépendances, décrite dans (Candito et al., 2010a). La version actuelle contient 664 000 tokens annotés.

Le corpus national russe intègre également une partie sous forme de treebank nommée SynTagRus (Boguslavsky et al., 2002b). Le projet met en place une annotation en dépendances, et la ressource a dépassé la taille de 770 000 tokens (Boguslavsky, 2014).

Après ces débuts relativement lents, on note une généralisation des treebanks de plus en plus marquée, notamment depuis le début des années 2000. Pour n'en citer que quelques-uns, des projets de création de treebanks se lancent pour le bulgare (cf. Simov et al., 2002), le néerlandais (cf. Van der Beek et al., 2002), l'espagnol et le catalan (cf. Taulé et al., 2008), l'espagnol (cf. Mille et al., 2013), l'italien (cf. Montemagni et al., 2003 ; Tonelli et al., 2008), l'arabe (cf. Maamouri et al., 2004 ; Smrž et al., 2008 ; Habash et al., 2009), etc. Les campagnes d'évaluation CoNLL [7], et notamment les éditions de 2006 (Buchholz & Marsi, 2006) et 2007 (Nivre et al., 2007a), ont contribué à la stabilisation du format d'encodage des données pour les treebanks, mais aussi à la constitution et à la diffusion des corpus. Les journées d'étude SPMRL ont apporté une contribution spécifique sur ces points pour les langues morphologiquement riches.

Cette démocratisation des treebanks est peut-être la plus visible sur l'exemple du projet Universal Dependencies (dorénavant UD) (Nivre et al., 2016). Ce projet vise la création de treebanks dans différentes langues, basés sur les mêmes principes d'annotation aussi bien au niveau morphologique qu'au niveau syntaxique. L'objectif de cette démarche est de fournir une base unifiée de données langagières, facilitant les études contrastives en linguistique, mais aussi l'évaluation et la comparaison directe des systèmes de parsing. Actuellement, le projet comprend plus de 100 treebanks dans plus de 60 langues différentes [8].

2.2 Cadre théorique : syntaxe en constituants *vs* syntaxe en dépendances

Dans l'analyse syntaxique, aussi bien théorique qu'appliquée au TAL, deux approches principales existent : l'analyse en constituants et l'analyse en dépendances. Les deux cadres ont donné naissance à de nombreuses théories. Parmi les représentants de la grammaire en constituants, on trouve *Government and Binding Theory* (Chomsky, 1993, 1982), *Generalized Phrase Structure Grammar* (GPSG) (Gazdar et al., 1985) et *Head-Driven Phrase Structure Grammar* (HPSG) (Pollard & Sag, 1994). Du côté de la grammaire en dépendances, le travail précurseur est celui de Tesnière (1959). De nombreuses théories ont

7. *Conference on Computational Natural Language Learning*, `http://www.conll.org/previous-editions`.

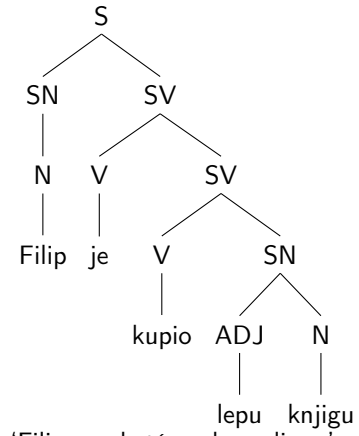
8. La liste complète est disponible sur le site du projet : `http://universaldependencies.org/`.

suivi, parmi lesquelles : *Word Grammar* (WG) (Hudson, 1984), *Functional Generative Description* (FGD) (Sgall et al., 1986), *Dependency Unification Grammar* (DUG) (Hellwig, 1986), *Meaning-Text Theory* (MTT) (Mel'čuk, 1988), ou *Functional Dependency Grammar* (FDG) (Tapanainen & Järvinen, 1997). Le principal critère de distinction entre ces deux approches théoriques est leur vision de la structure syntaxique et des représentations syntaxiques dont elles se servent. Comme les deux visions entraînent des implications importantes aussi bien pour la linguistique que pour le TAL, nous les présentons dans la suite.

2.2.1 Syntaxe en constituants

Dans le cadre de l'analyse syntaxique en constituants, on considère que les constituants (syntagmes ou groupes) sont l'unité de base de la structure syntaxique. Analyser une phrase consiste à la décomposer en constituants niveau par niveau, jusqu'à arriver aux mots mêmes. Sans entrer dans les finesses des théories citées ci-dessus, les arbres syntaxiques résultant de cette approche se présentent comme dans l'exemple 4.

(4)



'Filip a acheté un beau livre.'

Nous pouvons remarquer plusieurs caractéristiques de l'arbre :

1. l'arbre représente une structure à plusieurs niveaux ;
2. la racine de l'arbre (le nœud sommet) est un nœud représentant la phrase ;
3. il n'y a pas de marquage explicite des fonctions syntaxiques : on n'indique que les catégories morphosyntaxiques des syntagmes, et la fonction syntaxique est dérivée de la structure de l'arbre.

Une manière de transposer cette approche à l'annotation de corpus consiste à indiquer la structure en constituants de la phrase en ajoutant des parenthèses ou des crochets délimitant les syntagmes. L'arbre de l'exemple 4 serait donc représenté sous une forme comparable à celle de la figure 2.1.

[S [SN [N Filip]] [SV [V je] [SV [V kupio] [SN [ADJ lepu] [N knjigu]]]]]

FIGURE 2.1 – Analyse en constituants dans le cadre du TAL

Historiquement, c'est cette approche qui a été mise en œuvre la première dans les treebanks (cf. Marcus et al., 1993 ; Skut et al., 1997). Cependant, elle a un point faible important : elle n'offre pas la possibilité de représenter les constituants discontinus d'une manière simple. Quoique rare en anglais, ce phénomène est relativement fréquent dans les langues à morphologie flexionnelle riche et à ordre des constituants flexible (cf. Havelka, 2007), et comme nous l'avons vu dans la section 1.1.2, il existe également en serbe. Pour l'illustrer, nous reprenons ici l'exemple 3.

(3)

Lep-u	je	Filip	knjig-u	kupio.
beau-ACC.SG.F	est	Filip.NOM.SG	livre-ACC.SG.F	acheté

'C'est un beau livre que Filip a acheté.'

Dans un tel cas de figure, une simple décomposition de la phrase en constituants n'est plus possible : le sujet *Filip* se trouve au milieu du syntagme verbal, l'auxiliaire *je* est séparé du verbe principal *kupio* par la tête de l'objet direct *knjigu* et le sujet *Filip*, et par ailleurs, l'adjectif *lepu*, modifieur de l'objet direct, est séparé de sa tête *knjigu* par l'auxiliaire.

Pour rendre compte de ce type de structures dans le cadre de la syntaxe en constituants, il est nécessaire de mettre en place des mécanismes d'annotation complexes, basés sur les traces et les *fillers*. C'est pour cette raison qu'on considère généralement que l'analyse en constituants n'est pas adaptée au traitement des langues dont l'ordre des constituants flexible. Des exceptions existent : le travail de Simov et al. (2002) ; Simov & Osenova (2003) sur le treebank bulgare BulTreeBank, ainsi que celui de Marciniak et al. (1999) sur le polonais, exploitent la théorie HPSG de Pollard & Sag (1994). Néanmoins, force est de constater que la majorité des treebanks des langues de ce type (et notamment des langues slaves) font appel à la syntaxe en dépendances (cf. Hajič, 1998 ; Boguslavsky et al., 2002b ; Džeroski et al., 2006 ; Tadić, 2007 ; Agić & Ljubešić, 2014), et cette tendance s'étend à d'autres types de langues (cf. les treebanks du projet UD).

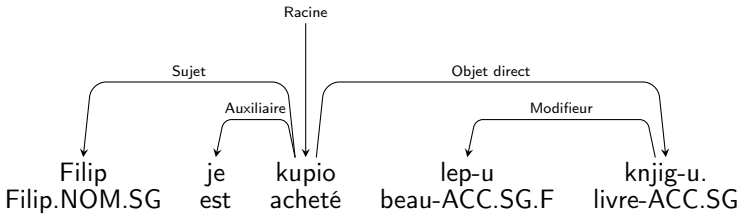
2.2.2 Syntaxe en dépendances

La syntaxe en dépendances considère que la structure syntaxique d'une phrase correspond à un ensemble de relations de dépendance qui s'établissent entre les mots considérés individuellement : chaque mot a un gouverneur et peut à son tour en gouverner d'autres. Pour une introduction à la grammaire en dépendances, nous revnoyons vers (Kahane,

2001).

Dans ce cadre théorique, les arbres syntaxiques se présentent sous la forme indiquée dans l'exemple 5.

(5)



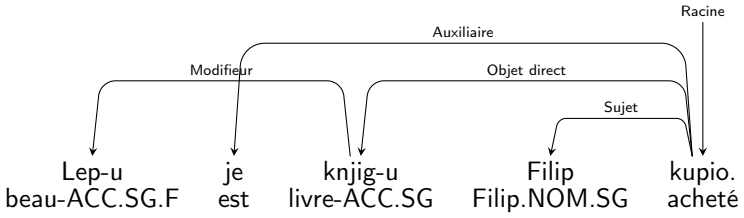
'Filip a acheté un beau livre.'

Les différences principales entre un arbre en constituants et celui-ci sont les suivantes :

1. ici, l'arbre a une structure plate : les relations s'établissent directement entre les mots individuels ;
2. c'est le verbe qui est considéré comme la racine de la phrase [9] ;
3. les fonctions syntaxiques sont annotées explicitement.

L'avantage évident de cette approche réside dans le fait que l'ordre linéaire des mots dans la phrase a peu d'impact sur la représentation de l'analyse. Ainsi, les mêmes relations présentes dans l'exemple 5 se mettent en place dans l'exemple 6, malgré une distribution des mots différente.

(6)



'C'est un beau livre que Filip a acheté.'

À partir de ces observations, nous adoptons ce cadre théorique pour la création de notre treebank. Nous sommes néanmoins consciente des points faibles de cette approche, à savoir notamment la représentation des structures qui ne permettent pas une identification facile du gouverneur, comme la coordination, l'ellipse et les phrases averbales (Brants et al., 2003). Malgré cela, la syntaxe en dépendances gagne une popularité de plus en plus large dans le TAL. En effet, elle pourrait être considérée comme un standard *de facto* pour le

9. Notre choix de considérer le verbe principal en tant que racine de la phrase est discuté dans la section 5.2.3.

parsing, promu notamment par de nombreuses campagnes d'évaluation CoNLL (Buchholz & Marsi, 2006 ; Nivre et al., 2007a ; Surdeanu et al., 2008 ; Hajič et al., 2009 ; Zeman et al., 2017). Outre les treebanks slaves cités dans la section 2.2.1, il existe également des treebanks en dépendances pour le français (Candito et al., 2010a), l'allemand (Foth et al., 2014), l'espagnol (Mille et al., 2013), l'arabe (Hajič et al., 2004), etc., sans mentionner les langues du projet UD.

Avant de poursuivre, nous consacrons la section suivante à l'examen de quelques propriétés communément admises des arbres syntaxiques en dépendances. Cela nous permettra de définir les contraintes que doit respecter l'annotation syntaxique dans notre corpus.

2.2.3 Propriétés des arbres en dépendances

Les arbres de dépendances tels qu'ils sont exploités en TAL prennent la forme d'un graphe. Par conséquent, on utilise souvent la terminologie de la théorie des graphes pour les décrire : les mots (ou les tokens) représentent les **nœuds** du graphe, alors que les dépendances correspondent aux **arêtes**. En plus des caractéristiques globales citées dans la section 2.2.2, ces graphes respectent plusieurs autres critères. Kübler et al. (2009) définissent les propriétés suivantes :

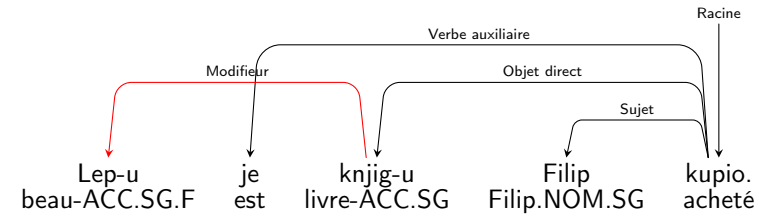
1. **racine unique** : dans l'arbre, il existe exactement un nœud racine qui n'est gouverné par aucun autre nœud ;
2. **caractère couvrant** : l'arbre couvre tous les tokens de la phrase ;
3. **caractère connecté** : il existe un chemin entre n'importe quelle paire de nœuds dans l'arbre ;
4. **caractère orienté** : toutes les arêtes de l'arbre sont orientées ;
5. **gouverneur unique** : un token ne peut avoir qu'un seul gouverneur (chaque token peut être la cible d'une seule dépendance) ;
6. **acyclicité** : il ne peut y avoir de cycles fermés dans l'arbre (aucun token ne peut gouverner un de ses ancêtres).

La contrainte de la racine unique est artificielle, mais elle facilite la réalisation d'autres propriétés (notamment 2 et 3). Elle est souvent instanciée par l'introduction d'un nœud artificiel ROOT en début de phrase. Ce fait permet de satisfaire le caractère couvrant de l'arbre sans avoir à relier tous les sous-arbres entre eux : ils peuvent simplement être rattachés à la racine externe. Ceci est par ailleurs le cas dans Prague Dependency Treebank, où le token racine de la phrase, la ponctuation finale et les modifieurs phrastiques peuvent se trouver simultanément rattachés au nœud ROOT. D'autres projets adoptent l'approche selon laquelle la racine externe n'a qu'un descendant dans la phrase (cf. Candito et al., 2009). L'existence du ROOT facilite également la satisfaction du caractère connecté, vu que

tous les tokens sont en effet des descendants de ce nœud et un chemin entre n'importe quelle paire de tokens peut être retrouvé à travers le ROOT. La propriété 2 (caractère couvrant) répond au principe défini dans des travaux théoriques (cf. Mel'čuk, 1988) selon lequel tous les mots doivent participer à la structure syntaxique de la phrase. Le caractère orienté des arêtes exprime simplement la nature asymétrique des relations de dépendance, autrement dit, le fait que chaque dépendance relie un gouverneur et un dépendant. D'après (Mel'čuk, 1988), les dépendances sont orientées du gouverneur vers le dépendant. La majorité des théories en syntaxe de dépendance reconnaissent également la contrainte du gouverneur unique et celle de l'acyclicité, mais des exceptions existent (cf. Hudson, 1984 ; Debusmann et al., 2004).

Dans certains cadres applicatifs en TAL, on impose une dernière contrainte aux arbres de dépendances exploités : celle de la **projectivité**. Un arbre syntaxique est considéré comme non projectif si au moins une dépendance qui le constitue est elle-même non projective. Si l'on reprend la définition proposée dans (Kuhlmann & Nivre, 2006), une dépendance entre un gouverneur G et un dépendant D est considérée comme non projective s'il existe au moins un token entre G et D dans l'ordre linéaire de la phrase qui n'est pas dominé par G (autrement dit, que G n'est ni sont gouverneur immédiat, ni son ancêtre). Si l'on revient encore une fois à l'exemple 6, on remarque que la relation entre le gouverneur *knjigu* et le dépendant *lepu* est non projective, vu que l'auxiliaire *je*, situé entre ces deux formes, n'est pas dominé par *knjigu*. En revanche, la dépendance entre le verbe principal *kupio* et l'auxiliaire *je* est projective, car les deux tokens intervenants *knjigu* et *Filip* sont gouvernés par *kupio*.

(6)



'C'est un beau livre que Filip a acheté.'

La non-projectivité est le reflet de la nature discontinue des constituants dans une phrase ; le taux de non-projectivité dans un corpus peut être utilisé comme un indicateur du degré de liberté de l'ordre des constituants dans la langue en question. D'après les résultats de Havelka (2007), le pourcentage de dépendances non projectives dans les langues à morphologie riche est variable : à titre d'illustration, il est de 0,40 % (pour le bulgare) et de 2,13 % (pour le tchèque). Cependant, ces dépendances relativement peu nombreuses se répartissent sur une portion beaucoup plus importante de phrases : 5 % en bulgare et

23 % en tchèque.

Ce phénomène est important pour le parsing. Plus particulièrement, le besoin de traiter ce type de dépendances augmente la complexité de la tâche, et l'un des deux types de parsers principaux – les parsers par transitions – n'est pas capable d'analyser les dépendances non projectives de manière naturelle. L'utilisation des parsers par transitions peut donc exiger qu'une analyse projective artificielle soit imposée au treebank traité. Cependant, des parsers par transitions modernes disposent souvent des extensions permettant de gérer la non-projectivité (cf. section 3.4).

Dans notre travail, nous adoptons les 6 principes de base cités ci-dessus, mais nous n'imposons pas la non-projectivité à notre corpus : les structures discontinues comme celle de l'exemple 6 sont fidèlement représentées dans le corpus. Ceci nous permet de capter l'une des caractéristiques essentielles du serbe, qui est sa capacité d'admettre des structures syntaxiques discontinues, et c'est également avantageux du point de vue pratique : une telle annotation peut par la suite être transformée en une représentation projective si nécessaire ; l'inverse n'est pas vrai. Le phénomène de non-projectivité dans notre corpus est étudié en détail dans le chapitre 11.

2.3 Jeux d'étiquettes : principes et enjeux

Au-delà du choix du cadre théorique, la constitution d'un treebank présuppose de définir les principes d'annotation du corpus. Une fois que les niveaux d'annotation à apporter au corpus sont déterminés, il faut sélectionner les informations qui seront encodées pour chacun d'entre eux. Ceci est réalisé à travers la construction des jeux d'étiquettes. Un jeu d'étiquettes représente l'ensemble des indications utilisées pour annoter le contenu d'un corpus avec un type d'informations donné. Au niveau morphosyntaxique, les étiquettes expriment les catégories (et possiblement les traits) morphosyntaxiques, alors qu'en syntaxe en dépendances elles correspondent aux fonctions syntaxiques.

Les jeux d'étiquettes sont déterminés en premier lieu par la nature de la langue en question, mais ils sont également soumis au choix des concepteurs du corpus. Ce choix est typiquement guidé par la vocation double des corpus annotés : d'une part,les exploitations en linguistique sont facilitées si le corpus est doté d'informations aussi détaillées que possible. D'autre part, pour les exploitations en TAL, il est essentiel de minimiser la dispersion des données afin de maintenir dans le corpus des conditions propices à l'apprentissage automatique. Lors de la constitution des jeux d'étiquettes, il est donc nécessaire de garantir un équilibre entre la taille du corpus et la taille du jeu d'étiquettes.

2.3.1 Diversité des jeux d'étiquettes morphosyntaxiques

La taille d'un jeu d'étiquettes morphosyntaxiques est déterminée en premier lieu par la richesse de la morphologie flexionnelle de la langue à traiter. Typiquement, les langues à tendance analytique, comme l'anglais, sont traitées avec des jeux d'étiquettes restreints vu qu'il y a relativement peu d'informations à encoder. En revanche, les langues à systèmes flexionnels plus complexes, telles les langues slaves, présentent un défi de ce point de vue. À titre d'illustration, le jeu d'étiquettes standard pour le traitement de l'anglais, celui de PennTreebank, contient 36 tags [10]. Le jeu d'étiquettes tchèque du projet MultextEast en compte 1425.

Ces deux jeux illustrent les deux tendances principales dans la construction des jeux d'étiquettes morphosyntaxiques que l'on peut identifier à travers différents projets. Le jeu de PennTreebank est un jeu à gros grain, dans lequel une étiquette représente de manière générale une catégorie ou une sous-catégorie grammaticale. Ces jeux sont typiquement de taille petite ou moyenne et contiennent en général quelques dizaines de tags. D'autres corpus qui mettent en place ce type de jeu sont, par exemple, NEGRA (Skut et al., 1997), FTBDep (Candito et al., 2010a) ou encore les corpus du projet UD (Nivre et al., 2016). Un extrait du jeu PennTreebank est donné dans le tableau 2.1.

Étiquette	Description
JJ	Adjectif
JJR	Adjectif, comparatif
JJS	Adjectif, superlatif
LS	List item marker
MD	Modal
NN	Nom, singulier ou massif
NNS	Nom, pluriel
NNP	Nom propre, singulier
NNPS	Nom propre, pluriel
PDT	Pré-déterminant

Table 2.1 – Exemple d'un jeu d'étiquettes à gros grain : PennTreebank

Le jeu d'étiquettes tchèque (et c'est également le cas de tous les autres jeux du projet MultextEast) est constitué d'étiquettes positionnelles : une étiquette est une suite de positions associées à un trait morphosyntaxique particulier, et différents codes sont utilisés pour indiquer la valeur des traits. Quelques étiquettes de ce jeu sont données dans le tableau 2.3.

L'objectif de ce type de jeu est de capter un maximum d'informations morphosyn-

10. Nous faisons référence ici aux tags dédiés aux catégories morphosyntaxiques. Le jeu dispose de plusieurs autres étiquettes pour le traitement des caractères spéciaux.

Étiquette	Description
Ncmsg	Type de nom=commun Genre=masculin Nombre=singulier Cas=génitif
Ncmsa–n	Type de nom=commun Genre=masculin Nombre=singulier Cas=accusatif Animé=non
Npmpn–y	Type de nom=propre Genre=masculin Nombre=pluriel Cas=nominatif Animé=oui
Vmip1s-an	Type de verbe=principal Forme verbale=indicatif Temps=présent Personne=première Nombre=singulier Voix=active Négation=non
Vmps-pmay—yn	Type de verbe=principal Forme verbale=participe Temps=passé Nombre=pluriel Genre=masculin Voix=active Négation=oui Animé=oui Clitique=non
Afsmsv—c	Type d'adjectif=qualificatif Degré=superlatif Genre=masculin Nombre=singulier Cas=vocatif Formation=composé
As-fsd	Type d'adjectif=possessif Genre=féminin Nombre=singulier Cas=datif

TABLE 2.3 – Exemple d'un jeu d'étiquettes positionnel : MultextEast

taxiques disponibles. Ils sont donc en général utilisés pour des langues à morphologie flexionnelle riche et ils contiennent typiquement plusieurs centaines d'étiquettes. Des jeux d'étiquettes de ce type ont été définis, à titre d'exemple, pour le français (le tagset GRACE, (cf. Rajman et al., 1997)), ainsi que pour un ensemble de langues de l'Europe centrale et de l'Europe de l'est dans le cadre du projet MultextEast (Erjavec, 2012).

2.3.2 Ajuster la taille d'un jeu d'étiquettes morphosyntaxiques

Les jeux d'étiquettes détaillés engendrent une dispersion des données au niveau morphosyntaxique, et cela a un impact sur les performances des étiqueteurs : des expériences sur le même corpus avec le même étiqueteur font état d'une chute importante des résultats lors du passage d'un jeu à gros grain vers un jeu détaillé (cf. le travail de Ljubešić et al. (2016), analysé dans la section 3.2.7). C'est ce fait qui motive la réduction du jeu d'étiquettes morphosyntaxiques afin d'en obtenir un plus performant dans le cadre du TAL.

Le jeu d'étiquettes anglais de PennTreebank, actuellement considéré comme le jeu standard pour l'étiquetage de l'anglais, est le résultat de ce processus de réduction. Avant la diffusion de PennTreebank, les jeux d'étiquettes morphosyntaxiques utilisés sur l'anglais étaient plus étendus : celui du corpus Brown contenait 87 étiquettes, celui du LOB en avait 135, et celui du London-Lund corpus (LLC) en comptait 197. En constituant PennTreebank, Marcus et al. (1993) ont effectué une réduction systématique du jeu mor-

phosyntaxique du corpus Brown en indiquant explicitement que leur objectif était de faciliter l'apprentissage d'outils automatiques (p. 314). Cependant, cette réduction n'a pas mené à une perte d'informations : les étiquettes éliminées s'appliquaient à des items lexicaux spécifiques, et par conséquent, les distinctions qui ont été perdues au niveau des étiquettes pouvaient être récupérées au niveau lexical.

Malheureusement, une simplification aussi efficace n'est pas toujours envisageable pour les langues à morphologie flexionnelle riche. Par ailleurs, dans le cas de ces langues, l'encodage des traits morphosyntaxiques n'est pas requis seulement pour des études en linguistique théorique. Comme nous l'avons vu sur l'exemple du serbe (cf. chapitre 1), ces informations ont un rôle essentiel dans le décodage des fonctions syntaxiques et, par conséquent, leur présence facilite le parsing (cf. (Ljubešić et al., 2016), analysé dans la section 3.2.7). Il est donc essentiel de trouver une solution optimale pour l'encodage de ces informations.

Une approche possible consiste à limiter le jeu d'étiquettes aux traits morphosyntaxiques qui contribuent au décodage du fonctionnement syntaxique au lieu de chercher l'exhaustivité. Le travail d'Agić et al. (2013a) suit cette option. Ils prennent comme point de départ le jeu d'étiquettes croate du projet MultextEast (660 étiquettes instanciées en corpus) et simplifient d'abord le traitement des formes verbales. À partir de cette première modification, ils proposent trois réductions : 1) sans le trait de définitude des adjectifs ; 2) sans la distinction nom propre *vs* nom commun ; 3) sans les deux traits cités. Sur leur corpus d'environ 87 000 tokens, ces réductions permettent de diminuer la taille du jeu de respectivement 42, 26 et 71 tags. La réduction 3 apporte les meilleurs résultats (p. 53). Le jeu d'étiquettes le plus restreint reste néanmoins beaucoup plus important que les jeux à gros grains : 589 étiquettes sont instanciées dans le corpus.

Une autre stratégie pour préserver les informations morphosyntaxiques nécessaires au parsing est de séparer l'annotation morphosyntaxique en deux niveaux. Un jeu d'étiquettes réduit est utilisé pour encoder les catégories grammaticales. C'est lui qui est utilisé comme base d'apprentissage pour l'étiquetage morphosyntaxique et pour le parsing. Les traits morphosyntaxiques sont encodés séparément, typiquement comme paires *trait=valeur*, et dans le cadre du parsing, ils sont exploités comme traits d'apprentissage supplémentaires, et non pas comme classes statistiques. Ce type d'annotation est préconisé par le format de données CoNLL-X, qui prévoit de représenter séparément les étiquettes des parties du discours à gros grains (*CPOS*), les étiquettes des parties du discours plus spécifiques (*POS*), et les traits morphosyntaxiques (*FEATS*) (cf. Buchholz & Marsi, 2006, p. 151). La même approche est utilisée dans le cadre du projet UD.

Afin d'assurer un encodage optimal des informations morphosyntaxiques dans notre treebank, nous adoptons deux principes. Tout d'abord, nous n'établissons pas un jeu d'étiquettes morphosyntaxiques exhaustif à l'instar du projet MultextEast (Krstev et al.,

2004b) : nous ne retenons que les traits morphosyntaxiques pertinents pour le fonctionnement syntaxique de la langue (cf. section 5.1.1). Ainsi, nous réduisons la quantité d'informations encodées en corpus. Deuxièmement, nous adoptons le principe d'annotation en plusieurs couches : nous représentons séparément les parties du discours, les étiquettes morphosyntaxiques détaillées et les paires individuelles *trait=valeur*. Cela permet de sélectionner la couche d'annotation selon la nature des traitements automatiques envisagés.

2.3.3 Diversité des jeux d'étiquettes syntaxiques

Dans la majorité des treebanks existants, l'annotation syntaxique couvre toutes les formes fléchies constituant la phrase. D'après la Théorie Sens-Texte d'I. Mel'čuk, ce niveau de structure syntaxique, qui s'articule entre tous les lexèmes d'une phrase (y compris les mots fonctionnels) correspond à la structure syntaxique de surface (cf. Mel'čuk, 2009, p. 6-7). À la différence des relations syntaxiques profondes, qui s'articulent entre les lexèmes pleins et qui sont considérées comme indépendantes de la langue (cf. Mel'čuk, 2009, p. 6), les relations syntaxiques de surface sont spécifiques à la langue et doivent être identifiées pour chaque langue individuelle. Ce travail a été fait par Mel'čuk (1995) pour le russe, par Mel'čuk & Pertsov (1987) et Mel'čuk (2003) pour l'anglais, et par Iordanskaja & Mel'čuk (2009) pour les dépendants verbaux en français, mais peu d'autres langues disposent d'un tel inventaire pour guider la constitution des jeux d'étiquettes syntaxiques. Nous avons également mentionné le choix de certains corpus de maintenir une annotation aussi neutre que possible par rapport aux théories syntaxiques (cf. section 2.2), et d'autres projets cherchent à rendre l'annotation syntaxique manuelle plus simple en utilisant un ensemble d'étiquettes minimal (cf. Merkler et al., 2013 ; Erjavec et al., 2010). Tous ces facteurs font que les jeux syntaxiques existants varient aussi bien en taille qu'en relations syntaxiques encodées.

En ce qui concerne le nombre d'étiquettes utilisés, les jeux syntaxiques existants se répartissent sur un continuum. Slovene Dependency Treebank (STD) ne contient que 10 étiquettes (Džeroski et al., 2006), et le treebank croate SETimes en exploite 15 (Agić & Ljubešić, 2014). Au milieu du spectre on trouve FTBDep avec 20 étiquettes provenant de la conversion automatique du corpus en constituants et 8 étiquettes supplémentaires réservées à l'annotation manuelle (Candito et al., 2009). Le projet UD met en place un jeu de base de 37 étiquettes possibles qui peut être enrichi par des étiquettes plus précises (cf. section 2.3.5). Enfin, le treebank SynTagRus illustre le cas des corpus mettant en place une annotation syntaxique détaillée, avec un jeu de 80 étiquettes différentes (Apresjan et al., 2006).

Pour illustrer la diversité des principes d'annotation mis en place, considérons le traitement de la fonction objet (ou plus largement, des dépendants verbaux à statut argumen-

tal) dans différents treebanks. Dans FTBDep, on distingue l'objet direct (obj), deux types d'objet indirect introduits par les prépositions *à* et *de* (respectivement a_obj et de_obj), et l'objet prépositionnel (p_obj), qui correspond à tout autre dépendant prépositionnel à statut argumental (Candito et al., 2009). Dans le cadre du projet UD, les étiquettes obj et iobj sont utilisées respectivement pour l'objet direct et l'objet indirect, alors que tout autre dépendant nominal d'un verbe (indépendamment de son statut argumental) est traité comme dépendant oblique (obl) [11]. Qui plus est, seules les réalisations prototypiques des objets direct et indirect sont admises pour les deux premières étiquettes citées. Enfin, dans PDT, tous les arguments verbaux sont traités avec la seule étiquette Obj.

2.3.4 Taille des jeux d'étiquettes syntaxiques : granularité faible obligatoire ?

La tendance à réduire le jeu d'étiquettes, que nous avons remarquée dans le cadre de l'étiquetage morphosyntaxique, peut également être repérée en parsing. C'est le cas des premiers projets de création de treebank pour le croate et le slovène. Le premier treebank croate (cf. Tadić, 2007 ; Berović et al., 2012) exploite un jeux de 70 étiquettes syntaxiques basé sur celui du corpus tchèque PDT. Or, suite aux remarques des annotateurs humains, relatives au fait que ce jeu était mal adapté au croate, un nouveau jeu de 15 étiquettes a été mis en place (Agić & Merkler, 2013). Quant au slovène, la création du premier treebank pour cette langue a également été basée sur un jeu d'étiquettes adapté à partir de celui de PDT (Džeroski et al., 2006) ; celui-ci a été remplacé par un ensemble de 10 étiquettes lors de la création d'un deuxième treebank slovène (Erjavec et al., 2010).

Cette démarche a eu des effets positifs dans le cas du croate : elle a permis d'améliorer l'accord inter-annotateurs lors de la création du treebank SETimes, et elle a également mené à de meilleurs résultats de parsing (cf. Agić & Merkler, 2013). Cependant, cette réduction importante du jeu d'étiquettes entraîne de nombreuses simplifications difficiles à justifier du point de vue linguistique. Par exemple, dans SETimes.hr (Agić & Ljubešić, 2014), tous les arguments d'un verbe portent l'étiquette Obj sans faire la distinction entre les objets directs et indirects, même si celle-ci est systématiquement marquée au niveau des cas. On y trouve également des étiquettes regroupant des phénomènes très hétérogènes : l'étiquette Atv est utilisée pour annoter les participes, mais aussi les compléments verbaux sous forme d'infinitifs, ainsi que des éléments qui correspondent à l'attribut du sujet et à l'attribut de l'objet direct en français. Cette perte d'information limite l'utilité du corpus non seulement pour des recherches en linguistique, mais aussi pour diverses exploitations en TAL : une distinction telle que celle entre l'objet direct et l'objet indirect peut être précieuse dans le cadre de la traduction automatique ou bien de l'extraction d'événements.

11. http ://universaldependencies.org/u/overview/simple-syntax.html

Or, d'après les résultats de Mille et al. (2012) sur l'espagnol, ce sont précisément les étiquettes qui couvrent des phénomènes trop hétérogènes qui posent le plus de problèmes aux parsers. Le même travail indique également qu'un jeu d'étiquettes syntaxiques plus étendu n'entraîne pas forcément une perte de performances en parsing. Nous analysons les détails de ce travail dans la suite.

Cette étude a été effectuée sur le treebank espagnol Ancora-UPF. Dans le cadre de ce projet, un effort important a été consacré à la constitution du jeu d'étiquettes syntaxiques et à l'examen de l'effet de sa granularité sur les performances en parsing. Tout d'abord, Burga et al. (2011) ont mis en place un système d'identification des fonctions syntaxiques basé sur des critères inspirés de la Théorie Sens-Texte (TST), portant notamment sur les propriétés morphosyntaxiques du gouverneur et du dépendant et sur les caractéristiques de leur ordre linéaire dans la phrase. L'utilisation de ce système a abouti à un ensemble de 70 fonctions syntaxiques. Dans une deuxième étape présentée dans (Mille et al., 2012), ces 70 fonctions ont été transformées en un jeu maximal de 60 étiquettes, avec 3 versions plus restreintes, contenant respectivement 44, 31 et 15 étiquettes. Les 4 versions du jeu ont été utilisées pour évaluer 4 parsers différents : celui de (Che et al., 2009) désigné comme *Che*, celui de (Gesmundo et al., 2009) désigné comme *Merlo*, celui de (Bohnet, 2009) désigné comme *Bohnet*, ainsi que Malt parser de (Nivre et al., 2007b). Il y a donc 16 scénarios d'évaluation au total. Les résultats indiqués dans (Mille et al., 2012) sont repris dans le tableau 2.4.

	LAS				UAS			
	Taille jeu synt.				Taille jeu synt.			
Parser	15	31	44	60	15	44	31	60
(Bohnet, 2009)	84,69	84,28	84,11	81,95	90,27	90,31	90,39	90,49
(Che et al., 2009)	85,11	84,67	84,24	75,14	90,6	90,57	90,37	86,28
(Nivre et al., 2007b)	82,2	82,1	81,9	79,7	87,75	87,83	88	87,91
(Gesmundo et al., 2009)	84,52	84,05	84,53	82,32	-	90,39	90,67	90,11

TABLE 2.4 – Résultats du parsing en fonction de la taille du jeu syntaxique (Mille et al., 2012)

Les résultats montrent que le score LAS [12] diminue de plus de 2 points entre le jeu minimal et le jeu maximal pour les 4 parsers. En revanche, il diminue très légèrement (voire pas du tout) en passant du jeu minimal à celui de 31 tags. Même lors du passage vers le jeu de 44 étiquettes, la diminution la plus importante est de 0,87 pour le parser de Che et al. (2009). En ce qui concerne le score UAS [13], les résultats sont encore plus

12. Labelled Attachment Score : pourcentage des tokens pour lesquels le gouverneur et la fonction syntaxique ont été correctement identifiés (cf. section 3.4).

13. Unlabelled Attachment Score : pourcentage des tokens pour lesquels le gouverneur a été bien iden-

stables : pour les parsers de (Bohnet, 2009) et (Nivre et al., 2007b), la variation maximale à travers les scénarios est de 0,25 et c'est également le cas du parser (Che et al., 2009) sauf pour le scénario avec 60 étiquettes, où il marque la perte la plus importante de toute l'expérience (4,32 points). Pour l'outil de (Gesmundo et al., 2009), la variation maximale est de 0,56.

D'autres corpus utilisant des jeux syntaxiques à granularité forte confortent ces résultats. Le treebank russe SynTagRus utilise environ 80 étiquettes syntaxiques, dont plus de la moitié proviennent des travaux de I. Mel'čuk sur le russe dans le cadre de la TST (Boguslavsky et al., 2002b). Malgré ce nombre d'étiquettes important, les tests de parsing donnent des résultats solides : Malt parser atteint un score LAS de 82,3 points, et un score UAS de 89,1 points (Nivre et al., 2008). Ces résultats sont comparables à ceux obtenus sur d'autres langues slaves dans le cadre de la tâche partagée de CoNNL-X (Buchholz & Marsi, 2006). Il faut néanmoins noter que le corpus est large (environ 400 000 tokens), ce qui facilite l'apprentissage.

Le travail de Agić et al. (2014) sur HOBS 2.0 (*Hrvatska ovisnosna banka stabala*, Croatian Treebank, version 2.0) est un autre exemple des effets positifs d'une augmentation de granularité dans le jeu d'étiquettes syntaxiques. Dans cette deuxième version du corpus, l'étiquette unique utilisée pour tout type de propositions subordonnées a été remplacée par des étiquettes dédiées, de sorte que la taille du jeu est passée de 70 à 81 étiquettes. Une évaluation des deux schémas d'annotation a montré que le nouveau jeu d'étiquettes apportait une amélioration des scores LAS (+3,88 points) et UAS (+2,72 points) (Agić et al., 2014). Cet effet est probablement dû à l'effort d'homogénéiser le traitement des différents types des subordonnées.

Les travaux présentés ci-dessus ont motivé deux principes retenus pour notre travail. Tout d'abord, les résultats mitigés des travaux qui ont réutilisé le jeu d'étiquettes du PDT sur d'autres langues slaves nous ont motivée à abandonner cette piste. Par ailleurs, une telle démarche aurait été contradictoire avec notre souhait de constituer une ressource dédiée au serbe. Nous optons donc pour la création d'un nouveau jeu d'étiquettes syntaxiques. Quant aux principes de constitution de ce jeu, nous prenons en compte les implications du travail de Burga et al. (2011) et de Mille et al. (2012) : nous n'essayons pas de minimiser le nombre d'étiquettes syntaxiques à tout prix ; nous cherchons plutôt une bonne expressivité au niveau syntaxique, avec des distinctions basées sur des critères de surface. Nous visons également une structuration du jeu qui permet une conversion facile vers des jeux moins étendus selon les besoins.

---

tifié, sans prendre en compte l'identification de la fonction (cf. section 3.4).

2.3.5 Jeu d'étiquettes syntaxiques Universal Dependencies

Étant donné la popularité croissante du projet UD, le jeu d'étiquettes qu'il propose mérite d'être abordé plus en détail. En effet, comme mentionné dans la section 2.1, ce projet compte désormais plus de 100 treebanks et la campagne d'évaluation CoNLL2017 a été entièrement dédiée au parsing basé sur ce formalisme [14]. Nous présentons dans la suite ses principales propriétés.

Comme nous l'avons déjà dit, le projet UD a pour objectif la création d'un ensemble de corpus de différentes langues qui partagent les mêmes principes d'annotation. Ainsi se constitue une base de données linguistiques directement comparables à travers les langues. Pour ce faire, le projet définit des inventaires d'étiquettes pour différents niveaux d'annotation, censées permettre la description linguistique de toute langue ; les auteurs d'un treebank particulier puisent ensuite dans ces répertoires pour sélectionner les étiquettes pertinentes pour la langue traitée.

Au niveau morphosyntaxique, l'annotation est divisée en deux couches : pour l'annotation des parties du discours, un jeu de 17 étiquettes est proposé (adjectif, adverbe, nom, adposition, pronom, etc.), alors que le jeu pour la description des traits morphosyntaxiques fins contient 48 traits différents, qui peuvent être lexicaux (type de pronom, type de numéral, etc.), nominaux (genre, nombre, définitude, etc.) ou verbaux (mode, temps, polarité, etc.). Les valeurs possibles de ces traits sont également définies par le projet.

Quant à l'annotation syntaxique, le projet propose un ensemble de 37 étiquettes de base, communes à tous les treebanks. Pour accommoder les spécificités des langues individuelles, un deuxième ensemble d'étiquettes est mis en place : il s'agit d'une sous-catégorisation des étiquettes basiques, ce qui permet donc une qualification plus fine des relations syntaxiques. Il existe 198 étiquettes fines, mais en général seul un petit sous-ensemble est utilisé dans une langue donnée. À titre d'illustration, 17 étiquettes de ce type sont utilisées pour l'annotation du français.

L'uniformisation de l'analyse syntaxique pour l'ensemble des langues représentées dans le projet introduit des contraintes fortes. Nous avons déjà évoqué le traitement des dépendants verbaux dans la section 2.3.3 ; la décision de n'annoter comme objet direct ou indirect que les réalisations prototypiques de ces dépendants vient de la volonté d'harmoniser la représentation de ces éléments à travers les différentes langues. Cependant, cela signifie que la phrase française *Donne leur les jouets* contient un objet indirect (exprimé par le pronom au datif), alors que la phrase *Donne les jouets aux enfants* est plutôt dotée d'un dépendant oblique du verbe, vu que la forme *enfants* est introduite par une préposition.

14. Un descriptif de la campagne est disponible à l'adresse suivante : `http://universaldependencies.org/conll17/`.

Afin d'établir un parallèle entre les langues à cas et les langues qui n'en disposent pas, on introduit un traitement particulier pour les prépositions : elles sont considérées (tout comme les postpositions) comme marqueurs de cas et annotées comme des dépendants des noms qu'elles introduisent. Par conséquent, dans une phrase comme *Je donne le livre à Pierre*, le verbe *donner* est considéré comme le gouverneur du nom *Pierre*, qui gouverne à son tour la préposition *à*.

Un autre exemple concerne le traitement des phrases à copule. Pour avoir des traitements comparables entre les langues qui ont des verbes copules et celles qui n'en ont pas (cf. le russe ou l'arabe), on considère que la racine de ce type de phrase est la forme introduite par la copule. Autrement dit, dans la phrase *Pierre est honnête*, l'adjectif *honnête* est annoté comme la racine, et il gouverne le nom *Pierre* en tant que sujet, et le verbe *est* en tant que copule.

Ces deux derniers exemples illustrent l'un des principes de base du projet UD : la primauté des mots lexicaux par rapport aux mots fonctionnels. Comme mentionné ci-dessus, ce mécanisme est adopté dans un souci d'assurer un traitement universel entre différentes langues, et il permet effectivement de lisser certaines différences. Nous constatons cependant qu'il peut mener à des règles d'annotation qui sont contraires à la tradition linguistique d'une langue donnée. Qui plus est, comme le soulignent Groß & Osborne (2015), ce principe va à l'encontre de la majorité des travaux en syntaxe théorique, où la position prépondérante est que les mots fonctionnels gouvernent les mots lexicaux (cf. Pollard & Sag, 1994 ; Bresnan, 2001 ; Chomsky, 1995, 1993 ; Hudson, 1984 ; Mel'čuk, 1988). Au-delà de ce statut problématique du point de vue théorique, Groß & Osborne (2015) indiquent également des situations problématiques en corpus, parmi lesquelles la représentation des structures à verbes supports, de la négation phrastique et de l'ellipse du groupe verbal en anglais (*ibid*, p. 112-115). Du fait que cette représentation s'appuie sur des relations entre les mots lexicaux, elle peut également être interprétée comme plus proche d'une structure sémantique que syntaxique, notamment si l'on considère une syntaxe de surface telle que définie dans la théorie TST (Mel'čuk, 1988).

Le nombre de langues qui participent au projet UD semble démentir ces critiques : elle sont plus de 60 à avoir été annotées en utilisant ce jeu d'étiquettes. Toutefois, nous avons décidé de ne pas l'adopter dans le cadre de cette thèse. Cette décision est motivée en premier lieu par le fait que ce formalisme a des effets concrets sur la représentation de divers phénomènes dans le corpus. Plus particulièrement, ces principes d'annotation divergent de manière importante de la tradition grammaticale serbe, qui favorise les têtes fonctionnelles. Nous ne les considérons donc pas adaptés à l'annotation de notre corpus. Pour rappel, notre treebank devait devenir la première ressource de ce type pour cette langue. Il était donc justifié d'adopter un schéma d'annotation spécifique à cette langue.

2.4 Qualité de l'annotation manuelle

L'utilité d'un corpus dépend directement de la qualité de l'annotation manuelle : si elle est mauvaise, cela affecte l'apprentissage des outils automatiques, les résultats d'évaluation, mais aussi la ré-utilisabilité du corpus. Or, comme le rappelle Fort (2012, p. 49), il n'est pas possible de mesurer automatiquement la validité d'une annotation manuelle (son caractère vrai ou faux) ; on doit se contenter d'en évaluer la fiabilité, autrement dit, la cohérence avec laquelle elle a été réalisée. Une fois les jeux d'étiquettes définis, il est donc nécessaire de s'assurer que les étiquettes retenues soient utilisées de manière systématique à travers le corpus. Des méthodes ont été définies pour accompagner ce processus, qui incluent : les schémas et guides d'annotation (section 2.4.1), l'annotation manuelle redondante (section 2.4.2) et l'évaluation de l'accord inter-annotateurs (section 2.4.3).

2.4.1 Schémas et guides d'annotation

Un schéma d'annotation correspond à l'ensemble des règles d'application d'un jeu d'étiquettes. Un guide d'annotation est un document qui exprime le schéma d'annotation dans un format destiné à des annotateurs humains. Il contient donc des instructions détaillées pour l'utilisation de chaque étiquette, et notamment pour le traitement des cas de figure problématiques. Les annotateurs humains sont censés maîtriser les guides et s'y reporter systématiquement au cours de l'annotation afin de vérifier leurs décisions. Cela permet de garantir la cohérence entre différents annotateurs (accord inter-annotateurs), mais aussi entre les annotations produites par le même annotateur tout au long du projet (accord intra-annotateur).

Un exemple de guide d'annotation disponible est celui de PDT [15]. Ce document d'environ 300 pages indique les principes globaux d'annotation (structure des arbres, attributs des nœuds, lien avec l'annotation morphosyntaxique) et donne des instructions détaillées pour l'utilisation de toutes les étiquettes du jeu syntaxique, en les accompagnant de nombreux exemples. Hajič (2005) précise que la constitution du guide a été un processus long et cyclique : les règles posées étaient modifiées et complétées en continu selon le matériel rencontré dans les données.

Vu l'effort nécessaire pour constituer un tel document, il pouvait être bénéfique d'adopter un guide d'annotation déjà existant, tel que celui du projet PDT ou encore celui du projet UD. Cependant, nous avons déjà détaillé les raisons pour lesquelles nous ne le faisons pas (cf. sections 2.3.4 et 2.3.5). Par conséquent, nous avons rajouté la création des guides d'annotation parmi les tâches à réaliser dans cette thèse.

15. `https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/index.html`

2.4.2 Annotation manuelle redondante

Une autre manière d'assurer la qualité de l'annotation manuelle est de mettre en place des annotations redondantes. Dans ce cadre de travail, plusieurs annotateurs traitent le même contenu indépendamment les uns des autres. Cela permet de confronter leurs productions par la suite et de détecter les points de divergence. Ces divergences doivent ensuite être étudiées et résolues pour établir l'annotation finale du corpus.

Différents scénarios de travail concrets peuvent être envisagés. Par exemple, dans la constitution de PDT, l'annotation morphosyntaxique était faite entièrement en double, et les divergences ont été résolues par un troisième annotateur (Hajič, 2005). Dans le cas du treebank NEGRA, tous les niveaux d'annotation ont été faits en double, et les divergences étaient résolues par un consensus des deux annotateurs travaillant en parallèle (Brants et al., 2003). Enfin, dans le cas de FTB, l'annotation morphosyntaxique était effectuée par un annotateur et validée par un deuxième (Abeillé et al., 2003).

Ces démarches assurent effectivement une meilleure cohérence des annotations, et permettent par ailleurs d'identifier les points problématiques récurrents, ce qui peut mener à une amélioration des guides d'annotation. En revanche, elles présentent un désavantage pratique important : elles augmentent le temps nécessaire pour réaliser l'annotation du corpus.

Comme la durée de ce projet était fortement contrainte, une annotation redondante complète était difficile à réaliser. Nous l'avons cependant intégrée ponctuellement dans le processus d'annotation, ce qui nous a permis de faire des évaluations de l'accord inter-annotateurs dans le but d'évaluer les guides et les annotateurs humains.

2.4.3 Accord inter-annotateurs

La cohérence des annotations peut être explicitement mesurée en évaluant l'accord inter-annotateurs. Cette mesure exprime le taux d'accord entre deux (ou plusieurs) annotateurs traitant le même contenu indépendamment l'un de l'autre. Il peut être basé sur un simple pourcentage d'annotations concordantes ou bien sur des mesures plus complexes, telles les mesures de la famille *kappa*, dont le *kappa* de Cohen. Le taux d'accord inter-annotateurs est typiquement exploité de deux manières principales : au début d'un projet, cette mesure peut être utilisée pour évaluer la qualité du guide d'annotation, alors que dans des étapes ultérieures elle permet d'estimer la cohérence des annotations dans le corpus. Il peut être calculé entre différents annotateurs, ou entre un annotateur et l'annotation de référence. Ce dernier mode d'utilisation permet notamment d'évaluer la production d'un annotateur.

À titre d'illustration, Marcus et al. (1993) indiquent que le degré de désaccord au niveau des annotations morphosyntaxiques dans PennTreebank est de 4,1 % entre annotateurs, et

de 4 % en moyenne entre les annotateurs et l'annotation de référence, typiquement établie à travers une annotation redondante. Dans le cas de PDT, le désaccord entre les annotateurs est de 3 %, et celui entre les annotateurs et la référence varie entre 1 % et 5 % (Hajič, 2005). Il est intéressant de remarquer que le jeu d'étiquettes morphosyntaxiques décrit dans ce travail contient 4712 étiquettes différentes (idem Hajič, 2005, p. 56), alors que le jeu de PennTreebank en contient 36. La cohérence de l'annotation du corpus tchèque n'a donc pas été impactée par la taille du jeu d'étiquettes. Le corpus NEGRA signale un taux d'accord en morphosyntaxe (avec 54 étiquettes) de 98,6 % (Brants, 2000a). Pour le même treebank, l'accord inter-annotateurs au niveau des structures syntaxiques est de 92,4 %.

Le *kappa* de Cohen (Carletta, 1996) prend en compte l'accord relatif des annotateurs et la probabilité d'un accord aléatoire. Si l'accord des annotateurs est parfait, la valeur de *kappa* est égale à 1, et s'ils sont complètement en désaccord ou en accord uniquement dû au hasard, le *kappa* est proche de 0. L'interprétation des valeurs intermédiaires est une question plus complexe. Plusieurs échelles ont été définies dans ce but (cf. Krippendorff, 1980 ; Green, 1997 ; Landis & Koch, 1977) ; nous reprenons ici celle de Landis & Koch (1977) (cf. tableau 2.5).

Valeur de *kappa*	Force de l'accord
<0,00	Mauvais
0,00-0,20	Faible
0,21-0,40	Médiocre
0,41-0,60	Modéré
0,61-0,80	Important
0,81-1,00	Quasi-parfait

TABLE 2.5 – Échelle de valeurs de *kappa* de Cohen définie dans (Landis & Koch, 1977)

À titre d'illustration, cette mesure a été utilisée par Urieli (2013) pour évaluer l'accord inter-annotateurs sur le corpus FrWikiDisc, qui contient des discussions Wikipédia. L'accord a été calculé sur les dépendances labellisées du corpus. Le *kappa* de Cohen était de 0,86 entre les deux annotateurs, et les annotateurs avaient atteint respectivement un taux de 0,97 et 0,88 par rapport à l'annotation de référence.

Malgré des critiques récentes et des propositions d'autres mesures (cf. Artstein & Poesio, 2008), le *kappa* de Cohen reste une mesure standard communément utilisée pour l'évaluation de l'accord inter-annotateurs (cf. Urdu treebank (Bhat & Sharma, 2012a), Hinoki treebank du japonais (Bond et al., 2008), treebank EPEC du basque (Uria et al., 2009), Prague Discourse Treebank (Poláková et al., 2014), treebanks HOBS et SETimes.hr du croate (Agić & Merkler, 2013)). Nous l'avons également mise en œuvre dans le cadre d'une évaluation de la qualité des guides d'annotation et des annotateurs humains (cf.

section 7.5).

2.5 Minimiser le temps nécessaire : exploitation des outils du TAL existants

Étant donné la quantité de travail nécessaire pour constituer un treebank, il est fortement souhaitable d'optimiser l'annotation manuelle en la rendant aussi aisée et rapide que possible. Des jeux d'étiquettes bien sélectionnés et des guides d'annotation ergonomiques y contribuent. Une autre méthode explicitement orientée vers cet objectif consiste à exploiter les outils de TAL existants pour réaliser une préannotation automatique. Comme le montrent les résultats des expériences de Fort (2012, p. 148-149) sur ce sujet, même une préannotation de qualité médiocre facilite la tâche des annotateurs humains et accélère la validation du corpus. La mise en place concrète varie d'un projet à l'autre et dépend fortement de la disponibilité des outils. Elle peut faire appel à des outils à base de règles, parfois créés *ad hoc* pour les besoins du projet (cf. Boguslavsky et al., 2002a ; Hajič, 2005). Alternativement, elle peut se baser sur la méthode de *bootstrapping*, introduite par Breiman (1996). Le *bootstrapping* consiste à produire un corpus d'apprentissage minimal sur lequel un premier modèle de traitement est entraîné. Ce modèle est ensuite utilisé pour préannoter le reste des données et accélérer ainsi la validation du corpus complet. L'outil peut ensuite être ré-entraîné sur le corpus entier. Cette méthode est applicable à tout outil basé sur l'apprentissage automatique.

Le lien positif entre la préannotation automatique et la vitesse de l'annotation manuelle est démontré d'une manière explicite dans (Marcus et al., 1993) sur PennTreebank. Au niveau morphosyntaxique, l'étiqueteur stochastique de Church (1988) a été utilisé au début du projet pour être ensuite remplacé par un système en cascade de différents outils développés dans le cadre du projet. La préannotation est validée par les annotateurs humains dans sa totalité.

Une évaluation explicite des deux modes d'annotation est réalisée : on compare le mode *correction* (basé sur la préannotation automatique) avec le mode *étiquetage de zéro*, où les annotateurs traitent le texte brut. Les résultats étaient clairs : avec la préannotation, les annotateurs mettent en moyenne 20 minutes pour valider 1000 tokens, alors que sans elle, la même tâche leur prend en moyenne 44 minutes. La préannotation a également un effet positif sur la qualité de l'annotation : le taux de désaccord brut entre les annotateurs est de 7,2 % sans la préannotation, et de 4 % avec elle. Cela indique que le recours au prétraitement automatique simplifie la tâche des annotateurs humains et leur permet à la fois d'être plus efficaces et de produire des annotations de meilleure qualité.

D'autres projets de constitution de treebank ont exploité des méthodes comparables,

sans avoir cherché à évaluer spécifiquement leur apport. Le projet NEGRA met en place un système d'algorithmes basés sur les modèles de Markov cachés (HMM, voir la section 3.2.1) en cascade, qui produit une préannotation aussi bien pour la morphosyntaxe que pour la structure des constituants et les labels fonctionnels. Comme il s'agit de modèles par apprentissage automatique, un entraînement initial minimal des systèmes avait été nécessaire. Le système s'est cependant montré très efficace, permettant aux annotateurs humains expérimentés d'atteindre la vitesse de 1300 tokens par heure, tous niveaux d'annotation confondus.

Le projet de constitution de FTB a eu recours à plusieurs outils développés dans le cadre du projet même. L'étiquetage morphosyntaxique était réalisé avec un outil majoritairement basé sur des règles écrites manuellement, couplé à un lexique externe d'environ 360 000 formes fléchies (Reyes, 1997 ; Abeillé et al., 1998). Après la correction manuelle des étiquettes morphosyntaxiques, les lemmes étaient ajoutés automatiquement à partir du lexique. Les indications des parties du discours permettaient de résoudre la majorité des ambiguïtés dans le lexique, et le reste était résolu manuellement. Sur ce segment de travail, la vitesse des annotateurs humains était de 500 tokens/h.

La préannotation syntaxique était effectuée en deux passes. L'identification des constituants était faite avec un parser à base de règles (Clement & Kinyon, 2000) puis corrigée manuellement. Les fonctions des constituants étaient identifiées avec un tagger basé sur des règles (Abeillé & Barrier, 2004), également corrigées manuellement par la suite. Malheureusement, il n'y a pas d'indication de vitesse pour ces niveaux d'annotation dans (Abeillé et al., 2003).

La création du treebank russe SynTagRus s'est basée sur des préannotations morphosyntaxique et syntaxique effectuées par les modules dédiés du moteur de traduction automatique ETAP-3 de Apresjan et al. (1992, 1993). Il s'agit d'un système de traduction anglais-russe symbolique, basé sur des descriptions des deux langues, élaborées d'après la Théorie Sens-Texte. La préannotation a été corrigée manuellement dans sa totalité (cf. Boguslavsky et al., 2002a).

Dans l'annotation morphosyntaxique de PDT, une préannotation à l'aide d'un dictionnaire morphologique était effectuée : à chaque token, tous les lemmes et toutes les étiquettes morphosyntaxiques possibles étaient attribués. La tâche des annotateurs consistait donc à effectuer la désambiguïsation du traitement automatique. Quant au parsing, une première analyse était effectuée en utilisant un script à base de règles créé *ad hoc*, qui ignorait explicitement des contextes complexes. Les annotateurs humains ont signalé cependant que la qualité de cette préannotation était satisfaisante (environ 80 % de fonctions assignées étaient correctes) (cf. Hajič, 2005).

D'autres travaux démontrent l'utilité de cette démarche pour différents niveaux d'an-

notation, par exemple (Xue et al., 2005), (Fort & Sagot, 2010) et (Tellier et al., 2014).

Il faut néanmoins noter que l'utilisation d'une préannotation peut introduire un biais : Fort (2012, p. 144) constate que la présence d'une préannotation peut mener à la propagation des erreurs faites par l'outil dans la production de l'annotateur humain. L'effet positif sur la qualité globale de l'annotation manuelle n'est cependant pas contesté. Par conséquent, nous adoptons cette approche, que nous qualifierons d'outillée, pour la constitution de notre treebank, et ceci pour les trois niveaux d'annotation à apporter (l'étiquetage morphosyntaxique, la lemmatisation, l'annotation syntaxique). Cette méthode implique donc la nécessité de se doter d'outils adaptés ; nous abordons cette question en détail dans le chapitre 3.

2.6 Organisation de la campagne : annotation agile

Dans le chapitre 4 de sa thèse consacrée à la méthodologie d'annotation de corpus (Fort, 2012), K. Fort identifie 4 stades d'une campagne d'annotation : 1) travail préparatoire, 2) pré-campagne, 3) annotation, et 4) finalisation. Le travail préparatoire est dédié à l'identification des acteurs, à une exploration initiale du corpus (afin d'identifier les échantillons adaptés pour l'entraînement des annotateurs et d'envisager le schéma d'annotation) et à la création et modification des guides d'annotation. La pré-campagne comprend la mise au point d'un échantillon de référence et la formation des annotateurs. Le stade d'annotation proprement dite commence par une période de rodage des annotateurs, suivi du véritable travail d'annotation. Enfin, la finalisation porte sur les activités nécessaires pour publier le corpus. On remarque que cette méthode est principalement linéaire : les différentes phases s'enchaînent l'une après l'autre. Fort évoque néanmoins le fait que des modifications des guides sont encore nécessaires dans la phase de rodage, mais les guides sont stabilisés après cette étape (Fort, 2012, p. 81).

Une autre approche de l'organisation de la campagne d'annotation préconise une démarche itérative. Il s'agit de la méthode d'annotation agile, évoquée par ailleurs par K. Fort (Fort, 2012, p. 61-62). Cette méthode a été définie par Voormann & Gut (2008) et appliquée par Alex et al. (2010). Elle préconise une organisation itérative du processus d'annotation, dans laquelle chaque cycle d'annotation est suivi d'une évaluation de l'accord inter-annotateurs et de modifications du guide d'annotation. Ainsi, on vérifie la qualité de l'annotation manuelle. Qui plus est, on garantit plus de flexibilité aux guides d'annotation, en permettant de traiter les problèmes rencontrés durant l'annotation.

Bien que l'apport exacte de cette méthode par rapport à une organisation linéaire n'ait pas encore été évalué dans le cadre d'un travail pratique, les avantages potentiels de l'annotation agile nous semblent importants. Par conséquent, nous avons basé notre

projet sur cette approche (cf. section 4.2).

2.7 Principes retenus

La présentation des différents aspects de la création des treebanks nous a permis d'identifier les principes sur lesquels notre projet est basé. Nous les résumons ci-dessous.

Cadre théorique : syntaxe en dépendances. La syntaxe en dépendances permet une représentation simple des structures syntaxiques discontinues dont le serbe est doté. Elle est également adoptée dans la majorité des treebanks de langues à ordre de constituants flexible, et son utilisation devient de plus en plus répandue. Pour toutes ces raisons, nous retenons ce cadre théorique pour la création de notre treebank.

Annotation morphosyntaxique : encodage d'informations optimisé. À la fin du chapitre 1, nous avons constaté qu'il était nécessaire d'inclure les informations morphosyntaxiques fines dans notre treebank. Afin d'en garantir l'exploitation optimale, nous adoptons les deux principes suivants : 1) nous visons l'optimisation de la quantité de données encodées en ne retenant que les traits morphosyntaxiques pertinents au niveau du fonctionnement syntaxique, et 2) nous adoptons l'annotation en plusieurs couches, séparant ainsi les étiquettes des parties du discours, les étiquettes détaillées et les traits morphosyntaxiques fins. Ceci facilite la sélection de la couche appropriée dans les applications en TAL.

Annotation syntaxique : constitution d'un nouveau jeu propre au serbe. Guidée par notre souhait de créer une ressource qui reflète les spécificités du serbe, nous avons renoncer à exploiter des jeux d'étiquettes déjà existants, comme celui de PDT ou celui du projet UD. Pour le jeu PDT, cette décision est par ailleurs corroborée par les résultats mitigés des expériences de ce type sur le croate et le slovène (cf. section 2.3.4). Nous ne posons pas de contraintes au niveau de la taille du jeu, mais la définition des étiquettes est guidée par des critères de surface dans l'objectif de garantir leur accessibilité au parser.

Qualité de l'annotation manuelle : importance des guides d'annotation. Le fait que nous définissons un nouveau jeu d'étiquettes syntaxiques signifie que nous devons également constituer le guide d'annotation correspondant. Ceci est également vrai pour l'annotation morphosyntaxique. Pour assurer leur qualité, nous retenons les évaluations de l'accord inter-annotateurs comme un moyen de détecter et combler des lacunes. En revanche, nous abandonnons l'annotation redondante systématique du fait de son coût élevé.

Rapidité du processus d'annotation : approche outillée. Les contraintes temporelles ont également déterminé un autre aspect de notre méthode : nous adoptons pleinement l'approche outillée présentée dans la section 2.5. Nous exploitons une préannota-

tion automatique pour les trois couches d'annotation à réaliser (l'annotation morphosyntaxique, la lemmatisation et l'annotation syntaxique).

Méthode d'annotation globale : annotation agile. Pour maximiser la qualité de l'annotation manuelle, nous adoptons une organisation du travail inspirée de la méthode agile. Par conséquent, nous modifions le stade 3 tel que défini par Fort (2012) : le corpus à annoter est divisé en échantillons de taille comparable, traités tour à tour, et chaque cycle d'annotation se clôt par une étape visant à capitaliser le travail réalisé. L'élément le plus important de cette étape est le retour d'expérience des annotateurs, qui permet d'adapter les guides d'annotation si nécessaire. Cette organisation est détaillée dans le chapitre 4.

Ainsi, nous avons identifié les principes sur lesquels notre méthode est basée. Il nous manque cependant un élément : le fait d'adopter une approche outillée signifie qu'il faut se doter d'outils. Le chapitre 3 est dédié à la sélection d'outils adaptés à nos besoins.

Chapitre 3

Outils d'analyse automatique

Comme nous l'avons vu dans la section 2.5, les outils automatiques peuvent être exploités pour faciliter la constitution des corpus annotés. Néanmoins, ils sont en premier lieu destinés à l'annotation autonome de grandes quantités de données, que ce soit pour alimenter des recherches linguistiques ou pour faciliter différentes autres tâches en TAL comme l'identification des rôles sémantiques, l'extraction d'information, l'analyse du discours, etc. Dans ce cadre d'utilisation, l'annotation produite n'est pas systématiquement vérifiée et doit être exploitée en prenant en compte un certain taux d'erreur (cf. à titre d'exemple Ljubešić & Klubička, 2014). Dans le cadre de cette thèse, les outils sélectionnés seront utilisés selon les deux modalités : ils seront exploités pour créer des modèles de traitement automatique destinés à une annotation autonome, mais ils seront également utilisés en tant qu'outils de prétraitement pour faciliter l'annotation manuelle de notre treebank. Comme notre méthode d'annotation prévoit une démarche itérative, elle sous-entend plusieurs cycles d'entraînement et d'évaluation des outils. Il est donc essentiel d'identifier les outils les plus performants pour maximiser la qualité et l'utilité de l'annotation produite, mais il est tout aussi nécessaire qu'ils soient rapides et faciles d'utilisation pour assurer un bon déroulement de la campagne d'annotation. Étant donné la diversité des méthodes disponibles pour différents niveaux de traitement, cette question n'est pas simple.

Les premiers outils du traitement automatique du langage reposaient sur des méthodes symboliques, mettant en place des systèmes de règles de traitement manuellement construits par des experts humains. Cependant, cette approche implique la création d'un outil pour chaque niveau de traitement et pour chaque langue traitée, ce qui a évidemment un coût rédhibitoire. Pour dépasser ce problème, depuis la fin des années 1990, les efforts dans le domaine du TAL portent majoritairement sur les méthodes statistiques basées sur l'apprentissage automatique supervisé. Ces approches ont été appliquées avec succès à différents niveaux de traitement (étiquetage morphosyntaxique, parsing, extraction d'opinion, systèmes des questions-réponses, traduction automatique).

Jusqu'à récemment, les méthodes de modélisation linéaire étaient clairement l'approche dominante dans le TAL, avec l'utilisation massive d'algorithmes basés sur le modèle de Markov caché (cf. Merialdo, 1994 ; Brants, 1996, 2000b ; Halácsy et al., 2007), le perceptron (cf. Collins, 2002 ; Gesmundo & Samardžić, 2012), les machines à vecteurs de support (cf. Giménez & Marquez, 2004 ; Kudo & Matsumoto, 2002 ; Yamada & Matsumoto, 2003) ou encore la régression logistique (cf. Ratnaparkhi, 1996 ; Toutanova et al., 2003 ; Denis & Sagot, 2009 ; Chrupała et al., 2008). Bien qu'elles aient contribué à des avancées importantes dans différents domaines du TAL, ces méthodes partagent un point faible : elles sont affectées par la nature éparse des données linguistiques. Autrement dit, les phénomènes rares (ou sous-représentés dans les données d'apprentissage) ne sont pas acquis d'une manière satisfaisante par ces outils. Depuis 2014, un nouveau type d'algorithmes permet de s'affranchir de cette limitation : les réseaux de neurones. Comme indiqué par Goldberg (2017, p. xvii), ces algorithmes, et notamment les réseaux de neurones récurrents (*Recurrent Neural Networks* ou RNNs en anglais), sont capables d'exploiter des chaînes d'entrée de longueur arbitraire et de produire des représentations denses des données traitées. Grâce à ces spécificités, les outils basés sur des approches neuronales atteignent des résultats impressionnants en étiquetage morphosyntaxique (Ling et al., 2015 ; Plank et al., 2016), parsing (Chen & Manning, 2014 ; Dyer et al., 2015 ; Kiperwasser & Goldberg, 2016), modélisation du langage (Adel et al., 2013) ou traduction automatique (Cho et al., 2014 ; Sundermeyer et al., 2014).

La suite de cette section sera consacrée aux trois niveaux de traitement sur lesquels porte cette thèse : l'étiquetage morphosyntaxique, la lemmatisation et le parsing. Dans la section 3.1, nous présentons les spécificités de ces trois tâches et la manière dont elles s'intègrent dans une chaîne de traitement typique en TAL. Nous proposons ensuite un tour d'horizon rapide des outils existants pour chacune d'entre elles, en accordant une attention particulière aux travaux effectués sur le serbe, ce qui nous permet de sélectionner ceux qui seront inclus dans notre méthode d'annotation de corpus (cf. sections 3.2 pour l'étiquetage, 3.3 pour la lemmatisation et 3.4 pour le parsing). Enfin, nous résumons les points essentiels dégagés (cf. section 3.5).

3.1 Chaîne de traitement : principes

Les outils basés sur l'apprentissage automatique supervisé ont deux modes de fonctionnement : l'entraînement (ou l'apprentissage) et l'annotation (ou l'analyse). Durant l'**entraînement**, ils parcourent un **corpus d'apprentissage**[1] annoté et utilisent leur module statistique afin de dériver les probabilités de différents phénomènes rencontrés

1. Dans le cas de la lemmatisation, il peut également s'agir d'un lexique (cf. section 3.3.2).

en corpus. L'ensemble de ces probabilités représente le **modèle** de traitement. Durant l'**annotation**, ces outils font appel au modèle créé pour annoter de nouveaux textes. Grâce à cette architecture, ils sont indépendants de la langue : le même outil peut apprendre à traiter toute langue, pourvu qu'il existe des ressources d'entraînement adaptées. La tâche concrète à effectuer diffère cependant d'un niveau de traitement à l'autre. Nous proposons dans ce qui suit une présentation rapide pour les trois niveaux retenus.

3.1.1 Tâche d'étiquetage morphosyntaxique

L'étiquetage morphosyntaxique consiste à attribuer à chaque forme fléchie du corpus ses propriétés morphosyntaxiques. Le niveau de détail apporté par cette annotation est variable : il peut s'agir d'une simple identification des parties du discours (angl. ***part-of-speech (POS) tagging***), ou bien d'une analyse plus poussée, qui vise à identifier également les valeurs de différents traits morphosyntaxiques (cf. section 2.3). Dans le cadre de cette thèse, nous appellerons ce deuxième type d'annotation **étiquetage morphosyntaxique fin**. Une illustration des deux types d'annotation est donnée dans le tableau 3.2 [2]. Les outils dédiés à ce niveau de traitement sont appelés **étiqueteurs** ou **taggers**.

Token	POS tagging	(Signification)	Etiq. détaillé	(Signification)
Filip	PROPN	(nom propre)	Npmsny	(nom propre masculin au nominatif singulier)
studira	VERB	(verbe)	Vmip3s-annp	(verbe principal à la troisième personne singulier du présent)
lingvistiku	NOUN	(nom)	Ncfsan	(nom commun féminin à l'accusatif singulier)
u	ADP	(adposition)	Spsl	(préposition complétée par le locatif)
Italiji	PROPN	(nom propre)	Npfsln	(nom propre féminin au locatif singulier)

TABLE 3.2 – Deux types d'étiquetage morphosyntaxique pour la phrase *Filip studira lingvistiku u Italiji* 'Filip étudie la linguistique en Italie'

L'étiquetage morphosyntaxique est une des tâches de base en TAL, utile à de nombreuses applications car il permet de résoudre une partie des phénomènes d'ambiguïté ; dans les corpus destinés aux recherches linguistiques, il facilite la formulation de requêtes plus globales ; il permet aux parsers de faire des généralisations par rapport au comportement de différentes catégories grammaticales, etc.

2. Le POS tagging est fait en utilisant le jeu d'étiquettes du projet UD, et l'étiquetage détaillé avec celui de MultextEast. Pour des descriptions détaillées, voir respectivement les sections 2.3.5 et 2.3.1.

3.1.2 Tâche de lemmatisation

La lemmatisation consiste à identifier la forme canonique de chaque token du corpus. En serbe, on considère que la forme canonique des noms est le nominatif singulier, celle des adjectifs est le nominatif singulier masculin, et celle des verbes est l'infinitif. Une illustration en est donnée dans le tableau 3.3.

Token	Lemme
Filip	Filip
studira	studirati
lingvistiku	lingvistika
u	u
Italiji	Italija

TABLE 3.3 – Lemmatisation de la phrase *Filip studira lingvistiku u Italiji* 'Filip étudie la linguistique en Italie'

Les outils dédiés à cette tâche s'appellent des **lemmatiseurs**. Tout comme l'étiquetage morphosyntaxique, la lemmatisation permet de formuler des requêtes plus générales dans des corpus annotés, et elle est également exploitée dans le cadre de la recherche d'information. Cependant, pour le périmètre de cette thèse, son principal intérêt réside dans le fait qu'elle augmente la densité des données au niveau lexical et facilite ainsi la tâche des parsers (cf. section 1.3.2).

3.1.3 Tâche de parsing

Le parsing consiste à déterminer la structure syntaxique d'une phrase, mais les particularités de la tâche dépendent du cadre théorique adopté. Dans le cadre d'une **analyse en constituants** (cf. section 2.2.1), le résultat du parsing est généralement représenté sous la forme d'un texte tabulé, chaque niveau de tabulation correspondant à un niveau de l'arbre syntaxique. Les constituants sont délimités à l'aide de parenthèses, et les étiquettes syntaxiques sont indiquées à l'intérieur de la parenthèse ouvrante du constituant en question (cf. figure 3.1a). Dans le cadre d'une **analyse en dépendances** (cf. section 2.2.2), le résultat du parsing se présente en colonnes qui contiennent, pour chaque token de la phrase, l'identifiant du token, le token lui-même, l'identifiant de son gouverneur et l'étiquette de la fonction syntaxique du token (cf. figure 3.1b). Les outils qui effectuent le parsing sont des **parsers**.

Au-delà du fait de fournir des informations sur le fonctionnement syntaxique d'une langue, le parsing est également exploité dans le cadre d'autres applications, comme l'identification des rôles sémantiques, l'extraction de relations ou d'évènements, ou encore la traduction automatique.

```
[ S
  [ NP Filip ]
  [ VP studira
    [ NP lingvistiku ]
    [ PP u
      [ NP Italiji ]
    ]
  ]
]
```

(a) Parsing en constituants

ID token	Token	ID gouverneur	Relation
1	Filip	2	Sujet
2	studira	0	Racine
3	lingvistiku	2	Objet direct
4	u	2	Circonstant
5	Italiji	4	Compl. de prép.

(b) Parsing en dépendances

FIGURE 3.1 – Illustrations de la tâche de parsing sur la phrase *Filip studira lingvistiku u Italiji* 'Filip étudie la linguistique en Italie'

3.1.4 Métriques d'évaluation

Pour évaluer la qualité de l'annotation produite par les outils automatiques, différentes métriques peuvent être utilisées. Pour l'étiquetage morphosyntaxique, on utilise le plus souvent l'exactitude, la précision et le rappel (cf. à titre d'exemple Paroubek, 2007), ou encore la f-mesure (cf. Manning & Schutze, 1999). L'**exactitude** correspond au pourcentage de tokens bien annotés dans le corpus. Il s'agit d'une mesure globale calculée sur la totalité du corpus. Si l'on souhaite évaluer les performances de l'outil par rapport à une étiquette spécifique, on peut utiliser la précision et le rappel. La **précision** correspond au pourcentage de tokens classés par l'outil dans une catégorie qui appartiennent effectivement à cette catégorie. Inversement, le **rappel** est calculé comme le pourcentage de tous les tokens appartenant à une catégorie qui ont été correctement identifiés par l'outil. Il faut noter que ces deux mesures fournissent des informations complémentaires : la précision ne renseigne pas sur le nombre d'occurrences que le système n'a pas identifiées, alors que le rappel ne renseigne pas sur le nombre d'annotations incorrectes parmi les annotations produites par le système. Pour permettre une vision globale de ces caractéristiques de l'annotation, la **f-mesure** combine ces deux mesures sous la forme d'une moyenne harmonique. [3]

Pour illustrer, imaginons qu'un étiqueteur ait produit l'annotation fournie par le tableau 3.2, à une différence près : la forme *lingvistiku* a été annotée comme adjectif. Dans ce cas, l'exactitude de l'étiqueteur serait de 80 %, étant donné que 4 tokens sur 5 ont été bien annotés. Les formes étiquetées comme noms (*Filip* et *Italiji*) appartiennent effectivement à cette classe toutes les deux ; la précision de l'outil pour la classe des noms est donc de 100 %. En revanche, son rappel est plus bas : des 3 noms présents dans l'exemple (*Filip*, *lingvistiku* et *Italiji*), l'outil n'en a détecté que 2 (*Filip* et *Italiji*). Le rappel est par

3. Elle est en général calculée de la manière suivante : $F_1 = 2 \times \frac{\text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}}$.

conséquent de 66,66 %. La f-mesure correspondante est donc $F_1 = 2 \times \frac{100 \times 66,66}{100+66,66} = 79,99$.

Quant à la lemmatisation, c'est typiquement l'exactitude qui est utilisée pour l'évaluation. Dans le cadre de ces deux tâches, on accorde une attention spéciale au traitement de **mots inconnus** (ou **hors vocabulaire**). Il s'agit des mots qui ne figurent pas dans le corpus d'apprentissage ; par conséquent, le taux de réussite avec lequel un outil traite ces mots reflète sa capacité à faire des généralisations à partir du corpus d'apprentissage.

En ce qui concerne les parsers, différentes mesures sont utilisées pour l'analyse en constituants et l'analyse en dépendances. Comme cette thèse exploite le parsing en dépendances, nous nous focaliserons sur celui-ci. Dans ce cadre, les deux métriques les plus répandues sont le **LAS** (*labelled attachement score*, score de rattachement labellisé) et l'**UAS** (*unlabelled attachment score*, score de rattachement non labellisé). Le LAS exprime le pourcentage des tokens pour lesquels le parser a correctement identifié le gouverneur et la fonction syntaxique, alors que l'UAS correspond au pourcentage des tokens pour lesquels le parser a bien identifié le gouverneur sans prendre en compte la fonction attribuée. Une troisième mesure est parfois utilisée : **LA**, ou *label accuracy* (exactitude du label). Cette mesure correspond au pourcentage des tokens auxquels la bonne relation syntaxique a été attribuée, sans prendre en compte l'identification du gouverneur. Notons que les trois mesures représentent en effet des taux d'exactitude (pourcentage d'annotations correctes), calculés en considérant différents éléments de l'annotation. Pour chacune de ces trois configurations (rattachement labellisé, rattachement non labellisé, exactitude du label), il est également possible de calculer la précision, le rappel et la f-mesure au niveau de différentes étiquettes. Cependant, ces métriques sont en général réservées à des évaluations plus ciblées.

3.1.5 Chaîne de traitement typique

L'étiquetage morphosyntaxique, la lemmatisation et le parsing font souvent partie d'une chaîne de traitement complexe, dans laquelle ces tâches s'effectuent en cascade. L'ordre d'exécution des cascades est conditionné par les besoins des outils statistiques : en annotation, les étiqueteurs morphosyntaxiques n'exploitent en général que les tokens (formes fléchies), les lemmatiseurs se basent sur la forme fléchie et l'étiquette morphosyntaxique, et les parsers utilisent la forme fléchie, les informations morphosyntaxiques et le lemme. Par conséquent, dans une chaîne de traitement typique, c'est d'abord l'annotation morphosyntaxique qui est effectuée, suivie de la lemmatisation, pour finir avec l'annotation syntaxique. Le processus est schématisé dans la figure 3.2 [4].

4. La figure n'intègre pas la phase préalable de tokénisation, qui réalise la segmentation du texte courant en unités de traitement. Ces unités sont typiquement des mots graphiques, mais il peut également s'agir d'expressions polylexicales. Pour l'exemple d'un tokéniseur basé sur l'apprentissage automatique, voir (Urieli, 2013, p. 80-85).

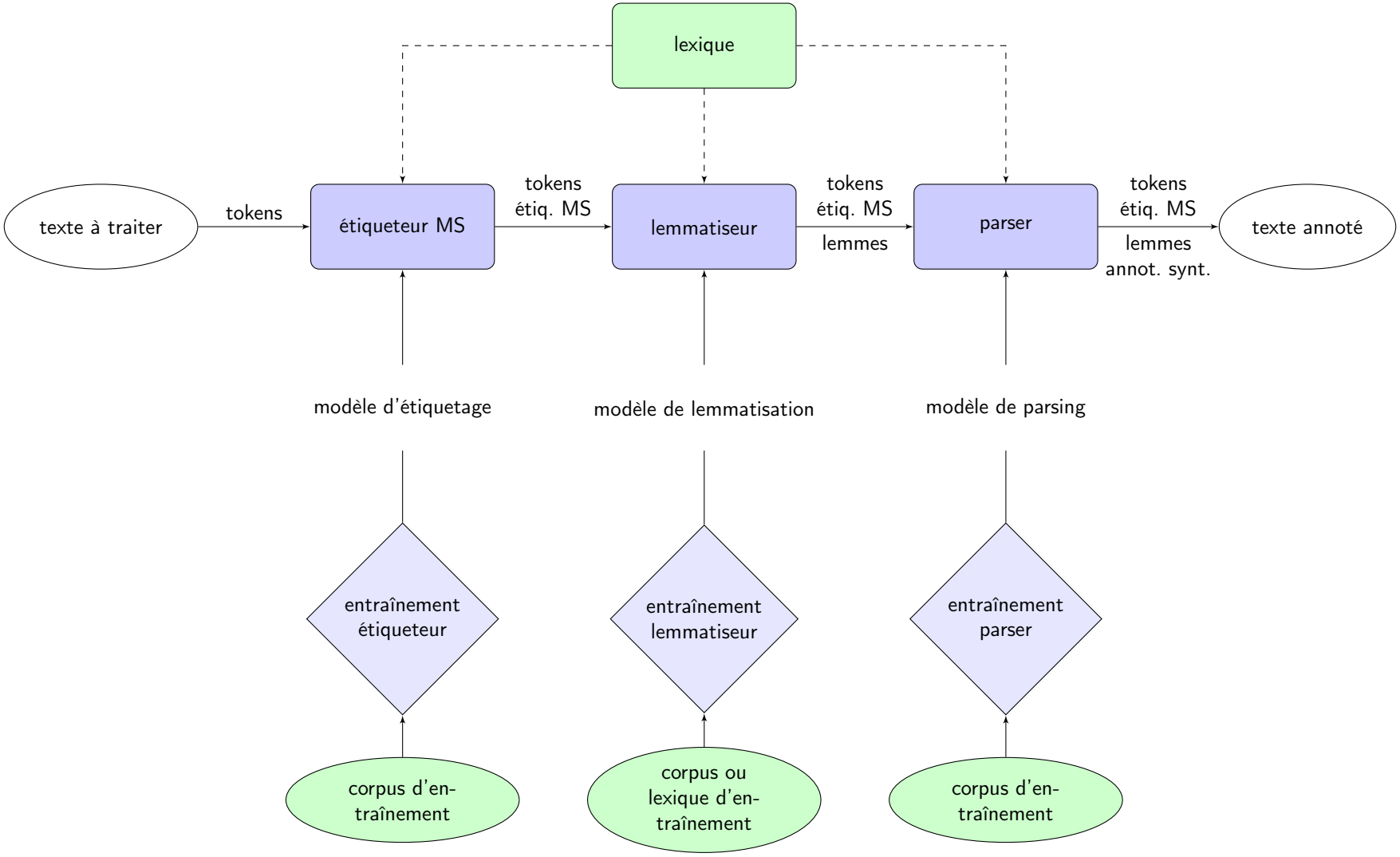


FIGURE 3.2 – Exemple d'une chaîne de traitement en TAL

Comme le montre le schéma, chacun des modules de traitement repose sur un modèle créé à partir d'une ressource d'entraînement dans le cadre d'un processus d'apprentissage. Par ailleurs, la majorité des outils, qu'il s'agisse d'un étiqueteur, d'un lemmatiseur ou d'un parser, intègrent la possibilité d'exploiter un lexique externe au moment de l'analyse. Une telle ressource doit idéalement être complémentaire par rapport au corpus d'entraînement et faciliter le traitement des formes hors vocabulaire. Tout comme un seul lexique peut être exploité par les trois outils s'il est doté de tous les types d'informations nécessaires, un même corpus peut suffire pour l'entraînement s'il dispose des trois couches d'annotation.

Notons encore que chacun des modules de traitement s'appuie fortement sur la sortie du module précédent. Par conséquence, la présence d'erreurs dans une couche de traitement peut se propager à travers toute la chaîne, d'où l'intérêt d'assurer un bon niveau de performances des outils individuels.

3.2 Étiqueteurs morphosyntaxiques

Dans cette section, nous présentons les principaux types d'algorithmes utilisés en étiquetage morphosyntaxique, ainsi que les outils qui les mettent en place. Pour chacun d'entre eux, nous donnons le taux d'exactitude rapporté dans la publication originale présentant l'outil. Nous soulignons cependant que les résultats ne sont pas toujours directement comparables, vu que les modalités d'évaluation (notamment la taille du corpus d'apprentissage et la taille du jeu d'étiquettes) diffèrent d'un travail à l'autre. Les conditions exactes de chaque évaluation sont données dans la table récapitulative 3.5.

3.2.1 Modèles de Markov cachés

Les modèles de Markov cachés (angl. *Hidden Markov Models*, dorénavant HMM) figurent parmi les premiers algorithmes appliqués à la tâche d'étiquetage morphosyntaxique. Ces algorithmes exploitent les séquences de tokens et d'étiquettes qui leur sont associées pour prédire l'étiquette du token à traiter. Le modèle à bigrammes (exploitant les séquences de deux tokens) mis en place par Merialdo (1994) a atteint une exactitude moyenne de 97 % sur l'anglais. Afin d'améliorer l'exploitation du contexte dans l'apprentissage, des versions de HMM à trigrammes ont été introduites, cf. l'étiqueteur TnT (Brants, 2000b). Au-delà des tokens et des étiquettes, cet outil exploite également le lien entre les étiquettes et les suffixes aussi bien que celui entre les étiquettes et la présence de majuscules dans le token. L'outil atteint 96,7 % d'exactitude sur le corpus allemand NEGRA, ainsi que sur le PennTreebank. Au moment de la publication de l'outil, ces performances étaient au même niveau que celles d'algorithmes plus complexes tels le modèle à maximisation d'entropie de Ratnaparkhi (1996) (cf. section 3.2.3).

Malgré la mise au point d'autres méthodes plus performantes, cet outil reste très populaire. À titre d'illustration, il a été utilisé récemment dans le travail de Maier et al. (2014) sur l'allemand et s'est montré, d'après les auteurs, le plus fiable pour l'étiquetage de cette langue dans différentes conditions d'apprentissage (*ibid*, p. 9). Outre ses performances compétitives, l'outil est également doté d'une vitesse d'apprentissage et d'exécution élevée, garantie par la relative simplicité de son algorithme.

3.2.2 Arbres de décision

Un autre type d'algorithmes initialement utilisés est celui des arbres de décision, dont le représentant le plus connu est TreeTagger de Schmid (1994). Cet algorithme envisage l'étiquetage comme un problème de prise de décision. L'ensemble des décisions possibles est représenté sous forme d'un arbre, où les nœuds représentent les questions (des tests sur les propriétés du contexte du mot à étiqueter) et les branches représentent les réponses possibles à chaque question. Chaque nœud définit une distribution de probabilité pour toutes les décisions possibles. Le parcours de l'arbre s'arrête au moment où l'algorithme atteint un nœud terminal exprimant la probabilité pour la décision donnée. TreeTagger a été évalué sur le corpus PennTreebank. La version de l'algorithme la plus performante atteint 96,36 % d'exactitude.

Tout comme TnT, TreeTagger a connu ce qui est sans doute l'une des exploitations les plus larges dans la communauté du TAL. Son utilisation est facilitée par la simplicité d'installation et d'utilisation, ainsi que par le fait qu'un grand nombre de modèles pour des langues différentes ont été entraînés et librement diffusés [5].

3.2.3 Modèles de Markov à maximisation d'entropie

Par rapport aux modèles de Markov cachés de base, les modèles de Markov à maximisation d'entropie (angl. *Maximum Entropy Markov Models*, dorénavant MEMM) élargissent le contexte exploité et facilitent l'inclusion de traits d'apprentissage plus riches. Ils permettent notamment de prendre en compte les tokens à droite du token observé et améliorent la gestion des mots inconnus à travers des traits dédiés. Une autre différence par rapport aux HMM relève de la manière dont les traits d'apprentissage sont exploités. Les HMM supposent l'indépendance des traits, ce qui peut amener à surestimer les informations fournies par des traits corrélés. En revanche, les MEMM prennent en compte la corrélation des traits et ajustent le calcul de probabilité en fonction de ce facteur. La première implémentation de cet algorithme pour l'étiquetage morphosyntaxique est MXPOST de Ratnaparkhi (1996), qui atteint 96,6 % d'exactitude sur l'anglais.

5. `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`. Dernier accès : 31 octobre 2017.

Un point faible important des MEMM est l'unidirectionnalité : le fait que le système procède de gauche à droite lors de l'étiquetage l'empêche d'exploiter les informations sur le contexte droit du token traité. Pour combler cette lacune, Toutanova et al. (2003) introduisent une extension du MEMM nommée réseau de dépendance cyclique (angl. *cyclic dependency network*). Cette augmentation permet de prendre en compte de manière explicite les contextes gauche et droit au niveau des étiquettes. L'outil optimise également l'exploitation des informations lexicales et le traitement des formes hors vocabulaire. Ces modifications mènent à une exactitude moyenne de 97,24 % sur l'anglais.

Des implémentations de MEMM ont été développées pour d'autres langues, comme Morfette (Chrupała et al., 2008) et MElt (Denis & Sagot, 2009), proposés pour le français et, de manière générale, pour les langues à morphologie flexionnelle riche. Morfette effectue la lemmatisation et l'étiquetage de manière conjointe. Le système a été évalué sur le roumain, l'espagnol et le polonais en atteignant respectivement 96,83 %, 95,4 % et 84,91 % d'exactitude. La particularité de MElt (Sagot, 2016) réside dans l'utilisation d'un lexique à large couverture en tant que source de traits d'apprentissage supplémentaires, à la différence de l'approche plus traditionnelle qui consiste à exploiter ce type de données comme contraintes au moment de l'étiquetage. Il obtient 96,14 % sur le français et, à titre d'exemple, 98,58 % sur le tchèque, 96,70 % sur le croate et 97,19 % sur l'allemand.

3.2.4 Machines à vecteurs de support

Des modèles plus puissants ont également été adaptés à la tâche d'étiquetage morphosyntaxique, tels les outils basés sur les machines à vecteurs de support (angl. *Support Vector Machines*, dorénavant SVM). Le SVM n'exploite pas les probabilités, mais crée une fonction discriminante qui effectue la classification des données dans des catégories. Les outils basés sur cet algorithme disposent d'une bonne capacité de généralisation. Grâce à la fonction de noyau (*kernel*), ils sont capables d'avoir des règles de décision non linéaires et d'exploiter des combinaisons de plusieurs traits dans l'apprentissage. Cet algorithme a été implémenté dans SVMTool de Giménez & Marquez (2004) et a obtenu 97,16 % d'exactitude sur l'anglais et 96,89 % sur l'espagnol. Le module d'étiquetage intègre la possibilité de combiner les deux directions de traitement en effectuant deux passes sur les données (de gauche à droite et de droite à gauche) et ce traitement bi-directionnel améliore les résultats de l'outil de manière significative (*ibid*, p. 45).

3.2.5 Modèles à champs aléatoires conditionnels

À la différence des modèles cités ci-dessus, les outils basés sur les champs aléatoires conditionnels (angl. *Conditional Random Fields*, dorénavant CRF) sont capables d'assurer le traitement bi-directionnel des données en une seule passe. Introduits par Lafferty et al.

Outil	Algorithme	Langue	Corpus	Taille	Jeu étiq.	Exact.
Merialdo (Merialdo, 1994)	HMM à bigrammes	anglais	IBM Associated Press	1 M	76	97 %
TnT (Brants, 2000b)	HMM à trigrammes	anglais	PennTreebak	2 M	36	96,7 %
		allemand	NEGRA	355 K	57	96,7 %
TreeTagger (Schmid, 1994)	arbres de décision	allemand	TIGER	885 K*	57*	96,36 %
				(avec lexique)	>700	91,07 %
MXPOST (Ratnaparkhi, 1996)	MEMM	anglais	Wall Street Journal	962 K	36	96,6 %
Stanford (Toutanova et al., 2003)	MEMM + cyclic dep. net.	anglais	Wall Street Journal	1 M*	36	97,24 %
Morfette (Chrupała et al., 2008)	MEMM	roumain	MultextEast	88 K	616*	96,83 %
		espagnol	CESS-ECE	168 K	280*	95,40 %
		polonais	IPI PAN	219 K	>2000*	84,91 %
MElt (Sagot, 2016)	MEMM + lexique comme traits	français	UD	366 K	18	96,14 %
		tchèque	UD (PDT)	1,1 M	18	98,58 %
		croate	UD (SETimes)	78 K	14	96,70 %
		allemand	UD (TIGER)	77 K	54*	97,19 %
SVMTool (Giménez & Marquez, 2004)	SVM	anglais	Wall Street Journal	900 K	36	97,16 %
		espagnol	LEXESP	86 K	(pas indiqué)	96,89 %
MarMoT (Müller & Schütze, 2015)	CRF	allemand	TIGER	885 K	>700	90,60%
		tchèque	CoNLL 2009	6,5 M	>1800	93,06%
Ling et al. (2015)	bi-LSTM	anglais	Wall Street Journal	1 M*	36	97,36 %
		allemand	TIGER	855 K*	>700*	98,08 %
		turque	(Atalay et al., 2003)	67 K	15	94,59 %
Plank et al. (2016)	bi-LSTM + fréquence	arabe	UD	242 K*	16*	98,91 %
		tchèque	UD	1,5 M*	17*	98,24 %
		français	UD	391 K*	17*	96,11 %

TABLE 3.5 – Évaluations des étiqueteurs. Les valeurs marquées par l'astérisque ne sont pas indiquées dans la publication originale, mais ont été récupérées dans la documentation des corpus.

(2001), les CRF se sont montrés très performants sur de nombreuses tâches du TAL, mais ils ont un point faible non négligeable : leur temps d'apprentissage (et d'exécution) est une fonction polynomiale de la taille du jeu d'étiquettes. Par conséquent, l'utilisation des outils à base de CRF avec des jeux d'étiquettes étendus est très coûteuse en termes de temps, et ces systèmes sont plus souvent utilisés pour la reconnaissance des entités nommées et le chunking. Cependant, l'étiqueteur MarMoT (Müller et al., 2013) met en place une approximation qui permet d'appliquer cet algorithme à des jeux d'étiquettes étendus tout en maintenant le temps d'apprentissage et d'exécution dans des limites raisonnables. L'outil a été évalué sur l'arabe, le tchèque, l'espagnol, l'allemand et le hongrois en atteignant de manière systématique des résultats élevés, aussi bien au niveau de l'étiquetage en parties du discours que de l'annotation morphosyntaxique détaillée (à titre d'exemple, 93,06 % sur le tchèque et 90,60 % sur l'allemand avec les jeux d'étiquettes détaillés).

D'autres implémentations des CRF comprennent, par exemple, Wapiti (Lavergne et al., 2010) et l'étiqueteur de Constant & Sigogne (2011). Les CRF ont également été utilisés par Ljubešić et al. (2016) pour l'étiquetage du croate et du serbe, avec une exactitude au-delà de 97 % en étiquetage en parties du discours, et au-delà de 92 % en étiquetage détaillé (cf. section 3.2.7).

3.2.6 Méthodes à base de réseaux de neurones

Les innovations les plus récentes dans l'étiquetage morphosyntaxique mettent en place des méthodes par réseaux de neurones, et notamment par réseaux de neurones récurrents (angl. *Recurrent Neural Networks*, dorénavant RNN). Ling et al. (2015) utilisent un modèle bi-LSTM (*bidirectional long short-term memories*) pour faire de la modélisation au niveau des caractères. Ceci leur permet de capter des similarités fonctionnelles entre les tokens partageant une similarité orthographique, apport particulièrement utile pour les langues à morphologie flexionnelle riche. Ce modèle atteint une exactitude de 97,36 % sur l'anglais, 98,08 % sur l'allemand et 94,59 % sur le turc.

Plank et al. (2016) exploitent également un modèle de bi-LSTM, mais ajoutent la possibilité de prédire la fréquence des tokens dans le but d'améliorer le traitement des mots rares et inconnus. L'outil a été évalué sur 22 corpus du projet Universal Dependencies couvrant des langues slaves, romanes, germaniques et sémitiques. Parmi les résultats les plus parlants, on peut citer une exactitude de 98,91 % obtenue sur l'arabe, 98,24 % sur le tchèque, 96,11 % sur le français. L'exactitude moyenne sur les langues slaves était de 97,50 %, et le résultat le moins élevé 93,30 % sur le néerlandais.

3.2.7 Étiquetage morphosyntaxique du serbe

Malgré cette diversité d'approches et outils, le serbe reste une langue relativement peu explorée en ce qui concerne l'étiquetage en parties du discours et l'analyse morphosyntaxique fine. Le tableau 3.7 présente les conditions d'évaluation détaillées pour chacun des travaux mentionnés ci-dessous.

Les meilleurs résultats rapportés jusqu'à maintenant en annotation morphosyntaxique détaillée du serbe sont ceux cités par Jakovljević et al. (2014). Ces auteurs indiquent que AlfaNum POS tagger (Sečujski, 2009), un étiqueteur basé sur des règles, atteint une exactitude de 93,2 % sur un jeu de plus de 700 étiquettes (Jakovljević et al., 2014, p. 43-44). Malheureusement, cet outil n'est pas librement disponible.

Quant aux outils statistiques, plusieurs algorithmes ont été testés sur le serbe selon des modalités différentes. BTagger (Gesmundo & Samardžić, 2012), le seul étiqueteur par apprentissage automatique développé pour le serbe, met en place un système de classification bi-directionnelle de séquences basé sur l'algorithme de perceptron. Il atteint une exactitude de 86 % en analyse morphosyntaxique fine avec un jeu d'étiquettes détaillé de plus de 900 tags.

TreeTagger (Schmid, 1994) a été évalué par Utvić (2011) et a obtenu le score de 96,5 % sur un jeu d'étiquettes minimaliste de 16 tags. Cet entraînement a été fait sur un corpus de 1 million de tokens, qui n'est malheureusement pas en diffusion libre.

HunPos (Halácsy et al., 2007) a été testé sur le croate et le serbe par Agić et al. (2013a). Les auteurs ont signalé une exactitude de 87 % pour le croate et de 85 % pour le serbe avec un jeu d'étiquettes détaillé de plus de 600 tags, et des scores de 97 % et 96 % respectivement pour l'étiquetage en parties du discours.

Dans une expérience antérieure à la thèse (Miletic, 2013), nous avons testé trois outils : BTagger, TreeTagger et TnT (Brants, 2000b). Le jeu d'étiquettes utilisé encode les parties du discours et les sous-catégories grammaticales (47 étiquettes). Les étiqueteurs ont atteint une exactitude moyenne de 94 % (BTagger), 93 % (TnT) et 92 % (TreeTagger).

Plus récemment, Ljubešić et al. (2016) ont utilisé un étiqueteur basé sur les CRF sur le croate et le serbe. L'outil a atteint respectivement 98,11 % et 97,86 % d'exactitude sur les étiquettes des parties du discours (12 tags), alors qu' en étiquetage morphosyntaxique détaillé avec plus de 1200 étiquettes il a obtenu respectivement 92,53 % et 92,33 %.

Si l'on observe les travaux effectués sur des corpus de tailles relativement comparables (80-110 K tokens) (cf. tableau 3.7) avec des outils librement disponibles, une tendance claire se dégage : en accord avec les observations globales concernant la dispersion des données au niveau morphosyntaxique (cf. section 2.3.2), les résultats de l'étiquetage du serbe semblent impactés auss bien par la taille du jeu d'étiquettes que par le choix de l'étiqueteur. En effet, comme le montre le travail d'Agić et al. (2013a), l'exploitation d'un

Outil	Algorithme	Travail d'éval.	Corpus	Taille	Jeu étiq.	Exact. (%)
AlfaNum (Sečujski, 2009)	symbolique	(Jakovljević et al., 2014)	-	-	>700	93,2
BTagger (Gesmundo & Samardžić, 2012)	perceptron	(Gesmundo & Samardžić, 2012)	MultextEast	108 K	>900	86,0
		(Miletic, 2013)	ParCoTrain	100 K	47	94,0
TnT (Brants, 2000b)	HMM à trigrammes	(Miletic, 2013)	ParCoTrain	100 K	47	93,0
TreeTagger (Schmid, 1994)	arbres de décision	(Miletic, 2013)	ParCoTrain	100 K	47	92,0
		(Utvić, 2011)	INTERA	1 M	16	96,5
HunPos (Halácsy et al., 2007)	HMM à trigrammes	(Agić et al., 2013a)	SETimes.hr	89 K	>600	85,0
					12	96,0
Ljubešić et al. (2016)	CRF	(Ljubešić et al., 2016)	hr500K	500 K	>1200	92,3
					12	97,9

TABLE 3.7 – Étiquetage morphosyntaxique du serbe

jeu minimaliste de 12 étiquettes permet de dépasser le seuil d'exactitude de 96 %. Avec un jeu de taille moyenne (47 étiquettes), les performances de différents outils sont autour de 92-94 % (cf. Miletic, 2013). Enfin, avec un jeu à plusieurs centaines d'étiquettes, les résultats chutent à 85-86 % (cf. Gesmundo & Samardžić, 2012 ; Agić et al., 2013a). À notre connaissance, le seul étiqueteur statistique qui dépasse ces valeurs avec un jeu d'étiquettes étendu est celui de Ljubešić et al. (2016). Ces résultats indiquent qu'un l'algorithme puissant (CRF) facilite l'étiquetage du serbe d'une manière importante comparé à d'autres algorithmes plus simples, comme l'arbre de décisions de TreeTagger, le HMM à bigrammes de TnT, celui à trigrammes de HunPos, voire le perceptron de BTagger. Il ne faut cependant pas négliger le fait que cet outil a été entraîné sur un corpus bien plus large que les autres outils (500 000 tokens). Une partie de l'amélioration est certainement due à ce fait.

Malheureusement, ce travail est ultérieur au moment où nous devions faire le choix de l'étiqueteur (automne 2015) ; nous n'avons donc pas pu l'exploiter. Précisons que l'outil sélectionné doit être capable de travailler avec un jeu d'étiquettes étendu. En effet, bien que nous envisagions la séparation de l'annotation morphosyntaxique en plusieurs couches dans le corpus final, le processus de création du corpus pose des exigences différentes. Dans cette étape, l'objectif est de fournir une annotation de base qui sera corrigée par les annotateurs humains. Cela signifie qu'elle doit contenir toutes les informations visées : les étiquettes des parties du discours, mais aussi les traits morphosyntaxiques fins. La manière

la plus simple de le faire est de baser la préannotation sur un jeu d'étiquettes étendu, avec des étiquettes détaillées englobant tous ces éléments. Par conséquent, l'étiqueteur choisi doit être capable de maîtriser un tel jeu.

Parmi les étiqueteurs présentés ci-dessus, deux ont été évalués selon cette modalité : BTagger et HunPos. Ils ont été entraînés sur des corpus de tailles relativement proches, et ils obtiennent des résultats quantitatifs comparables : 86 % d'exactitude pour BTagger, et 85 % pour HunPos. Nous avons cependant éliminé BTagger en raison d'un désavantage pratique important : sa vitesse d'exécution. Lors de nos propres expériences décrites dans (Miletic, 2013), BTagger s'est montré l'outil le plus lent parmi ceux testés : un cycle d'apprentissage sur un corpus d'entraînement de 60 000 tokens prenait 1 h 20 min, et un cycle d'évaluation sur 20 000 tokens durait 40 minutes. À titre de comparaison, l'exécution de TnT et de TreeTagger dans les mêmes conditions ne prenait que quelques secondes. Des observations comparables ont été faites par Agić et al. (2013a) : dans le cadre d'une évaluation préliminaire, BTagger avait mis plus de 6 h pour l'entraînement et 87 sec pour l'évaluation, alors que HunPos avait mis respectivement 1,1 sec et 0,11 sec. Nous avons donc retenu ce dernier pour la suite de notre travail.

3.3 Lemmatiseurs

La complexité de la lemmatisation dépend fortement de la langue traitée. Pour l'anglais, des outils performants ont été établis depuis les années 1980, comme le *stemmer* de Porter (1980). Cet outil n'effectue pas une lemmatisation proprement dite, mais une désuffixation, qui transforme les formes fléchies en leur radical plutôt qu'en la forme canonique pleine. Cet algorithme relativement simple basé sur une cinquantaine de règles symboliques est encore utilisé. En revanche, la lemmatisation des langues à morphologie flexionnelle riche s'est avérée plus difficile. La suite de cette section propose une description des principaux types d'approches et de leurs performances. Les conditions détaillées des différentes évaluations citées sont disponibles dans le tableau 3.8.

3.3.1 Méthodes par recherche en dictionnaire

Les premières méthodes de lemmatisation s'appuyaient sur une démarche de recherche des formes canoniques dans un dictionnaire externe. Ce procédé était éventuellement couplé à un mécanisme de désambiguïsation basé sur la catégorie du mot. Un exemple de ce type d'outils est Freeling de Carreras et al. (2004). Il a été évalué sur l'espagnol et le catalan dans (Chrupała, 2006) en atteignant respectivement 95,05 % et 98,32 % d'exactitude sur les mots connus. Cette approche a cependant un point faible très important : le traitement des formes hors vocabulaire. Si une forme est absente du lexique intégré à

l'outil, le système n'est pas capable d'effectuer la lemmatisation et son seul recours est de reprendre la forme fléchie au lieu de fournir le lemme.

Les méthodes de lemmatisation par apprentissage automatique cherchent à pallier ce problème. La majorité des travaux disponibles peuvent être regroupés en deux ensembles : ceux qui mettent en place des algorithmes pour l'apprentissage de règles de transformation basé sur des informations hors contexte (autrement dit, apprises sur des lexiques) et ceux qui redéfinissent le problème de la lemmatisation comme un problème de classification et s'appuient sur un apprentissage en contexte (sur un corpus annoté). Les deux types de méthodes sont décrits dans la suite.

3.3.2 Méthodes par apprentissage hors contexte

Les lemmatiseurs RDR (Plisson et al., 2008), CST (Jongejan & Dalianis, 2009) et LemmaGen (Juršic et al., 2010) mettent en œuvre le même principe de base : ils dérivent des règles de transformation d'une forme fléchie en lemme à partir de données présentées sous forme de paires *forme fléchie – lemme*. RDR construit progressivement son modèle de traitement : chaque transformation rencontrée dans les données d'apprentissage est confrontée au modèle existant et, si nécessaire, une nouvelle règle ou une exception à une règle existante est ajoutée. Lors du traitement, les transformations sont activées de la plus générale vers les plus spécifiques jusqu'à ce qu'on trouve la règle la plus spécifique qui s'applique au mot en question. D'après l'évaluation décrite dans (Juršic et al., 2010), cet outil atteint 97,8 % d'exactitude sur l'anglais et 96,8 % sur le français, mais 88,3 % sur le roumain et 85,2 % sur le serbe.

LemmaGen est basé sur l'algorithme de RDR, mais il propose une modification pour améliorer le traitement des formes inconnues : pour chaque mot hors vocabulaire rencontré, l'outil se sert d'une mesure de similarité pour identifier la classe de formes la plus proche. Il applique ensuite à la forme inconnue la transformation associée à cette classe. Cette extension mène à une amélioration légère, mais systématique sur l'ensemble des langues évaluées dans (Juršic et al., 2010) (généralement $<$0,5 %).

CST met en place une approche différente. Chaque règle de lemmatisation est exprimée sous la forme d'un patron de recherche et d'une chaîne de caractères de substitution. Le patron de recherche est appliqué à la fin des mots à traiter et le patron le plus long est utilisé pour effectuer la lemmatisation. Le système ne dispose pas de mécanisme de désambiguïsation : si un patron de recherche correspond à plusieurs substitutions, tous les lemmes possibles sont produits. Dans l'évaluation multilingue de Juršic et al. (2010), CST atteint 96,0 % sur l'anglais, 94,8 % sur le français, et 83,0 % sur le roumain et sur le serbe. L'outil a également été utilisé par Agić et al. (2013a) sur le croate et le serbe et a obtenu respectivement 97,78 % et 96,61 % d'exactitude (pour plus de détails, voir

la section 3.3.4). En 2009, l'outil a été modifié de sorte à prendre en compte tout type d'affixes, et non pas seulement les suffixes (Jongejan & Dalianis, 2009).

Ces approches partagent un point faible : de par leur système d'apprentissage, effectué sur un lexique hors contexte, leur capacité de généralisation à des formes inconnues est souvent insatisfaisante. Pour obtenir des résultats élevés, ces outils doivent être entraînés sur des ressources lexicales importantes. Ils ne sont pas équipés non plus pour faire de la désambiguïsation basée sur le contexte. Pour dépasser ces limites, des approches par apprentissage supervisé ont été développées.

3.3.3 Méthodes par apprentissage en contexte

Morfette (Chrupała et al., 2008), BTagger (Gesmundo & Samardžić, 2012) et Lemming (Müller et al., 2015) figurent parmi les méthodes de lemmatisation basées sur un apprentissage supervisé en contexte. Ces outils envisagent la lemmatisation comme une tâche de classification, comparable à celle de l'étiquetage morphosyntaxique. À la différence de ce dernier, où les classes attribuées aux données sont prédéterminées sous la forme d'un jeu d'étiquettes et typiquement appliquées aux données d'apprentissage par des humains, dans le cas de la lemmatisation les classes sont également inférées de manière automatique à partir des données lemmatisées. Les labels des classes correspondent à l'expression de la transformation nécessaire pour effectuer la lemmatisation d'une forme fléchie. Ces outils sont typiquement capables d'effectuer l'étiquetage morphosyntaxique et la lemmatisation de manière conjointe.

Morfette met en place un module dédié à l'étiquetage et un deuxième pour la lemmatisation, les deux basés sur le MEMM (cf. section 3.2.3) [6]. Les classes pour la lemmatisation sont identifiées en utilisant le *shortest edit script*. Il s'agit de l'ensemble minimal d'instructions d'insertion et de suppression de caractères qui permet de transformer une chaîne de caractères en une autre. L'apprentissage prend également en compte les suffixes et préfixes de la forme à lemmatiser, l'étiquette morphosyntaxique de la forme à traiter proposée par le système lui-même, et le patron d'écriture du mot (qui prend en compte la présence des majuscules, des chiffres, des signes de ponctuation). Morfette a été évaluée sur le roumain, l'espagnol et le polonais : sur les mots connus, l'outil atteint respectivement 97,78 %, 98,52 % et 95,55 % d'exactitude globale, alors que sur les formes inconnues les scores respectifs sont à 82,88 %, 91,22 % et 81,11 %.

Lemming est basé sur les CRF (cf. section 3.2.5). L'identification des classes est proche de celle de Morfette : les transformations nécessaires pour passer de la forme fléchie au lemme sont représentées sous la forme d'arbres d'édition, qui expriment les manipulations à effectuer sur les suffixes et les préfixes de la forme fléchie. Le processus d'apprentissage

6. Des versions ultérieures mettent en place le modèle de perceptron (Collins, 2002), cf. le fichier *readme* dans le dépôt GitHub : `https://github.com/gchrupala/morfette`

Outil	Type	Travail d'éval.	Langue	Ressource d'entraîn.	Taille	Exactitude	
						connus	inconnus
Freeling	par dictionnaire	(Chrupała, 2006)	espagnol	-	-	95,05	82,05 [7]
(Carreras et al., 2004)			catalan	-	-	98,32	77,16
RDR	hors contexte	(Juršic et al., 2010)	anglais	lexique Multext	66 K entrées	97,8	-
(Plisson et al., 2008)			français	lexique Multext	306 K entrées	96,8	-
			roumain	lexique MultextEast	55 K entrées	88,3	-
			serbe	lexique MultextEast	20 K entrées	85,2	-
LemmaGen	hors contexte	(Juršic et al., 2010)	anglais	lexique Multext	66 K entrées	98,0	-
(Juršic et al., 2010)			français	lexique Multext	306 K entrées	97,1	-
			roumain	lexique MultextEast	55 K entrées	88,6	-
			serbe	lexique MultextEast	20 K entrées	86,1	-
CST	hors contexte	(Juršic et al., 2010)	anglais	lexique Multext	66 K entrées	96,0	-
(Jongejan & Dalianis, 2009)			français	lexique Multext	306 K entrées	94,8	-
			roumain	lexique MultextEast	55 K entrées	83,0	-
			serbe	lexique MultextEast	20 K entrées	83,0	-
		(Agić et al., 2013a)	croate	SETimes.hr [8]		97,78	-
			serbe			95,95	-
Morfette	MEMM	(Chrupała et al., 2008)	roumain	corpus MultextEast	88 K tokens	97,78	82,88
(Chrupała et al., 2008)			espagnol	corpus CESS-ECE	168 K tokens	98,52	91,22
			polonais	corpus IPI PAN	219 K tokens	95,55	81,11
BTagger	perceptron	(Gesmundo & Samardžić, 2012)	serbe	corpus MultextEast	108 K tokens		
(Gesmundo & Samardžić, 2012)							
Lemming	CRF	(Müller et al., 2015)	allemand	corpus SPMRL 2013	100 K tokens	98,10	93,02
(Müller et al., 2015)			tchèque	corpus CoNLL 2009	*idem*	98,42	93,46
			hongrois	corpus SPMRL 2013	*idem*	98,08	94,26
			latin	corpus PROIEL	*idem*	95,36	80,94
			espagnol	corpus CoNLL 2009	*idem*	98,78	94,86

TABLE 3.8 – Évaluations des lemmatiseurs

exploite également le lemme, ses préfixes et suffixes, ainsi que les informations morphosyntaxiques disponibles. Lemming a été testé sur l'anglais, l'allemand, le tchèque, le hongrois, le latin et l'espagnol. Sur les formes connues, il dépasse 98 % d'exactitude sur toutes les langues sauf le latin, pour lequel il atteint 95,58 %. Sur les formes inconnues, ses résultats varient entre 93 % et 94 % pour toutes les langues sauf pour le latin, pour lequel ils sont de 81,47 %.

BTagger exploite l'algorithme de perceptron (Collins, 2002) dans le cadre de l'apprentissage guidé basé sur le classifieur bi-directionnel (Shen et al., 2007). L'inférence des classes pour la lemmatisation est plus simple que celle de Morfette : pour chaque paire *forme fléchie - lemme* dans le corpus d'entraînement, l'outil génère une étiquette contenant la longueur de la terminaison à supprimer et le suffixe à rajouter pour obtenir le lemme. Néanmoins, l'outil s'est montré compétitif : évalué sur le serbe (cf. Gesmundo & Samardžić, 2012), il a atteint 97,72 % d'exactitude moyenne, et 84,98 % sur les mots inconnus (cf. section 3.3.4.

3.3.4 Lemmatisation du serbe

Tout comme dans le cas de l'annotation morphosyntaxique, il existe relativement peu de travaux consacrés à la lemmatisation du serbe. L'un d'entre eux est l'évaluation de LemmaGen réalisée par Juršic et al. (2010) sur 12 langues des projets Multext et MultextEast. Les auteurs évaluent leur outil contre RDR de Plisson et al. (2008) et CST de Jongejan & Dalianis (2009). Leurs résultats globaux ayant été présentés dans la section 3.3.2, nous nous concentrerons ici sur leurs performances sur le serbe. Les trois outils ont été entraînés sur le lexique serbe du projet MultextEast, contenant environ 20 000 entrées. L'évaluation a été faite selon deux scénarios : avec et sans utilisation du lexique. Les résultats rapportés sont repris dans le tableau 3.10.

On remarque que LemmaGen obtient les meilleurs résultats dans les deux scénarios d'évaluation, mais les scores restent relativement bas. C'est notamment le cas quand on observe ses résultats sur d'autres langues typologiquement proches, comme le tchèque et le slovène : sans lexique, LemmaGen obtient respectivement 78,3 % et 79,8 % d'exactitude pour ces deux langues, alors qu'avec lexique il arrive à 90,6 % et 93,4 %. Cet écart en performances est imputé par les auteurs à la taille restreinte de la ressource d'apprentissage pour le serbe.

BTagger de Gesmundo & Samardžić (2012) obtient des scores beaucoup plus élevés dans le cadre de son évaluation initiale : 97,7 % d'exactitude moyenne, et 85,0 % sur les mots inconnus. L'outil avait été entraîné sur le corpus serbe MultextEast, qui contient environ 108 000 tokens provenant de 8 000 lemmes différents.

Grâce à ses résultats dans cette première évaluation, Agić et al. (2013a) retiennent

Outil	Type	Travail d'éval.	Ressource d'entraîn.	Taille	Mode d'éval.	Exact. (%)
LemmaGen (Juršic et al., 2010)	hors contexte	(Juršic et al., 2010)	lexique MultextEast	22 K entrées	*avec lexique*	86,1
					sans lexique	65,3
RDR (Plisson et al., 2008)	hors contexte	*idem*	*idem*	*idem*	*avec lexique*	85,2
					sans lexique	63,8
CST (Jongejan & Dalianis, 2009)	hors contexte	*idem*	*idem*	*idem*	*avec lexique*	83,0
					sans lexique	64,0
		(Agić et al., 2013a)	corpus SETimes.hr	84 K tokens	*en domaine*	95,9
				(8,9 K lemmes)	*hors domaine*	96,3
BTagger (Gesmundo & Samardžić, 2012)	perceptron	(Gesmundo & Samardžić, 2012)	corpus MultextEast	108 K tokens	*en moyenne*	97,7
				(8 K lemmes)	*mots inconnus*	85,0
		(Agić et al., 2013a)	corpus SETimes.hr	84 K tokens	*sur le croate*	96,2
				(8,9 K lemmes)		

TABLE 3.10 – Évaluations de lemmatiseurs sur le serbe

BTagger parmi plusieurs autres lemmatiseurs et étiqueteurs capables de lemmatiser (CST, PurePos (Orosz & Novák, 2012) et TreeTagger (Schmid, 1994)) pour des tests sur le croate et le serbe. Les outils ont été entraînés sur le corpus journalistique croate SETimes.hr (Agić & Ljubešić, 2014). Une évaluation initiale a été effectuée sur le croate dans laquelle BTagger et CST ont obtenu les résultats les plus élevés (respectivement 96,2 % et 97,8 %). Il est surprenant de constater que BTagger, qui est basé sur un mécanisme d'apprentissage plus puissant, ne dépasse pas CST sur cette tâche. Par ailleurs, il a fait preuve d'un temps d'exécution largement supérieur à celui de CST (6 h *vs* 1,8 secondes). Pour ces deux raisons, les auteurs ont retenu CST pour une évaluation plus détaillée sur le croate et le serbe. Dans cette deuxième étape, le modèle généré à partir des données en croate a été testé

sur les deux langues, aussi bien sur des textes journalistiques (du même domaine que le corpus d'apprentissage) que sur des échantillons de Wikipédia (domaine encyclopédique). Sur le serbe, l'outil atteint 95,9 % d'exactitude en domaine, et 96,3 % hors domaine. Nous décidons donc de retenir CST pour la suite de ce travail.

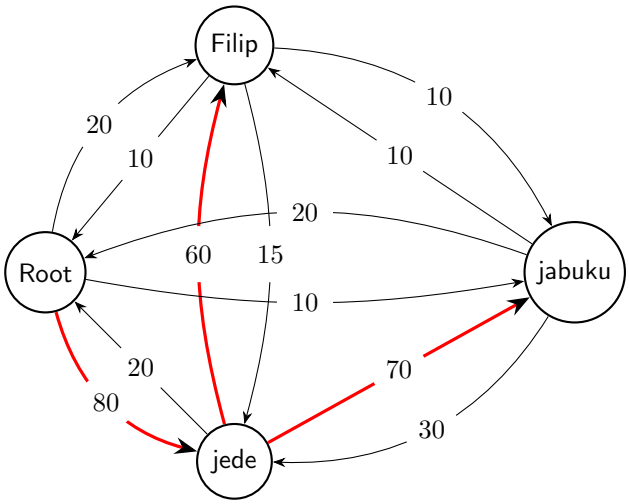
3.4 Parsers

Comme cette thèse adopte le cadre de la syntaxe en dépendances, cette section sera consacré à une présentation des parsers développés pour ce type d'analyse syntaxique. Quant aux parsers dédiés à l'analyse en constituants, nous nous contenterons d'en mentionner quelques-uns : à titre d'exemple, les travaux de Collins (2003), Charniak (2000) et Petrov et al. (2006) implémentent la PCFG (*Probabilistic Context-Free Grammar*, grammaire probabiliste hors contexte), alors que De La Clergerie et al. (2009) s'appuient sur le formalisme TAG (*tree-adjoining grammar*, grammaire d'arbres adjoints). Le lecteur est prié de se référer à ces travaux pour plus de détails.

Parmi les parsers en dépendances basés sur l'apprentissage automatique, on distingue deux approches principales : celle basée sur les graphes et celle basée sur les transitions. Chacune d'entre elles a connu des implémentations réussies, dont deux se sont établies en tant que références principales en parsing durant la dernière décennie : MST de McDonald et al. (2006) (à base de graphes) et Malt de Nivre et al. (2007b) (à base de transitions). Lors de la première campagne d'évaluation CoNLL dédiée au parsing en dépendances (Buchholz & Marsi, 2006), ces deux systèmes ont pris les deux premières places parmi 19 participants. Dans cette campagne, les outils ont été évalués sur 12 langues différentes (arabe, chinois, tchèque, danois, néerlandais, allemand, japonais, polonais, slovène, espagnol, suédois et turc). MST et Malt ont atteint respectivement 80,3 et 80,2 en score LAS, et 86,6 et 85,5 en score UAS en moyenne sur toutes les langues. Pour 11 langues sur 12, le meilleur score LAS a été obtenu par l'un de ces deux outils.

Malgré leurs performances globales très proches, MST parser et Malt parser sont plutôt complémentaires qu'équivalents. Comme le montrent McDonald & Nivre (2011), ces deux outils ont des distributions d'erreurs distinctes, qui s'expliquent par les différences théoriques des deux modèles. Ceci a été confirmé par le fait que l'intégration des deux modèles a mené à des améliorations systématiques des scores.

Le deux types de modèles et leurs implémentations les plus répandues seront présentés dans la suite de cette section. Pour tous les résultats indiqués dans le texte, les conditions exactes d'évaluation peuvent être trouvées dans la table récapitulative 3.14.



Chaque mot de la phrase est relié en graphe. L'arête arrivante avec le poids le plus élevé est sélectionnée comme gouverneur de chaque token.

FIGURE 3.3 – Représentation schématique du traitement de la phrase *Filip jede jabuku* 'Filip mange une pomme' par un algorithme basé sur les graphes.

3.4.1 Parsers à base de graphes

Les parsers basés sur les graphes considèrent l'arbre syntaxique comme un graphe et exploitent les méthodes et algorithmes de la théorie des graphes pour l'apprentissage et l'analyse. Cette approche a été introduite par Eisner (1996) et étendue ensuite par McDonald et al. (2005b), qui mettent en place un parser basé sur un arbre couvrant de poids maximal (angl. *maximum spanning tree*) basé sur l'algorithme Chu-Liu-Edmonds. L'objectif global de cet algorithme est de trouver, pour un graphe complet dont les arêtes sont pondérées, un arbre couvrant (un arbre qui connecte tous les sommets du graphe) avec une somme maximale des poids des arêtes. Plus concrètement, la version de base de l'algorithme procède de la manière suivante : la phrase à traiter est transformée en un graphe orienté complet. Autrement dit, l'outil pose des dépendances entre tous les tokens dans les deux sens. Chacune de ces dépendances se voit assigner une probabilité à partir des données observées en apprentissage, typiquement fondées sur les traits du gouverneur et du dépendant. Pour chaque token, le gouverneur le plus probable est retenu et les autres dépendances arrivantes sont éliminées, transformant ainsi le graphe en un arbre de dépendances. Un exemple avec des probabilités postiche est donné dans la figure 3.3.

Le point fort de cette approche réside dans le fait que les liens entre toutes les formes dans la phrase sont pris en considération. Autrement dit, il n'y a pas de biais lié à la longueur des dépendances, et qui plus est, rien n'empêche la création de dépendances non projectives. Pour cette raison, on considère souvent que ce type de parsers est mieux adapté au traitement des langues à morphologie flexionnelle riche, susceptibles d'avoir un

nombre important de dépendances de ce type.

En revanche, cette approche mène à une complexité temporelle élevée : le temps nécessaire pour traiter une phrase est une fonction quadratique du nombre de mots dans la phrase. Afin de maintenir un temps d'exécution raisonnable, les parsers basés sur les graphes sont souvent limités aux modèles dits « du premier ordre » : ces modèles exploitent seulement les traits liés aux deux tokens entre lesquels la dépendance doit être posée, sans s'intéresser à ceux des tokens dans le contexte linéaire (à droite ou à gauche dans la phrase) ou syntaxique (nœuds mères, filles ou sœurs des tokens traités).

Des efforts ont été faits pour dépasser ces limites et des solutions algorithmiques plus efficaces pour le parsing à base de graphes ont été proposées. Martins et al. (2009) proposent un parser basé sur la programmation linéaire en nombres entiers (angl. *integer linear programming*). Leur système a la capacité d'exploiter des traits non locaux et des contraintes globales, notamment les traits liés aux arcs voisins (ceux reliant les nœuds soeurs et mères par rapport aux nœuds analysés). Le parser a été évalué sur le danois, le néerlandais, le portugais, le slovène, le suédois, le turc et l'anglais et comparé à trois autres systèmes (McDonald & Pereira, 2006 ; Nivre & McDonald, 2008 ; Martins et al., 2008). Le parser obtient des résultats UAS entre 85,5 et 91,4 sur toutes les langues sauf le turc, pour lequel le résultat est de 76,3. Pour le néerlandais, le slovène, le suédois et l'anglais, il s'agit du meilleur score parmi les systèmes testés.

Le système de Martins et al. (2013) étend davantage la possibilité d'exploitation des traits d'apprentissage, en permettant l'inclusion des traits du troisième ordre (tokens sœurs partageant le même grand-parent, triplets des tokens sœurs) tout en gardant le parsing non projectif. L'ajout de ces traits a abouti à une amélioration du score UAS quasi-systématique à travers les 15 langues testées, et le système a atteint les meilleurs résultats rapportés sur le tchèque, l'anglais, l'allemand et le néerlandais au moment de la publication du travail.

Le travail de Bohnet (2010) est également focalisé sur l'accélération du processus de parsing. Le système implémente l'algorithme de Carreras (2007) et le combine avec l'algorithme de perceptron passif-agressif (Crammer et al., 2004 ; McDonald et al., 2005a). L'utilisation du *hash kernel* aboutit à un temps de parsing 3,5 fois plus court en moyenne que dans la version de base sur les quatre langues évaluées (l'anglais, le chinois, l'espagnol et l'allemand). Quant à la qualité de l'annotation, ses performances sont comparables à l'état de l'art sur 7 langues, avec des scores LAS allant de 76,99 pour le chinois jusqu'à 90,33 pour l'anglais.

Des systèmes basés sur ce parser ont obtenu les meilleurs résultats pour plusieurs langues dans différents scénarios dans la campagne d'évaluation SPMRL 2013 (cf. Seddah et al., 2013). L'outil a également été utilisé par Agić & Ljubešić (2015) en atteignant les scores de 86,9 LAS et 81,5 UAS sur le croate, et 86,0 LAS et 81,5 UAS sur le serbe. Les

détails de ce travail seront rediscutés plus loin (voir la section 3.4.3).

3.4.2 Parsers à base de transitions

Les parsers basés sur les transitions posent le problème du parsing comme une séquence d'actions (ou de transitions) sur les tokens de la phrase permettant de construire l'arbre syntaxique correspondant. La version de base de l'algorithme opère en utilisant les éléments suivants :

- la séquence de tokens en attente de traitement (*buffer*) ;
- la séquence de tokens qui ont été lus, mais dont le gouverneur n'a pas encore été identifié (*stack*, la pile) ;
- une séquence de transitions (d'opérations) qui permettent d'atteindre l'état actuel en partant d'un état initial.

Le token qui a été ajouté le dernier à la pile est appelé le *haut de la pile*, et le token dont c'est le tour d'être traité est appelé la *tête du buffer*. Dans la version de base de l'algorithme, les dépendances peuvent être créées seulement entre les tokens dans ces deux positions.

Au début du traitement d'une phrase, tous les tokens se trouvent dans le buffer, et la pile contient seulement le nœud racine artificiel. Les tokens sont traités un par un. Chaque fois, le parser examine si une relation de dépendance existe entre le token traité (la tête du *buffer*) et le token qui est en haut de la pile. Une fois que son gouverneur a été identifié, le token est éliminé de la structure de traitement.

Voici les décisions possibles que le parser peut prendre pour chaque paire de tokens examinée :

- transition *left-arc* : créer une dépendance allant de la tête du *buffer* vers le haut de la pile et éliminer ce dernier de la pile ;
- transition *right-arc* : créer une dépendance allant du haut de la pile vers la tête du *buffer*, éliminer ce dernier et remettre le token qui est en haut de la pile dans la position en tête du *buffer* pour qu'il soit traité de nouveau ;
- transition *shift* : mettre la tête du *buffer* en haut de la pile.

Si l'on reprend l'exemple de la figure 3.1b, nous pouvons essayer de l'analyser en simulant un parser par transitions. Le tableau 3.12 montre une séquence de transitions qui mène à la création de l'arbre syntaxique correct. Chaque ligne montre la transition effectuée et l'état du stack, l'état du buffer, ainsi que la dépendance créée après la transition.

Comme mentionné ci-dessus, l'un des outils les plus utilisés basés sur cet algorithme est Malt de Nivre et al. (2006). Ce parser a atteint le score LAS de 88,1 et le UAS de 86,3 sur l'anglais (Nivre et al., 2007b), alors que dans le cadre de la campagne d'évaluation CoNNL-X il a obtenu les scores LAS de 85,8 sur l'allemand, 81,3 sur l'espagnol, 70,3 sur

	Transition	Stack	Buffer	Dépendance créée
0	[état initial]	*root*	Filip, studira, lingvistiku, u, Italiji	
1	shift	*root*, Filip	studira, lingvistiku, u, Italiji	
2	left-arc	*root*	studira, lingvistiku, u, Italiji	Filip ← studira
3	shift	*root*, studira	lingvistiku, u, Italiji	
4	right-arc	*root*	studira, u, Italiji	studira → lingvistiku
5	shift	*root*, studira	u, Italiji	
6	shift	*root*, studira, u	Italiji	
7	right-arc	*root*, studira	u	u → Italiji
8	right-arc	*root*	studira	studira → u
9	right-arc	-	*root*	*root* → studira
10	shift	*root*	-	

TABLE 3.12 – Illustration de l'algorithme par transitions

le slovène, et 78,4 sur le tchèque (Buchholz & Marsi, 2006). Depuis sa publication, il a connu de nombreuses extensions et optimisations (cf. Nivre, 2009a ; Ballesteros & Nivre, 2012 ; de Lhoneux et al., 2017) et a été appliqué à de nombreuses autres langues, comme le français (Candito et al., 2010b), le russe (Nivre et al., 2008), les langues indiennes (Nivre, 2009b), ou encore les langues de la campagne d'évaluation CoNLL de 2017 (Zeman et al., 2017).

En ce qui concerne l'apprentissage lui-même, les parsers par transitions traitent le parsing comme un problème de classification. Des modèles utilisés en étiquetage s'appliquent donc à cette tâche aussi. C'est le cas des SVM, appliqués au parsing en dépendances par Kudo & Matsumoto (2002) et Yamada & Matsumoto (2003). D'autres modèles sont également exploités, parfois au sein du même outil : Talismane de (Urieli, 2013) propose un choix entre le MEMM (Ratnaparkhi, 1996), le perceptron (Collins, 2002) et le SVM (Giménez & Marquez, 2004). Dans le cadre d'une évaluation sur le français, c'est le SVM qui s'est montré le plus performant (89,35 de LAS et 91,55 de UAS), mais la différence par rapport aux deux autres algorithmes était relativement faible (d'environ 1 point pour les deux scores).

Plus récemment, ces modèles d'apprentissage traditionnels ont été remplacés ou parfois combinés avec des approches en réseaux de neurones. Les implémentations réussies incluent les travaux de Chen & Manning (2014), Dyer et al. (2015), Kiperwasser & Goldberg (2016) ou encore Andor et al. (2016). Il s'agit encore de systèmes basés sur les transitions, mais ils exploitent la capacité des réseaux de neurones d'inférer les structures sous-jacentes aux

données, ce qui leur permet de manière générale de dépasser l'état de l'art posé par les méthodes linéaires. Pour illustrer, la méthode d'Andor et al. (2016) atteint 92,79 de LAS et 94,61 d'UAS sur l'anglais. La tendance de plus en plus prononcée à l'utilisation des réseaux de neurones se manifeste dans la campagne d'évaluation CoNLL de 2017, où 23 sur 28 participants disposent d'au moins un composant neuronal dans leur système (cf. Zeman et al., 2017).

L'un des points faibles principaux des parsers par transitions réside dans la nature locale de leur prise de décision. Concrètement, chaque dépendance est posée de manière isolée et les décisions déjà prises ne peuvent pas être exploitées pour informer les décisions ultérieures. Elles ne peuvent pas non plus être corrigées si des informations rencontrées plus loin indiquent que c'est nécessaire. Pour dépasser cette limitation, on a recours à la recherche par faisceau (angl. *beam search*). Cela signifie que l'algorithme considère une séquence de décisions à la fois plutôt qu'une décision seule, ce qui lui permet d'optimiser les choix faits sur une portion limitée de la phrase.

Deux autres désavantages bien identifiés de cette classe de parsers sont liés à l'ordre linéaire de la phrase et à la manière dont il est traité. Quoique les mots ne soient pas nécessairement rattachés à l'arbre dans l'ordre dans lequel ils apparaissent dans la phrase, le traitement respecte de manière globale l'ordre linéaire de la phrase. Plus particulièrement, à la différence des parsers par graphes, les outils par transitions ne considèrent pas toutes les paires de tokens possibles en posant les dépendances. Cela se traduit par un biais vers les dépendances plus courtes : leurs performances globales diminuent avec l'augmentation de la distance entre le gouverneur et le dépendant (cf. McDonald & Nivre, 2011).

Il a également été démontré par Nivre et al. (2007b) que ce type d'algorithme n'est pas capable de produire des dépendances non projectives : la nature des transitions permises par l'algorithme de base fait que seules les paires de tokens qui respectent la contrainte de projectivité sont examinées. Cette contrainte diminue la couverture des phénomènes linguistiques, mais elle réduit également la complexité temporelle de leur algorithme par rapport à celle des parsers par graphes. Par conséquent, les parsers par transitions sont en général beaucoup plus rapides.

Pour assurer le traitement des constructions non projectives, plusieurs extensions de l'algorithme de base ont été proposées. On note trois approches principales : le parsing pseudo-projectif proposé par Nivre & Nilsson (2005), l'approche de Attardi (2006) qualifiée de *non-adjacent arc transitions* (transitions sur arcs non adjacents) par Kuhlmann & Nivre (2010) et *online reordering* (réordonnancement en ligne) introduit par Nivre (2009a).

Le parsing pseudo-projectif repose sur un processus de projectivisation des données d'entraînement : le dépendant de toute relation non projective est rattaché à son ancêtre projectif le plus proche dans la structure de l'arbre. L'étiquette de chaque dépendance

Outil	Algorithme	Eval.	Langue	Corpus	Taille	#T	#D	LAS	UAS
MST	graphes	(McDonald et al., 2006)	anglais	PennTreebank	1,1 M*	48*	12*	/	90,9
(McDonald et al., 2006)		(Buchholz & Marsi, 2006)	allemand	CoNLL-X	700 K	52	46	87,3 /	
			slovène	CoNLL-X	29 K	28	25	73,4	/
			tchèque	CoNLL-X	1,2 M	63	78	80,02	/
Malt	transitions	(Nivre et al., 2007b)	anglais	PennTreebank	1,1 M	48	12	86,3	88,1
(Nivre et al., 2007b)		(Buchholz & Marsi, 2006)	allemand	CoNLL-X	700 K	52	46	85,8	/
			slovène	CoNLL-X	29 K	28	25	70,3	/
			tchèque	CoNLL-X	1,2 M	63	78	78,4	/
(Martins et al., 2009)	graphes	(Martins et al., 2009)	anglais	CoNLL2008	1,1 M	48*	12*	/	91,14
			slovène	CoNLL-X	29 K	28	25	/	85,41
(Martins et al., 2013)	graphes	(Martins et al., 2013)	anglais	CoNLL2008	1,1 M	48*	12*	/	93,33
			allemand	CoNLL-X	700 K	52	46	/	92,41
			néerlandais	CoNLL-X	195 K	302	26	/	86,19
			tchèque	CoNLL-X	1,2 M	12	49	80,96	
Mate	graphes	(Bohnet, 2010)	anglais	CoNLL2009	958 K	48	69	90,33	/
(Bohnet, 2010)			chinois	CoNLL2007	609 K	41	41	76,99	/
			tchèque	CoNLL-X	652 K	12	49	80,96	/
Talismane (Urieli, 2013)	transitions	(Urieli, 2013)	français	FTBDep	339 K	27	30	89,35	91,55
(Nivre & Nilsson, 2005)	transitions	(Kuhlmann & Nivre, 2010)	anglais	CoNLL2009	958 K	48*	69*	85,01	88,55
	+ pseudo-proj. parsing		tchèque	CoNLL2009	652 K	12	49	80,58	86,24
(Attardi, 2006)	transitions	(Kuhlmann & Nivre, 2010)	anglais	CoNLL2009	958 K	48*	69*	84,64	88,37
	+ non-adj. arc trans.		tchèque	CoNLL2009	652 K	12	49	80,64	86,24
(Nivre, 2009a)	transitions	(Kuhlmann & Nivre, 2010)	anglais	CoNLL2009	958 K	48*	69*	85,0	88,63
	+ online reordering		tchèque	CoNLL2009	652 K	12	49	80,71	86,34
(Andor et al., 2016)	feed-forward neural net.	(Andor et al., 2016)	anglais	PennTreebank	1,1 K	48	12	92,79	94,61

TABLE 3.14 – Évaluations des parsers. #T = nombre d'étiquettes morphosyntaxiques. #D = nombre d'étiquettes syntaxiques. * = information récupérée de la documentation du corpus car indisponible dans la publication citée.

modifiée est augmentée d'un suffixe permettant de garder une trace de sa nature non projective. L'entraînement est effectué sur ces données projectivisées avec l'idée que les étiquettes modifiées seront apprises et reproduites lors du parsing, ce qui permet ensuite d'appliquer sur elles une opération inverse pour rétablir les véritables dépendances non projectives. Cette solution permet à l'algorithme de garder une complexité linéaire, mais elle est soumise à une condition importante : la non-projectivité étant relativement rare, les étiquettes modifiées risquent de ne pas être assez représentées dans les données d'apprentissage.

La proposition d'Attardi (2006) se base sur l'introduction de nouvelles transitions dans l'algorithme. Plus précisément, Attardi autorise la création d'arcs entre les tokens non adjacents (séparés par un autre token) dans la pile, permettant ainsi la gestion d'un sous-ensemble de constructions non projectives. Cette solution est également de complexité temporelle linéaire.

Le réordonnancement en ligne fait lui aussi appel à une nouvelle transition, nommée *swap*. Cette transition permute les deux mots les plus récents dans le stack tout en retournant celui qui était à la deuxième position dans le buffer pour qu'il soit traité à nouveau. Cette opération permet de réordonnancer les tokens de la phrase jusqu'à ce que la non-projectivité soit résolue. Cette opération est faite en temps réel, et non pas en amont de l'apprentissage, comme c'est le cas avec le parsing pseudo-projectif. En théorie, le temps d'exécution le moins favorable pour cette approche est le temps quadratique. En pratique, le temps d'exécution attendu est linéaire, étant donné la complexité limitée des phénomènes non projectifs observés dans le langage naturel (cf. Nivre, 2009a).

Une comparaison systématique de ces trois approches sur l'anglais, le tchèque et l'allemand a été effectuée par Kuhlmann & Nivre (2010). Les résultats ont montré que les trois méthodes améliorent les scores, et qu'il y a peu de différences entre elles quant aux résultats globaux. Il en est de même quand il s'agit des performances des outils sur les relations non projectives : les trois atteignent une précision relativement élevée (70 % - 85 %), mais un rappel plus bas (50 % - 65 %). En revanche, des différences significatives ont été observées sur les dépendances projectives : la méthode d'Attardi peut mener à une chute du rappel si les dépendances non projectives sont trop longues pour être traitées par les transitions ajoutées, ce qui peut même bloquer les dépendances projectives voisines.

3.4.3 Parsing du serbe

À notre connaissance, les premières expériences en parsing statistique basées sur un corpus en serbe ont été effectuées par Jakovljević et al. (2014). Leurs évaluations sont effectuées sur un échantillon de treebank initial d'environ 7 000 tokens annoté avec un

jeu d'étiquettes proche de celui du projet Prague Dependency Treebank [9]. Plusieurs algorithmes du parser Malt ont été testés sur ces données, le plus performant atteignant 58 points en LAS et 66 points en UAS. Il semble probable que ces résultats ont été largement déterminés par la taille restreinte du corpus d'entraînement, notamment si l'on considère les résultats obtenus par Malt sur le croate dans (Berović et al., 2012) (71 points en LAS, et 84 points en UAS) sur un corpus d'entraînement de 60 000 tokens annoté avec un jeu de 80 étiquettes syntaxiques, lui aussi largement basé sur celui de PDT. Les conditions d'évaluation pour ces travaux, ainsi que pour tous les autres évoqués dans la suite, sont données dans la table récapitulative 3.16.

Comme le projet de création de treebank pour le serbe décrit dans (Jakovljević et al., 2014) n'a pas abouti à la diffusion d'un corpus complet, d'autres travaux ont exploré des pistes alternatives pour la constitution des modèles de parsing pour le serbe. Par exemple, Agić et al. (2013b) et Agić & Ljubešić (2015) exploitent la proximité prononcée du serbe et du croate et se servent exclusivement des données annotées en croate pour entraîner des parsers et les appliquer aux deux langues. Dans les deux travaux, le corpus de base est le même : il s'agit de SETimes.hr, un corpus journalistique contenant environ 87 000 tokens (Agić & Ljubešić, 2014). Dans Agić et al. (2013b), le corpus est doté d'une annotation morphosyntaxique détaillée suivant les schémas d'annotation du projet MultextEast (cf. section 2.3.1), ainsi que d'une couche d'annotation syntaxique avec un jeu d'étiquettes basé sur celui de PDT, mais largement simplifié, présenté dans (Merkler et al., 2013). Le parser utilisé est MST (McDonald et al., 2006), et l'apprentissage est effectué avec des traits du deuxième ordre et l'algorithme non projectif. L'évaluation est faite sur les deux langues, sur deux échantillons différents : le premier en domaine (sur des textes journalistiques) et l'autre hors domaine (sur des textes de Wikipédia). Sur les textes journalistiques, le parser MST obtient un score LAS de 76,7 points et un score UAS de 81,6 points sur le croate, alors que sur le serbe ses résultats sont respectivement de 75,4 points et de 80,6 points. On remarque que les scores pour les deux langues sont très proches ; ceci est le cas dans le deuxième scénario d'évaluation, avec une variation inférieure à 1,5 point en LAS et inférieure à 1 point en UAS. En revanche, la perte est plus marquée lors du passage vers les textes encyclopédiques : environ 5 points en LAS et 1,5 point en UAS (cf. tableau 3.16).

Dans un travail récent, le corpus SETimes.hr a été doté de couches d'annotation supplémentaires en accord avec les schémas d'annotation du projet UD (Agić & Ljubešić, 2015). Les auteurs reprennent le scénario d'évaluation de (Agić et al., 2013b) : ils entraînent le parser Mate de Bohnet (2010) sur le corpus croate et l'évaluent sur des échantillons en serbe et en croate. 16 scénarios d'évaluation sont mis en place en variant les paramètres suivants : schéma d'annotation syntaxique (SETimes.hr ou UD), données morphosyntaxiques exploitées (étiquettes POS ou traits morphosyntaxiques), langue d'évaluation (croate ou

9. Pour plus de détails sur l'utilisation de ce schéma d'annotation, voir la section 2.3.4.

Outil	Algorithme	Eval.	Corpus	Taille	#T	#D	LAS	UAS
Malt (Nivre, 2009a)	transitions	(Jakovljević et al., 2014)	AlfaNum	7 K	748	28	58	66
MST (McDonald et al., 2006)	graphes	(Agić et al., 2013b)	SETimes	87 K	662	15		
			en domaine	croate			76,7	81,6
				serbe			75,4	80,6
			hors domaine	croate			71,9	80,0
				serbe			72,4	80,6
Mate (Bohnet, 2010)	graphes	(Agić & Ljubešić, 2015)	SETimes	87 K	662	15		
			en domaine	croate	POS		76,3	82,2
					POS+traits		79,2	84,3
				serbe	POS		74,0	80,8
					POS+traits		77,8	83,0
			hors domaine	croate	POS		67,9	77,1
					POS+traits		73,7	80,7
				serbe	POS		71,1	79,8
					POS+traits		74,7	82,6
			Croatian UD	87 K	14	39		
			en domaine	croate	POS		77,9	84,8
					POS+traits		81,5	86,9
				serbe	POS		75,8	82,4
					POS+traits		81,5	86,0
			hors domaine	croate	POS		72,4	80,8
					POS+traits		77,3	84,5
				serbe	POS		75,2	82,1
					POS+traits		77,9	83,7

TABLE 3.16 – Expériences en parsing du serbe et du croate

serbe) et type d'échantillon d'évaluation (en domaine ou hors domaine). Les résultats obtenus avec le schéma UD représentent l'état de l'art en parsing du serbe : le parser Mate a atteint un score LAS de 81,5 points et un score UAS de 86,9 points sur l'échantillon Wikipédia (cf. tableau 3.16). Par ailleurs, ils confirment les observations de (Agić et al., 2013b) : un parser entraîné sur le croate maintient sa stabilité globale en traitant le serbe et semble plus affecté par le changement de domaine que par le changement de langue.

Avant de considérer le choix de l'outil pour notre travail, deux remarques importantes sont à faire. Premièrement, on constate que les résultats du parser Mate obtenus sur le schéma d'annotation de UD sont systématiquement plus élevés que ceux réalisés sur le schéma de SETimes. Comme l'évaluation a été faite sur les mêmes échantillons avec le même parser, cela indique que le schéma d'annotation UD facilite la tâche. Cela soulève une nouvelle fois la question du choix du schéma d'annotation pour notre treebank, qui nous a conduit à ne pas opter pour UD. Rappelons que les raisons de cette décision ont été formulées dans la section 2.3.5.

Deuxièmement, en comparant les résultats obtenus sur les étiquettes POS seules à ceux obtenus avec l'utilisation des traits morphosyntaxiques détaillés, nous constatons des améliorations importantes (jusqu'à 5 % en LAS et jusqu'à 3 % en UAS), indépendamment du schéma d'annotation et du scénario d'évaluation. Ceci confirme encore une fois l'intérêt de disposer de ce type d'informations dans le cadre du parsing du serbe.

Compte tenu de ces résultats, il aurait été logique de retenir le parser Mate pour la constitution de notre corpus. Cependant, nous avions des réserves concernant la vitesse d'exécution de ce parser. Même si Agić & Ljubešić (2015) indiquent que l'outil est rapide, ils ne donnent pas d'informations explicites sur sa vitesse d'entraînement et de parsing. Cependant, les précisions fournies dans (Bohnet, 2010) montrent que, bien que l'outil soit optimisé pour le parsing, son temps d'apprentissage reste important : à titre d'exemple, il met 44 h pour effectuer l'apprentissage sur le corpus espagnol d'environ 427 000 tokens. Même si nous ne nous attendions pas à avoir de corpus d'entraînement de taille comparable, ce rapport ne nous a pas semblé favorable. Rappelons encore une fois que la vitesse d'apprentissage est critique dans notre environnement de travail, qui prévoit plusieurs cycles d'entraînement et de parsing.

Ce fait nous a amenée à considérer l'utilisation d'un parser par transitions. Comme nous l'avons vu dans la section 3.4.2, ces outils sont plus rapides que les parsers par graphes, et différentes extensions ont été développées pour compenser leur point faible principal - le traitement des structures non projectives. Or, un parser de ce type a été développé au sein de l'équipe CLLE-ERSS : il s'agit du parser Talismane créé par Assaf Urieli dans le cadre de sa thèse (cf. Urieli, 2013). Cet outil est basé sur les algorithmes à base de transitions décrits dans (Nivre, 2008). Initialement paramétré et testé sur le français, il atteint un score LAS de 86,9 à 88,0 points et un score UAS de 89,5 à 90,4 points sur cette langue, en fonction de la configuration utilisée (Urieli, 2013, p. 154). Ces résultats sont comparables à ceux obtenus par d'autres parsers comme Berkeley (Petrov et al., 2006), MSTParser (McDonald et al., 2006), et MaltParser (Nivre et al., 2006) sur le français, dont les performances sont présentées dans (Candito et al., 2010b). Talismane intègre en effet une chaîne de traitement complète, capable d'effectuer la tokénisation, l'étiquetage morphosyntaxique et le parsing. Il permet également de définir avec précision l'exploitation de différents traits d'apprentissage (tokens, étiquettes POS, lemmes, informations morphosyntaxiques détaillées). Par ailleurs, l'outil n'utilise pas les traits morphosyntaxiques désambiguïsés du corpus d'apprentissage, mais les puise plutôt dans un lexique externe en gardant toute l'ambiguïté rencontrée. Cette particularité est censée lui assurer une meilleure robustesse dans une situation réelle où il doit traiter un texte brut.

Au-delà de toutes ces propriétés techniques, Talismane présentait également un avantage pratique important : A. Urieli maintient des liens actifs avec le laboratoire CLLE, ce qui ouvrait la possibilité d'établir un contact direct avec lui. Cette situation privilégiée

d'être en contact avec l'auteur d'un outil était très prometteuse : elle pouvait nous permettre d'avoir une meilleure prise en main ainsi qu'une compréhension plus approfondie de l'outil que l'on ne pouvait s'attendre avec d'autres outils. Pour toutes les raisons citées ci-dessus, notre choix s'est arrêté sur Talismane.

3.5 Outils sélectionnés

Ce chapitre a été dédié à une revue de méthodes de traitement automatique existantes dans l'objectif de sélectionner les outils avec lesquels nous réaliserons la préannotation automatique de notre corpus. Nos choix sont résumés dans la suite.

Étiquetage morphosyntaxique : HunPos. Cet outil basé sur un algorithme relativement simple (HMM à trigrammes) a été parmi les plus performants sur le serbe. Comme sa simplicité lui garantit également une vitesse d'apprentissage et de traitement élevée, nous l'avons préféré à BTagger (basé sur le modèle de perceptron), qui offre des performances comparables, mais s'avère beaucoup plus lent.

Lemmatisation : CST. À ce niveau de traitement aussi, un outil moins complexe l'emporte sur un outil plus complexe : CST, basé sur un système d'apprentissage hors contexte, réalise de meilleurs résultats sur le serbe que BTagger, doté d'un système d'apprentissage en contexte. L'algorithme de CST s'est également montré beaucoup plus rapide que celui de BTagger. Nous avons donc intégré ce lemmatiseur dans notre chaîne de traitement.

Parsing : Talismane. À la différence des deux tâches précédentes, le choix du parser a été motivé plus par des considérations pratiques que par les performances des outils analysés. En effet, au moment où ce choix a été fait, c'est le parser Mate qui avait obtenu les meilleurs résultats sur le serbe. Néanmoins, nous lui avons préféré un parser par transitions, *a priori* plus rapide en apprentissage. Notre choix concret s'est arrêté sur Talismane pour ses caractéristiques techniques (résultats comparables à l'état de l'art sur le français, définition précise des traits d'apprentissage, robustesse face au texte inconnu), mais aussi pour la possibilité de collaborer directement avec l'auteur de l'outil.

Ainsi, les différents éléments nécessaires pour mettre en place une méthode d'annotation de corpus outillée ont été identifiés. La suite de ce document porte sur la mise en place de la méthode concrète et sur la constitution proprement dite de notre corpus.

Deuxième partie

De la constitution des ressources au parsing du serbe

Présentation de la partie II

Cette partie décrit l'ensemble de nos activités relatives à la constitution des ressources pour le traitement automatique du serbe. Tout d'abord, nous présentons les étapes antérieures à l'annotation du corpus proprement dite : nous explicitons la méthode adoptée (cf. chapitre 4), définissions les principes d'annotation aux différents niveaux de traitement (cf. chapitre 5) et documentons la création d'une ressource auxiliaire – d'un lexique morphosyntaxique (cf. chapitre 6). Ensuite, nous évoquons la constitution du corpus proprement dite : nous détaillons la mise en œuvre de la méthode posée (cf. chapitre 7) et retraçons les campagnes d'annotation manuelle et leurs résultats (cf. chapitre 8). Enfin, nous présentons l'entraînement et le paramétrage d'un modèle de parsing focalisé sur l'exploitation des informations morphosyntaxiques disponibles dans les ressources créées (cf. chapitre 9).

Chapitre 4

Constitution du treebank : méthode adoptée

Ce chapitre est dédié à une description détaillée de la méthode que nous avons mise en place pour créer le premier treebank pour le serbe. Cette méthode découle de différentes contraintes et principes identifiés dans la partie I, que nous rappelons et systématisons dans la section 4.1. Ensuite, nous présentons la méthode globale que nous adoptons (section 4.2) et précisons comment elle s'articule pour une annotation multicouches (section 4.3). Enfin, nous identifions les différents intervenants qui participent au projet (section 4.4).

4.1 Principes de constitution de treebank retenus

Dans la partie I, nous avons identifié les principes qui vont orienter la constitution de notre treebank. Nous les résumons ici.

En ce qui concerne les propriétés du corpus et des ressources externes à constituer :

— Annotation syntaxique :

1. L'annotation syntaxique est faite en syntaxe en dépendances ;
2. Nous constituons un jeu d'étiquettes spécifique pour le serbe ;
3. Le jeu est équilibré, aussi informatif et pertinent que possible au niveau linguistique, mais basé sur des critères de surface, accessibles à un parser ;

— Annotation morphosyntaxique :

1. L'annotation morphosyntaxique contient aussi bien les parties du discours que des traits morphosyntaxiques fins ;
2. Seuls les traits morphosyntaxiques pertinents pour le parsing sont encodés ;
3. Dans le corpus final, l'annotation morphosyntaxique est décomposée en couches

(étiquettes POS, étiquettes détaillées, traits individuels), facilitant ainsi la sélection de la couche appropriée dans le cadre du parsing ;

— Lemmatisation : le corpus est lemmatisé pour diminuer la dispersion des données ;
— Lexique : nous intégrons un lexique à large couverture pour faciliter le traitement automatique.

Pour garantir la vitesse et la qualité d'annotation nécessaires, nous retenons les principes suivants quant à la constitution du treebank :

1. Une préannotation automatique est utilisée aux trois niveaux d'annotation ;
2. La qualité des guides d'annotation est vérifiée par des évaluations de l'accord inter-annotateurs ;
3. L'évolution des guides est garantie par la mise en place d'échanges réguliers avec les annotateurs ;
4. Le travail global est organisé selon les principes de l'annotation agile.

Dans la section suivante, nous montrons comment ces différents principes se concrétisent.

4.2 Méthode d'annotation agile basée sur le *bootstrapping* itératif

Le schéma global de notre méthode est donné dans la figure 4.1. Nous gardons les quatre stades de base identifiés par Fort (2012) : le travail de préparation (en bleu), la pré-campagne (en jaune), la campagne proprement dite (en vert) et la finalisation (en rouge). Cependant, la phase de la campagne est plus complexe car elle est itérative et intègre des outils automatiques. Afin de mettre en oeuvre le principe d'annotation agile, nous divisons le corpus à annoter en échantillons qui sont traités tour à tour.

Le **travail de préparation** correspond à la période de mise en place du matériel nécessaire pour l'annotation du corpus. Concrètement, il s'agit de la sélection des outils automatiques à exploiter, du choix des textes qui composent le corpus, de la définition des jeux d'étiquettes, de la constitution des guides d'annotation et de leur première évaluation sur les données, de la préparation des ressources externes (lexique), et de la préparation des ressources d'entraînement initiales pour les outils automatiques. Nous reviendrons sur ce dernier point en expliquant le stade de la campagne.

Le stade de la **pré-campagne** est dédié au recrutement et à la formation des annotateurs. Cette étape permet aux annotateurs de s'approprier les guides et les interfaces d'annotation. Notons que cette organisation diverge légèrement de celle préconisée par Fort (2012), d'après laquelle les participants au projet sont identifiés dès le premier stade.

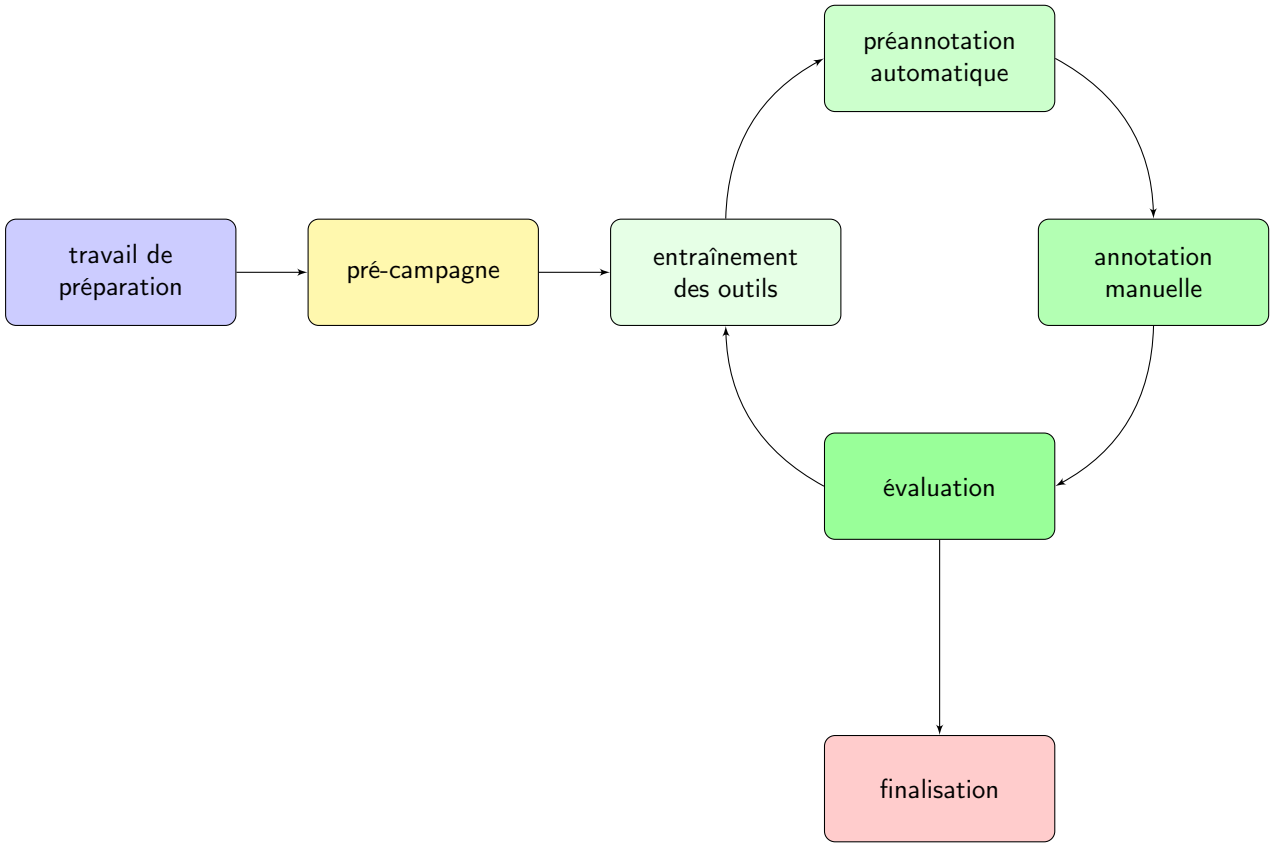


FIGURE 4.1 – Organisation globale de la méthode proposée

En revanche, il ne nous semble pas pratique de chercher à recruter des annotateurs avant d'avoir préparé tous les éléments nécessaires à la campagne, et d'autant plus que le travail de préparation est long. Pour cette raison, nous déplaçons le recrutement des annotateurs dans le stade qui précède immédiatement la campagne.

L'organisation de la **campagne** dans notre méthode est conditionnée par deux principes : l'agilité et l'utilisation d'outils automatiques. Le premier impose une organisation itérative du travail et introduit une étape d'évaluation à la fin de chaque cycle d'annotation manuelle. Le deuxième introduit deux étapes supplémentaires en début de chaque cycle : l'entraînement des outils et la préannotation automatique. Notons que ces deux étapes sont également exécutées itérativement par le recours au *bootstrapping* (cf. section 2.5). Lors du premier passage par la boucle, l'entraînement des outils est effectué sur les ressources d'apprentissage minimales constituées manuellement dans le stade du travail de préparation (v. *supra*). Ces premiers modèles sont ensuite utilisés pour la préannotation du premier échantillon du corpus, la préannotation automatique est corrigée manuellement, et ensuite l'échantillon nouvellement validé est rajouté aux ressources d'entraînement initiales. Lors du prochain passage par la boucle, les outils automatiques sont entraînés sur ces ressources augmentées, ce qui leur permet de s'améliorer avec chaque itération, et cela facilite à son tour l'étape d'annotation manuelle.

Quant à l'étape d'**évaluation**, dans notre méthode elle diffère de ce qui est préconisé par Voormann & Gut (2008). Étant donné le temps nécessaire pour effectuer systématiquement des évaluations de l'accord inter-annotateurs, nous ne les intégrons pas dans ce cycle. La qualité du travail des annotateurs est vérifiée à travers un contrôle ponctuel, de la part d'un annotateur expérimenté, des annotations produites. Ces contrôles sont focalisés sur les points identifiés comme problématiques par les annotateurs eux-mêmes ou dans le stade d'évaluation des guides d'annotation. Nous cherchons également à contrôler la qualité de l'annotation en faisant travailler les annotateurs dans un espace commun et en présence d'un annotateur expérimenté. Toute difficulté est donc soulevée, discutée et résolue en temps réel.

L'étape d'évaluation est également consacrée à un retour d'expérience des annotateurs : une séance de travail est organisée pour discuter des difficultés rencontrées avec les annotateurs et recueillir leurs remarques relatives aux guides d'annotation. Si les problèmes identifiés sont systématiques et suffisamment importants, les guides sont modifiés de sorte à les prendre en compte. Pour éviter les incohérences qu'une telle démarche peut introduire dans l'annotation, deux solutions sont possibles. Premièrement, on peut envisager une étape d'harmonisation des annotations déjà produites immédiatement après la modification des guides, et avant d'entamer l'itération suivante. Alternativement, on peut garder ce travail pour la phase de la finalisation du corpus : dans ce cas, l'harmonisation des annotations s'effectue selon la dernière version des guides, et elle peut être confiée aux

mêmes annotateurs, ou à l'annotateur expérimenté. Dans le cadre de cette thèse, nous optons pour cette dernière option, qui permet d'effectuer toutes les modifications nécessaires à la fois.

La **finalisation** du corpus comprend donc le travail d'harmonisation des annotations mentionné ci-dessus. Elle porte également sur toutes les activités nécessaires à la diffusion du corpus : la conversion du corpus vers un format de diffusion standard, l'élaboration d'une documentation, la diffusion du corpus proprement dite.

Notons encore qu'une telle organisation du processus permet d'avoir un livrable à la fin de chaque cycle d'annotation. Il est donc possible d'interrompre le processus à ce moment. Ceci est particulièrement utile pour les projets avec des contraintes de temps importantes : si l'on ne réussit pas à traiter la totalité du corpus dans les délais impartis, cette démarche garantit qu'on obtiendra une ressource de taille plus petite mais ayant l'ensemble des traitements envisagés.

Le schéma présenté concerne une seule couche d'annotation. La section suivante explique l'organisation de l'annotation multicouches.

4.3 Bootstrapping itératif à trois niveaux

Une première façon d'organiser l'annotation multicouches consisterait à transmettre la sortie d'un outil directement à l'outil suivant, et d'effectuer simultanément la correction manuelle de toutes les couches. Cependant, l'apprentissage se fait sur des ressources restreintes, au moins dans les premières itérations du processus, ce qui signifie que les performances des outils ne seront pas élevées. Si par ailleurs l'annotation de l'outil précédent est de piètre qualité, cela remet en question la capacité de l'outil de fournir une base exploitable aux annotateurs humains. Pour contrer cet effet, nous introduisons une étape de correction manuelle suite à chaque étape du prétraitement afin de produire des annotations fiables et faciliter le travail des annotateurs humains. Le processus est schématisé dans la figure 4.2.

Le processus commence donc par un entraînement des trois outils et les trois niveaux de traitement sont effectués en cascade. Étant donné le rapport entre ces trois tâches décrit dans la section 3.1, c'est d'abord l'étiquetage morphosyntaxique qui est fait, suivi de la lemmatisation, puis du parsing.

Ce schéma correspond donc aux cases relatives à l'entraînement des outils, à la préannotation de l'échantillon et à la correction manuelle de la figure 4.1 ; chaque passage par la boucle représentée ici est suivi par l'étape d'évaluation de la figure 4.1. En revanche, l'annotation manuelle du corpus d'entraînement initial est faite dans le cadre du travail de préparation.

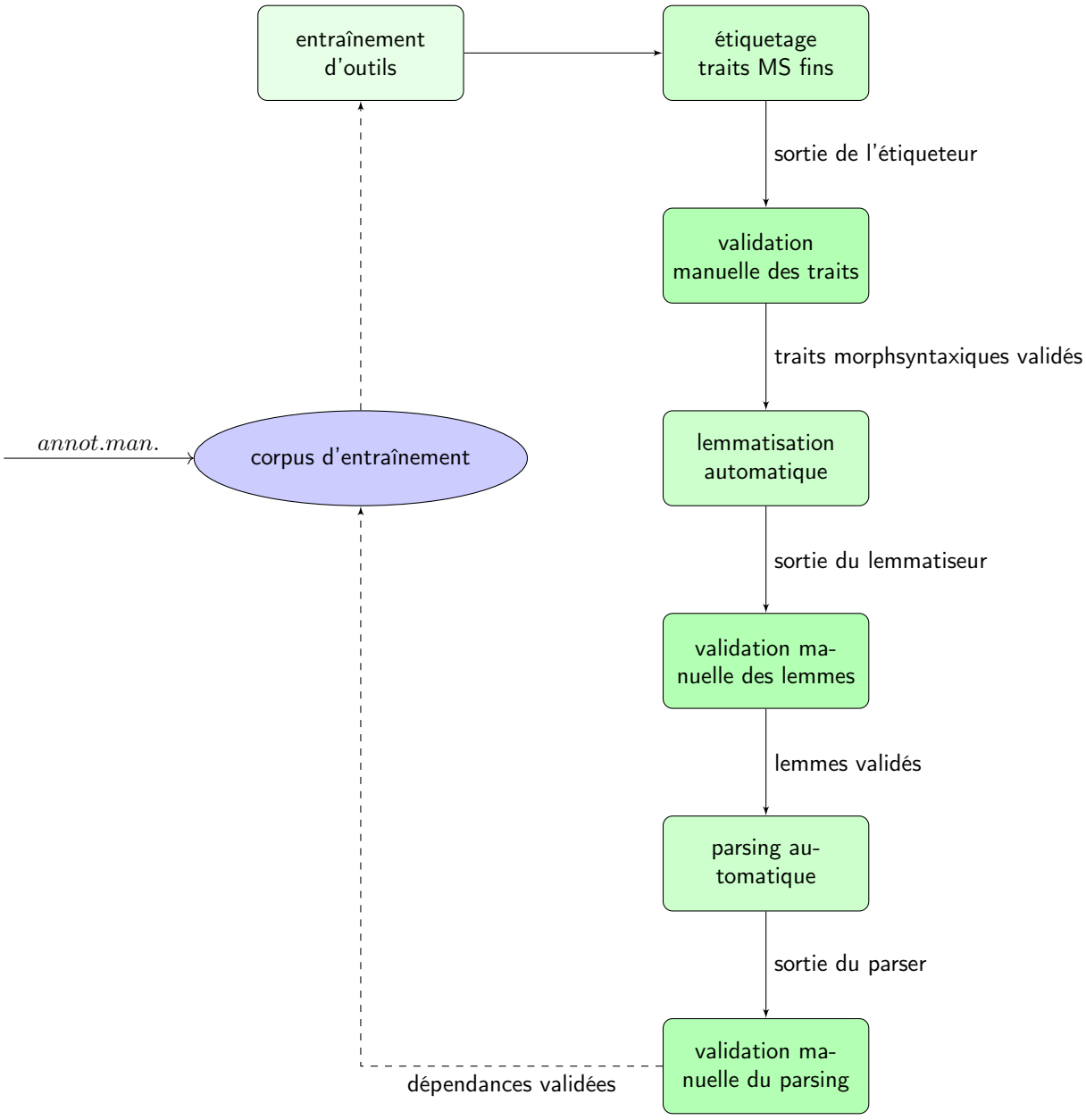


FIGURE 4.2 – *Bootstrapping* itératif pour une annotation multicouches

4.4 Participants au projet

Le processus décrit ci-dessus repose sur trois types de compétences : en linguistique, en TAL et en gestion. Ces différents types de compétences peuvent se répartir entre différents rôles.

Le **gestionnaire de la campagne** effectue la répartition et l'organisation matérielle du travail. Il s'assure également que les délais définis soit respectés. Son rôle peut inclure aussi la mise en place du recrutement des annotateurs et l'organisation des aspects pratiques des campagnes d'annotation. C'est également le gestionnaire qui assure la diffusion du corpus une fois le travail complété.

Le ou les **annotateur(s) expert(s)** a/ont des tâches à différents niveaux de la méthode. Dans la phase de préparation, ils définissent les jeux d'étiquettes, constituent les guides d'annotation et évaluent leur qualité. Ils sont également responsables de la création des ressources d'entraînement initial pour les outils automatiques. Dans le stade de la pré-campagne, ils sélectionnent les annotateurs qui effectueront l'annotation manuelle et dispensent une formation pour eux. Dans le stade d'annotation, les annotateurs experts surveillent le travail des autres annotateurs et contrôlent la qualité de l'annotation manuelle produite. Ils décident également de la modification éventuelle des guides d'anntoation en fonction du retour des annotateurs chargés de l'annotation manuelle. Enfin, dans le stade de finalisation, les annotateurs experts effectuent l'harmonisation de l'annotation si nécessaire.

Les **annotateurs chargés de l'annotation manuelle** ont le rôle suivant : en se basant sur les guides d'annotation fournis par les annotateurs experts, ils corrigent la préannotation automatique effectuée par les outils. Dans un projet avec une annotation multicouches, leur tâche exacte dépend de la couche d'annotation qu'ils sont en train de traiter. Dans l'étape de l'évaluation, ils remontent à l'annotateur expert leurs remarques quant à l'adaptation des guides d'annotation.

Le **taliste** intervient également à plusieurs stades de la méthode. Dans le travail de préparation, cette personne sélectionne les outils qui seront utilisés, et durant la campagne, elle assure leur fonctionnement, aussi bien en ce qui concerne la préannotation automatique que le ré-entraînement itératif. Enfin, dans le stade de finalisation, le taliste peut être chargé de la conversion du corpus vers un format de diffusion standard.

Certains de ces différents rôles peuvent être assurés par la même personne : à titre d'illustration, nous avons nous-mêmes assumé ceux de linguiste expert, taliste et gestionnaire, et pour une partie du corpus, nous étions également l'annotateur principal.

Dans la suite de cette partie nous décrivons comment nous avons appliqué la méthode décrite ici pour réaliser notre premier objectif : constituer un treebank pour le serbe.

Chapitre 5

Définition des jeux d'étiquettes et des schémas d'annotation

Comme expliqué dans le chapitre précédent, les jeux d'étiquettes et les guides d'annotation font partie des éléments à fournir dans le cadre du travail de préparation (stade 1 d'une campagne d'annotation), et il revient aux annotateurs experts d'effectuer ce travail. Durant cette étape, le rôle des annotateurs experts a été assuré par nous-mêmes et par D. Stosic.

Pour définir les jeux d'étiquettes morphosyntaxiques et syntaxiques, nous nous servons de la tradition grammaticale serbe comme point de départ. Néanmoins, nous soumettons les traitements traditionnels de différents phénomènes à un examen détaillé théorique et empirique. Si ces traitements ne se basent pas sur des critères explicites, accessibles aussi bien à un annotateur humain qu'à un parser, nous les modifions de sorte à satisfaire cette exigence. Ceci est fait dans un souci de garantir que les annotateurs humains pourront apporter des annotations cohérentes et que les parsers pourront distinguer les étiquettes retenues. En définissant les jeux d'étiquettes, nous posons également les règles de leur utilisation, et ces schémas d'annotation sont transformés en guides d'annotation destinés aux annotateurs humains.

Les détails de ce travail sur l'annotation morphosyntaxique sont présentés dans la section 5.1, alors que ceux relatifs à l'annotation syntaxique sont donnés dans la section 5.2. Vu la relative simplicité de la tâche de lemmatisation, les principes adoptés pour ce niveau de traitement sont rapidement exposés dans la section 5.3.

5.1 Jeu d'étiquettes morphosyntaxiques et schéma d'annotation

Comme nous l'avons vu dans le chapitre 4, nous cherchons à établir une annotation morphosyntaxique dotée de traits morphosyntaxiques fins. À la différence du jeu d'étiquettes détaillé existant dédié au serbe (Krstev et al., 2004b), nous ne visons pas l'exhaustivité dans l'encodage de ces traits : nous nous concentrons sur ceux qui sont le plus à même d'être impliqués dans le fonctionnement syntaxique du serbe. Ainsi, notre objectif est d'établir une annotation morphosyntaxique raisonnée, qui assure les informations nécessaires pour le traitement syntaxique sans pour autant la rendre trop complexe pour l'annotation manuelle. Le schéma d'annotation est majoritairement basé sur des grammaires serbes de référence (Stanojčić & Popović, 2012 ; Ivić, 2005), mais il contient également quelques traitements modifiés, adoptés déjà dans le travail de (Miletic, 2013).

5.1.1 Jeu d'étiquettes retenu

Un jeu d'étiquettes morphosyntaxiques détaillé pour le serbe a été développé dans le cadre du projet MultextEast (Krstev et al., 2004b) [1]. Ce jeu intègre tous les traits morphosyntaxiques traditionnellement reconnus en serbe, ainsi que certaines distinctions sémantiques : en plus des 12 parties du discours retenues par les auteurs, le jeu encode 21 traits morphosyntaxiques différents. Ces informations sont représentées sous la forme d'un jeu d'étiquettes positionnel (cf. section 2.3.1), qui contient 1243 étiquettes uniques au total. Cependant, l'utilité de certains des traits pour le parsing peut être questionnée. Il s'agit notamment de la définitude des adjectifs, de l'aspect verbal et de l'opposition animé/non animé. En ce qui concerne la définitude des adjectifs et l'aspect verbal, nous avons déjà discuté le fait que leur encodage n'apporte pas d'informations utiles au parsing, alors qu'il augmente la taille du jeu d'étiquettes (cf. respectivement sections 1.1.4 et 1.1.5). La distinction animé/non animé représente un cas de figure semblable : au niveau morphologique, elle est marquée seulement dans les formes de l'accusatif des noms masculins, et elle n'a pas de rôle au plan syntaxique.

Dans la constitution de notre jeu d'étiquettes, nous éliminons ce type de distinctions. Nous nous focalisons sur les traits morphosyntaxiques impliqués dans le fonctionnement syntaxique de la langue. Le tableau 5.2 présente les traits retenus pour chaque partie du discours.

Nous utilisons la sous-catégorie grammaticale (cf. nom propre *vs* nom commun, pronom personnel *vs* possessif *vs* indéfini, etc.) car elle est indicative de différents types de comportement syntaxique, notamment pour les noms, les pronoms et les adjectifs (cf. sec-

1. Une description détaillée est également disponible à l'adresse suivante :`http://nl.ijs.si/ME/V4/msd/html/msd-sr.html`. Dernier accès : le 10 décembre 2017.

Partie du discours	**Traits encodés**
Adjectif	Partie du discours, sous-catégorie, cas, nombre, genre, degré de comparaison
Nom	Partie du discours, sous-catégorie, cas, nombre, genre
Numéral	Partie du discours, sous-catégorie, cas, nombre, genre
Pronom	Partie du discours, sous-catégorie, cas, personne, nombre, genre
Verbe	Partie du discours, sous-catégorie, forme, personne, nombre, genre
Adverbe	Partie du discours, sous-catégorie, degré de comparaison
Conjonction	Partie du discours, sous-catégorie
Interjection	Partie du discours
Particule	Partie du discours
Préposition	Partie du discours

TABLE 5.2 – Traits morphosyntaxiques encodés dans ParCoLab

tion 5.1.2). À titre d'illustration, les adjectifs qualificatifs sont typiquement antéposés au nom (***puna*** *kuća* 'maison **pleine**'), mais ils peuvent également être postposés, surtout s'ils sont accompagnés d'un dépendant (*kuća* ***puna*** *dece* lit. 'maison pleine enfants.GEN', 'maison pleine d'enfants). Cette configuration est impossible pour les autres sous-catégories adjectivales (les possessifs, les démonstratifs, les indéfinis, les interrogatifs et les relatifs) [2].

Nous encodons le cas pour son rôle dans le marquage des fonctions syntaxiques, et les autres traits d'accord (personne, nombre, genre) car ils facilitent l'identification de la tête des formes qui s'accordent (typiquement les adjectifs). La manière dont ces traits permettent le décodage de certaines fonctions syntaxiques a été illustrée dans la section 1.1.2.

Le degré de comparaison a été retenu pour prendre en compte les constructions comparatives complexes auxquelles les adjectifs et les adverbes peuvent prendre part. Considérons la phrase suivante : *Pevao je* ***lepše*** *nego Marko* 'Il chantait **mieux** que Marko'. Ici, c'est l'adverbe au comparatif *lepše* 'mieux' qui licencie la structure comparative en *nego* 'que'. La même structure est impossible avec un adverbe au positif : **Pevao je* ***lepo*** *nego Marko* 'Il chantait bien que Marko'.

Quant à la forme verbale, elle nous permet de faire la distinction entre les formes finies et non finies et d'identifier les propositions participiales et infinitivales de manière non ambiguë.

Les valeurs possibles pour chacun des traits, des exemples de combinaisons de valeurs valides, ainsi que les règles de leurs utilisations sont disponibles dans le guide d'annotation présenté dans l'annexe A (tome 2).

Notons encore que dans le cadre de l'annotation manuelle, ces informations sont ap-

2. Ce phénomène est étudié en détail dans le chapitre 10.

portées indépendamment les unes des autres, comme des traits atomiques (cf. section 8.4). Elles sont ensuite converties en annotations différentes et réparties sur trois couches : les étiquettes POS de base (limitées à la partie du discours), les étiquettes morphosyntaxiques détaillées (combinant tous les traits annotés), et les traits morphosyntaxiques, où l'on indique les traits flexionnels du cas, du nombre, du genre et de la personne de manière indépendante (cf. section 8.8). Si l'on considère les étiquettes morphosyntaxiques détaillées, les informations retenues constituent un jeu de 1042 étiquettes différentes possibles.

À ce niveau de traitement, nous ne prenons pas en compte les unités polylexicales. Certaines d'entre elles – et notamment les locutions grammaticales – sont annotées au niveau syntaxique (cf. tableau 5.3, étiquette Polylex). Ce principe a été hérité du corpus ParCoTrain (Miletic, 2013), dont nous nous servons dans le cadre de cette thèse (cf. section 7.1). Nous sommes néanmoins consciente qu'un traitement de ces unités au niveau de la tokénisation peut être bénéfique aux applications du corpus en TAL. Nous considérerons donc cette piste dans la continuation de ce travail.

5.1.2 Particularités du schéma d'annotation

Le schéma d'annotation que nous avons défini diffère sur deux points principaux de la grammaire traditionnelle serbe et, par conséquent, du schéma du projet MultextEast. Il s'agit du traitement des formes dites « pronoms adjectivaux » et des formes verbales en *-t/n* appelées *glagolski pridev trpni* 'adjectif verbal passif' et de celles en *-o* appelées *glagolski pridev radni* 'adjectif verbal actif'. Dans la suite de ce document, ces formes verbales seront dénommées simplement participe passif et participe actif. La nature problématique du traitement traditionnel de ces deux types de formes a été identifiée dès notre travail de 2013 (Miletic, 2013), et les mêmes solutions ont été reprises ici.

La première question concerne les formes des possessifs, démonstratifs, indéfinis, interrogatifs et relatifs, traditionnellement considérés comme des pronoms (Stanojčić & Popović, 2012, p. 101-102). Or, comme illustré dans la section 1.1.3, ces formes disposent de deux patrons de comportement distincts : elles peuvent remplacer un groupe nominal et fonctionner effectivement comme des pronoms, mais elles peuvent également se trouver à l'intérieur d'un groupe nominal, et dans ce cas leur comportement peut être rapproché de celui des déterminants en anglais ou en français. Cependant, nous avons déjà indiqué que la classe de déterminants n'est pas reconnue dans la tradition grammaticale serbe (la seule exception étant le travail de (Mrazović, 2009)). Dans une tentative de compromis, nous traitons ces formes comme des adjectifs, en marquant systématiquement leurs sous-catégories respectives (cf. le guide d'annotation morphosyntaxique, tome 2, annexe A).

La deuxième modification a été adoptée pour des raisons de cohérence. Les formes verbales en *-t/n* comme *otvoren* 'ouvert' et celles en *-o* comme *zalutao* 'égaré' sont utilisées

dans deux contextes différents. Elles peuvent faire partie des formes verbales composées en tant que participes (cf. *Mače je zalutalo* 'Le chaton s'est égaré') ou bien être antéposées au nom et avoir un rôle de modifieur (*zalutalo mače* 'chaton égaré'). La distinction entre les emplois verbaux et adjectivaux est ardue quand la forme apparaît dans une phrase avec le verbe *jesam* 'être', comme c'est le cas des exemples comme *Hotel je otvoren* 'L'hôtel est ouvert', *Ključ je izgubljen* 'La clé est perdue', etc. Cette combinaison des formes peut être interprétée soit comme un passif, où la forme ambiguë correspondrait à un participe (et devrait donc être traitée comme un verbe), soit comme un emploi attributif du verbe *être*, où la forme ambiguë serait plutôt un adjectif employé comme attribut du sujet. Pour éviter de faire des distinctions sémantiques, nous introduisons un critère de désambiguïsation artificiel : toute occurrence de ces formes accompagnée du verbe 'jesam' *être* est annotée comme participe (donc, comme un verbe), alors que les autres emplois, indépendants du verbe *être*, sont systématiquement traités comme adjectifs. Par conséquent, la forme *otvoren* 'ouvert' dans l'exemple *Hotel je otvoren* 'L'hôtel est ouvert' est considérée comme un participe, alors que dans l'exemple *otvoren hotel* 'l'/un hôtel ouvert', elle est annotée comme adjectif.

Les principes d'annotation formulés ont été transformés en règles de traitement détaillées sous forme d'un guide d'annotation pour la morphosyntaxe. Ce guide a été élaboré dans un premier temps à partir de considérations théoriques, pour être ensuite retravaillé et complété lors de l'annotation morphosyntaxique du premier échantillon de 20 000 tokens. Cette mise au point a été complétée par une évaluation de l'accord inter-annotateurs, qui nous a permis d'identifier et corriger les points problématiques. Cette démarche est présentée en détail dans la section 7.5.1.

5.2 Jeu d'étiquettes syntaxiques et schéma d'annotation

Dans la constitution du jeu d'étiquettes syntaxiques, nos décisions ont été motivées par plusieurs principes différents. D'abord, ce sont les travaux de référence en syntaxe théorique du serbe, notamment (Stanojčić & Popović, 2012) et (Ivić, 2005), qui nous ont servi de point de départ pour la sélection des étiquettes. Il fallait cependant s'assurer que les distinctions faites dans ces ouvrages étaient pertinentes pour un travail en corpus. Pour ce faire, nous avons établi une liste de critères distinctifs des relations syntaxiques inspirée des travaux de Burga et al. (2011) et Mille et al. (2012), à leur tour basés sur la Théorie Sens-Texte d'I. Mel'čuk (Mel'čuk, 1988). Nous nous en sommes servie pour examiner la pertinence des relations syntaxiques recensées dans les ouvrages théoriques. Cet examen théorique a été complété par une revue des traitements mis en place dans des treebanks existants.

5.2.1 Théorie Sens-Texte : propriétés distinctives des relations

Dans la partie consacrée à la syntaxe de surface de la Théorie Sens-Texte (TST), I. Mel'čuk propose des critères pour identifier les relations syntaxiques de surface (dorénavant RSS) : il formule des principes pour déterminer si deux formes sont reliées par une RSS, pour distinguer le gouverneur et le dépendant dans une RSS, et pour déterminer si deux relations de dépendance relèvent de la même RSS ou non.

Pour pouvoir considérer que deux formes sont reliées par une relation syntaxique directe, la TST définit :

— le critère de **linéarité**, qui postule que la position dans la phrase de l'une des deux formes est déterminée par rapport à la position de l'autre si une relation de dépendance existe entre elles (l'une des formes doit se positionner à gauche ou à droite de l'autre), et

— le critère d'**unité prosodique** : deux formes liées par une dépendance font une unité prosodique, ou bien l'une des formes peut être liée prosodiquement avec une unité prosodique dont l'autre forme est la tête (Mel'čuk, 1988, pp. 129-132).

En ce qui concerne la direction de la relation, entre deux formes reliées par une relation syntaxique, le gouverneur est :

— la forme qui détermine la **valence passive du syntagme** (son schéma distributionnel), ou

— dans les cas où les deux formes appartiennent à la même catégorie grammaticale (et ont par conséquent les mêmes propriétés distributionnelles), le gouverneur est la forme qui représente le **point de contact morphologique** avec le contexte (*ibid*, pp. 133-138).

Mel'čuk propose également deux autres principes concernant l'orientation de la relation syntaxique : l'**omissibilité** (en règle générale, c'est le dépendant qui peut être omis de l'arbre syntaxique, et non pas le gouverneur) et la **prédictibilité** (le dépendant permet de « prédire » la présence de son gouverneur ; par exemple, un déterminant « prédit » la présence d'un nom). Il souligne cependant que ces critères ne sont pas infaillibles et leur accorde le statut d'heuristique utile, à utiliser avec précaution (*ibid*, pp. 139-140).

Enfin, pour considérer que deux relations de dépendance relèvent de la même fonction syntaxique, l'un ou plusieurs des critères suivants doivent être respectés :

— **Test des paires minimales** : une même relation de dépendance ne peut pas décrire deux constructions différentes basées sur les mêmes lexèmes qui exhibent un contraste sémantique tout en ne se distinguant que par un élément de nature syntaxique (cf. *the visible stars* 'les étoiles qui sont visibles en principe' *vs the stars visible* 'les étoiles qui sont visibles en ce moment') (cf. Mel'čuk, 2009, pp. 34-35) ;

— **Caractère interchangeable des sous-arbres** : si deux relations de dépendance relèvent de la même fonction syntaxique, il doit être possible de remplacer le dépendant de l'une par le dépendant (ou le sous-arbre dont le dépendant est la tête) de l'autre. Dans une variante moins stricte, ce critère exige que pour chaque relation de dépendance il existe un dépendant prototypique qui est adapté à chaque gouverneur possible de cette relation (cf. Mel'čuk, 2009, pp. 35-37).
— **Caractère répétable du dépendant** : le dépendant d'une relation peut être soit répétable avec n'importe quel gouverneur, soit non répétable ; si les deux paires de dépendances considérées n'ont pas le même comportement par rapport à ce critère, il s'agit de relations différentes (cf. Mel'čuk, 2009, pp. 37-39).

Ces critères généraux sont des outils qui permettent d'analyser les dépendances rencontrées, mais ils ne sont pas suffisants pour dresser un inventaire des RSS d'une langue. Ceci est en grande partie dû à la nature même des RSS : à la différence des relations de syntaxe profonde, qui peuvent être considérées comme indépendantes de la langue, l'ensemble des RSS est spécifique à chaque langue donnée. Le travail d'identification des relations syntaxiques doit donc être effectué pour chaque langue indépendamment des autres, et exige l'application de critères plus spécifiques, eux aussi dépendants de la langue, qui s'ajoutent aux critères de base présentés ci-dessus.

Cette approche a été utilisée par Mel'čuk et ses différents collaborateurs afin d'identifier les relations syntaxiques de surface dans différentes langues, notamment en russe (Mel'čuk, 1995), en anglais (Mel'čuk & Pertsov, 1987 ; Mel'čuk, 2003) et en français (Iordanskaja & Mel'čuk, 2009). Dans ce dernier travail, par exemple, les auteurs dérivent un ensemble de critères plus spécifiques qui leur permettent de comparer les différentes relations qui s'établissent entre un verbe et ses dépendants argumentaux en français. Ces critères relèvent des propriétés du gouverneur et du dépendant de la relation en question. Ils peuvent être syntaxico-sémantiques (le dépendant fait partie de la structure argumentale du verbe ou non), purement syntaxiques (caractère obligatoire ou non du dépendant, possibilité de pronominalisation par un clitique, implication dans la promotion/démotion, etc.), ou syntaxico-morphologiques (le fait d'être la cible de l'accord morphologique, le fait de gouverner l'accord du verbe principal, etc.) (p. 157-160). A travers cette méthode, les auteurs examinent différentes relations de dépendances de manière systématique (en utilisant le même ensemble de paramètres) et établissent ainsi une liste cohérente des relations syntaxiques qui peuvent exister entre un verbe et son argument en français.

La même approche a été mise en œuvre dans les travaux de (Burga et al., 2011 ; Mille et al., 2012). Pour garantir l'adaptation de cette méthode au travail en corpus, Burga et al. (2011) posent les contraintes supplémentaires suivantes sur l'élaboration des critères :

— les critères doivent exploiter exclusivement des informations de surface (syntaxiques

ou morphologiques) ;
— ils doivent être faciles à reconnaître et ne peuvent pas être trop nombreux, afin d'assurer une implémentation réussie dans le cadre d'une annotation manuelle.

Les résultats des évaluations effectuées, décrits dans la section 2.3.3, ont montré qu'il est possible d'augmenter la granularité d'un jeu d'étiquettes syntaxiques de manière importante sans remettre en cause la qualité du parsing. Comme ces conclusions coïncident parfaitement avec notre double objectif d'établir une annotation aussi détaillée que possible du point de vue linguistique mais capable d'assurer un parsing performant, nous adoptons le même procédé général. Nous reprenons donc la liste des critères d'identification des relations syntaxiques proposée par les auteurs et l'adaptons au serbe. L'ensemble des critères retenus est donné ci-dessous.

Liste des propriétés utilisées dans l'identification des relations syntaxiques

— catégories morphosyntaxiques et lemmes possibles du gouverneur et du dépendant ;
— flexion du dépendant et du gouverneur si des traits morphosyntaxiques spécifiques sont liés à la fonction en question (par exemple, un cas ou une forme verbale spécifique) ;
— pour les dépendants verbaux considérés comme objets, la possibilité de pronominalisation avec un clitique et le type de clitique utilisé ;
— accord : les constituants et les traits concernés ;
— règles de linéarisation :
  — ordre canonique gouverneur - dépendant ;
  — caractère flexible ou rigide de l'ordre gouverneur - dépendant ;
  — caractère obligatoire ou non de l'adjacence du gouverneur et du dépendant (possibilité que le dépendant soit séparé du gouverneur par un autre constituant) ;
  — possibilité que la relation soit non projective.

Nous nous servons de ces propriétés dans deux perspectives : pour évaluer la cohérence d'une étiquette et pour estimer sa démarcation par rapport à d'autres relations. Autrement dit, si les constructions syntaxiques qui sont censées être regroupées sous une même étiquette présentent des valeurs de ces propriétés très hétérogènes, c'était un indicateur que la relation était mal définie et qu'elle recouvre en effet plusieurs relations différentes. Dans ce cas, on cherchait à les dégager et à leur accorder des traitements appropriés. Et deuxièmement, si les valeurs des propriétés analysées pour une étiquette donnée ne permettaient pas de la distinguer nettement par rapport aux autres relations du jeu, nous considérions qu'il s'agissait d'une augmentation de granularité non justifiée du point de vue linguistique et qui risquait en outre de causer des difficultés dans le cadre du traitement automatique. Nous introduisions donc une étiquette plus globale, regroupant les

phénomènes ayant des propriétés similaires.

Cette démarche a bien évidemment ses limites : comme nous le verrons à la fin de cette section, les propriétés listées ci-dessus se sont montrées insuffisantes pour l'analyse de certaines relations, notamment des dépendants nominaux et des phénomènes syntaxiques complexes comme la coordination et l'ellipse. Il ne faut cependant pas remettre en cause l'utilité de ce procédé : il nous a permis de confirmer la pertinence de certaines distinctions présentes dans la tradition grammaticale serbe, mais abandonnées dans certains corpus, notamment dans les treebanks croates HOBS (Agić et al., 2014) et SETimes (Agić & Ljubešić, 2014).

5.2.2 Jeu d'étiquettes syntaxiques retenu

Ce procédé aboutit à un jeu de 48 relations syntaxiques de base, et un traitement spécifique pour l'ellipse, qui sera détaillé dans la suite (cf. section 5.2.10). Le jeu est présenté dans le tableau 5.3. Chaque étiquette est accompagnée d'une brève définition de la relation et d'un exemple. Dans l'exemple, le **gouverneur** est indiqué en gras, et le dépendant en souligné.

Avant de continuer, rappelons encore que dans le cadre de la syntaxe en dépendances, les relations s'établissent entre les mots individuels. Par conséquent, dans les exemples donnés ci-dessous, le dépendant correspond toujours à un seul mot. Dans le cadre de la syntaxe en constituants, ce mot correspondrait à la tête du constituant exerçant la fonction syntaxique en question. C'est pourquoi, dans la première ligne du tableau, nous annotons seulement la forme *predsednik* 'président' en tant qu'apposition, et non pas tout le syntagme *predsednik Francuske* 'président de la France'. Précisons encore que là où la traduction en français ne permet pas de transposer les mêmes relations syntaxiques qu'en serbe, nous introduisons des gloses à leur place. Enfin, ce tableau propose une présentation globale du jeu d'étiquettes. La signification exacte et le domaine d'application de chacune d'entre elles sont présentés dans le tome 2 de cette thèse.

Table 5.3 – Jeu d'étiquettes syntaxiques ParCoLab

Etiquette	Définition	Exemple
Ap	apposition	***Oland***, *predsednik Francuske* '**Hollande**, président de la France'
AuxV	verbe auxiliaire dans une forme verbale composée	*Filip je* ***stigao*** 'Filip est **arrivé**'

TABLE 5.3 – Jeu d'étiquettes syntaxiques ParCoLab

Etiquette	Définition	Exemple
Cit	élément métalinguistique	***misao*** *« budan sam »* '**idée** « je suis réveillé »'
ComplNum	complément d'un numéral sous forme du paucal [3]	***dva*** *čoveka* '**deux** *hommes*'
ComplPrep	complément de préposition	*kolač* ***od*** *višanja* 'gâteau **aux** cerises'
ConjCoord [4]	relation entre le coordonné précédant immédiatement la conjonction de coordination et la conjonction elle-même	*Filip je* ***vredan*** *i pametan* 'Filip est **travailleur** et intelligent'
Coord	relation entre la conjonction de coordination et le dernier coordonné	*Filip je vredan* ***i*** *pametan* 'Filip est travailleur **et** intelligent'
Correl	relation entre deux éléments d'une structure corrélative	***tako*** *vruće da peče* '**si** chaud que ça brûle'
DepAdjAdv	dépendant d'un adjectif sous forme d'un adverbe	*Jedva* ***vidljiv*** *i sasvim* ***tih*** 'A peine **visible** et complètement **silencieux**'
DepAdjCas	dépendant d'un adjectif sous forme d'un nom fléchi	***sličan*** *ocu* '**semblable** père.DAT'
DepAdjPrep	dépendant d'un adjectif sous forme d'un groupe prépositionnel	*On je* ***zaljubljen*** *u Milicu* 'Il est **amoureux** de Milica'
DepAdvAdv	dépendant d'un adverbe sous forme d'un adverbe [5]	*još* ***dugo*** 'encore **longtemps**'
DepAdvCas	dépendant d'un adverbe sous forme d'un nom fléchi	***mnogo*** *ljudi* '**beaucoup** gens.GEN'
DepAdvPrep	dépendant d'un adverbe sous forme d'une préposition	***više*** *od njega* '**plus** *que* lui'
DepEx_	ellipse : préfixe ajouté à l'étiquette de l'élément dont le gouverneur est élidé [6]	(cf. section 5.2.10)

TABLE 5.3 – Jeu d'étiquettes syntaxiques ParCoLab

Etiquette	Définition	Exemple
DepNAdj	dépendant de nom sous forme d'un adjectif	*dugo* ***pismo*** ' longue **lettre**'
DepNCas	dépendant de nom sous forme d'un nom fléchi	***zrak*** *sunca* '**rayon** soleil.GEN'
DepNPrep	dépendant de nom sous forme d'un groupe prépositionnel	***pismo*** *od Filipa* '**lettre** de Filip'
DepVCas	dépendant d'un verbe sous forme d'un nom fléchi et qui n'est pas un ObjDir, ObjIndir ou prédicatif	***Plaši*** *se grmljavine* 'Il **a peur** tonnerre.GEN'
DepVAdv	dépendant d'un verbe sous forme d'un adverbe	*Filip lepo* ***peva*** 'Filip **chante** bien'
DepVInf	dépendant infinitif d'un verbe	***prestati*** *plakati* '**arrêter** pleurer'
DepVPart	dépendant d'un verbe introduit par un participe présent ou passé	***Vratio*** *se pevajući* 'Il est **rentré** en chantant'
DepVPrep	dépendant d'un verbe sous forme d'un groupe prépositionnel qui n'est pas un ObjIndir ou un prédicatif	***Učestvovao*** *je u organizaciji* 'Il a **participé** dans l'organisation'
Emph	dépendant de la racine de la proposition exprimant l'emphase, privé de fonction syntaxique	*To* ***dolazi*** *zima* lit. 'Ça **vient** hiver'
ExtraPred	dépendant de la racine de la proposition sous forme d'un adverbe extra-prédicatif	*On zapravo* ***kasni*** 'Il **est** en fait en retard'
Interrog	dépendant de la racine de la proposition sous forme d'un marqueur d'interrogation	***Dolazi*** *li Filip ?* 'Est-ce que Filip **vient** ?'

TABLE 5.3 – Jeu d'étiquettes syntaxiques ParCoLab

Etiquette	Définition	Exemple
Juxt	juxtaposition de deux éléments de haut niveau où aucune autre relation ne s'applique	*Posledice su se brzo* ***osetile*** *: vrtoglavica i mučnina* 'Les conséquences se sont vite fait **sentir** : vertige et nausée'
Neg	négation (verbale ou nominale)	*Ne* ***dolazi*** 'Il ne **vient** pas'
ObjDir	objet direct, à l'accusatif ou au génitif	*Filip* ***jede*** *jabuku* 'Filip **mange** pomme.ACC'
ObjIndirCas	objet indirect au datif	*Filip* ***daje*** *jabuku Ani* 'Filip **donne** pomme.ACC *Ana.DAT*
ObjIndirPrep	objet indirect réalisé comme *o* 'de' + N_locatif	*Filip* ***misli*** *o putovanju* 'Filip **pense** de voyage.LOC'
Polylex	relation réunissant les éléments d'une locution prépositionnelle ou adverbiale ou d'une conjonction complexe	*Dolazi* ***zato*** *što mora* 'Il vient **parce** qu'il est obligé'
PredCompletive	relation entre le subordonnant et le prédicat de la complétive	*Zna* ***da*** *dolazim* 'Il sait **que** je viens'
PredPercont	relation entre le prédicat de la principale et le prédicat de la percontative	***Pitao*** *je zašto dolaze* 'Il a **demandé** pourquoi ils venaient'
PredRap	relation entre le verbe introductif et le verbe principal du discours rapporté	*« Dolazim »,* ***kaže****.* « J'arrive », **dit**-il.
PredRel	relation entre l'antécédent d'une relative et son prédicat	***čovek*** *koji je došao* '**l'homme** qui est venu'
PredSub	relation entre le subordonnant et le prédicat de la subordonnée	*Dolazi* ***kad*** *završi* 'Il vient **quand** il finit'
PredicAdv	prédicatif adverbial : complément adverbial du verbe *biti* 'être'	*Filip* ***je*** *u Beogradu* 'Filip **est** à Belgrade'

TABLE 5.3 – Jeu d'étiquettes syntaxiques ParCoLab

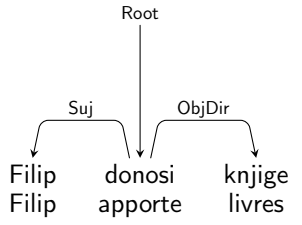
Etiquette	Définition	Exemple
PredicComplObj	prédicatif complémentaire lié à l'objet direct : complément nominal, adjectival ou prépositionnel d'un verbe obligatoirement attributif autre que le verbe *biti* 'être'	***Proglasili*** *su ga kraljem* 'Ils l'ont **proclamé** roi.INS'
PredicComplSuj	prédicatif complémentaire lié au sujet : complément nominal, adjectival ou prépositionnel d'un verbe obligatoirement attributif autre que le verbe *biti* 'être'	***Proglasio*** *se kraljem* 'Il s'est **proclamé** roi.INS'
PredicNom	prédicatif nominal : complément nominal du verbe *biti* 'être'	*Filip* ***je*** *profesor* 'Filip **est** professeur'
PredicOpt	prédicatif optionnel : complément nominal ou adjectival d'un verbe optionnellement attributif	*Filip se* ***vratio*** *umoran* 'Filip est **rentré** fatigué'
Ref	relie le verbe au pronom réflexif	***Osvežio*** *se* 'Il s'est **rafraîchi**'
Root	relie la racine externe et la tête de la phrase	***ROOT*** *Dolazi sutra* '**ROOT** Il vient demain'
Sub	relation entre le verbe principal et le subordonnant introduisant une proposition subordonnée	***Dolazi*** *kad završi* 'Il **vient** quand il finit'
Suj	sujet grammatical, exprimé par le nominatif	*Filip je* ***stigao*** 'Filip est **arrivé**'
SujLog	sujet logique, exprimé par le datif, génitif ou accusatif	*Filipu* ***je*** *dosadno* 'Filip s'**ennuie**' (lit. 'Filip.DAT est ennuyeux')
Ponct	relie la ponctuation au premier token précédent qui n'en est pas une	***Razočaran*** *,* *vratio se* ***kući*** *.* '**Déçu** , il est rentré à la **maison** .'

Comme on peut le remarquer, le traitement proposé ne se prononce pas sur le statut argumental ou non des dépendants verbaux. Cette décision est argumentée en détail dans la section 5.2.7. Dans la suite, nous nous intéresserons également aux relations syntaxiques qui diffèrent de celles communément admises dans la grammaire serbe, et nous aborderons aussi le traitement de quelques structures syntaxiques complexes qui ne sont pas centrales à la syntaxe théorique du serbe et qui sont parfois marginalisées dans les guides d'annotation des projets existants.

5.2.3 Statut du verbe auxiliaire

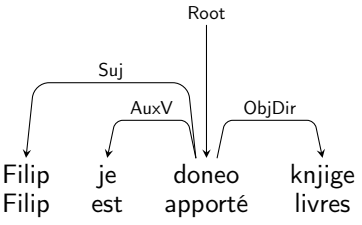
Dans notre schéma d'annotation, nous favorisons les têtes fonctionnelles par rapport aux têtes lexicales. Cependant, le traitement du verbe auxiliaire fait exception. En effet, nous considérons que c'est le verbe lexical qui est la racine d'une phrase, et par conséquent, dans une phrase à forme verbale complexe, le verbe auxiliaire est annoté comme un dépendant du verbe principal (cf. exemple 8). Milićević (2009) soutient que le rôle de la racine en serbe est dévolu aux verbes auxiliaires, et c'est également le cas dans de nombreuses études sur d'autres langues (cf. (Abeillé & Godard, 2002) pour le français, (Kupść & Tseng, 2005) pour le polonais, (Krapova, 1995) pour le bulgare). Néanmoins, nous considérons le verbe lexical comme le gouverneur, étant donné que cela permet une représentation plus immédiate de la structure argumentale du verbe, avec le sujet et tous les autres arguments qui dépendent directement du verbe lexical. Ce traitement garantit également que la racine de la phrase sera la même indépendamment de la forme verbale (cf. exemples 7 et 8) Le même choix a été fait, à titre d'exemple, dans les corpus FTBDep (cf. Candito et al., 2009, p. 9) et PDT (cf. Hajič et al., 1999, p. 19). Certains travaux en grammaire de dépendances – quoique minoritaires – soutiennent cette vision (cf. Hays, 1964 ; Matthews, 1981).

(7)



'Filip apporte les livres.'

(8)



'Filip a apporté les livres.'

3. Le paucal est une forme casuelle spécifique, imposée aux mots qui se déclinent par les numéraux *dva* 'deux', *tri* 'trois' et *četiri* 'quatre'. Il s'agit d'une trace de l'ancien dual.

4. Le traitement de la coordination sera présenté en détail dans la section 5.2.9.

5. Il s'agit typiquement d'intensifieurs.

6. Le traitement de l'ellipse a été repris de Prague Dependency Treebank (Hajič et al., 1999).

5.2.4 Traitement de la fonction du prédicatif

L'utilisation des propriétés distinctives des relations syntaxiques présentées ci-dessus nous a permis de confirmer rapidement la validité de la relation du prédicatif. Présente dans la syntaxe théorique serbe aussi bien que croate, elle n'est pourtant pas répercutée dans es jeux d'étiquettes existants pour le croate (cf. Agić et al., 2014 ; Merkler et al., 2013).

Selon (Stanojčić & Popović, 2012), on peut identifier 4 prédicatifs différents en serbe :

— *prédicatif nominal*[7], gouverné par le verbe *biti* 'être' et réalisé sous la forme d'un GN ou GA au nominatif, ou d'un GP à sens adjectival (cf. exemple 9a),

— *prédicatif adverbial*, gouverné lui aussi par le verbe *biti* 'être', mais réalisé sous la forme d'un GAdv ou d'un GP à sens adverbial (cf. exemple 9b),

— *prédicatif complémentaire*, gouverné par un verbe obligatoirement attributif, autre que *biti* 'être' et pouvant avoir la forme d'un GN ou d'un GA à l'instrumental, d'un GP introduit par *za* 'pour' complété par un accusatif, ou, pour certains verbes, d'un GN ou d'un GA au nominatif (cf. exemple 9c),

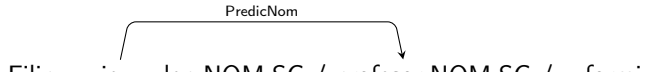
— *prédicatif optionnel*, gouverné par un verbe occasionnellement attributif, et pouvant prendre la forme d'un GA au nominatif ou à l'accusatif ou bien d'un GN au génitif (cf. exemple 9d).

Rappelons que dans la tradition française, ces relations sont réduites à une seule fonction, celle de l'attribut (du sujet et de l'objet). Cependant, si l'on examine ces structures en utilisant la liste des propriétés distinctives présentée dans la section 5.2.1, on constate que ces relations présentent des caractéristiques suffisamment spécifiques pour être distinguées les unes par rapport aux autres, mais aussi par rapport aux autres relations retenues. Il s'agit d'abord de leurs gouverneurs : pour les prédicatifs nominal et adverbial, c'est exclusivement le verbe *être*, alors que pour le prédicatif complémentaire il s'agit d'un groupe limité de verbes (dits *semi-copulatifs* dans (Stanojčić & Popović, 2012)). La seule exception est le prédicatif optionnel, pour lequel le gouverneur possible ne semble pas être restreint, mais cet élément est marqué par des traits morphosyntaxiques spécifiques. Ceci est par ailleurs le cas des trois autres aussi.

Par ailleurs, ces structures ont un ancrage fort dans la linguistique slave : les notions de *predicate nominal* 'prédicat nominal' et de *predicative nominal* 'prédicatif nominal', équivalent direct du prédicatif nominal en serbe, sont présentes dans les travaux fondateurs en grammaire formelle du tchèque (Sgall et al., 1969) et servent de base à l'étiquette

7. Il faut préciser que le terme *prédicat nominal* ne désigne pas le même phénomène en français et en serbe : alors que dans la tradition française il représente les noms qui expriment un processus verbal (typiquement des déverbaux comme *élection*, *bombardement* etc.), en serbe il s'agit d'un prédicat composé d'une copule et d'un élément nominal (nom, pronom, adjectif ou groupe prépositionnel au sens nominal). À titre d'exemple, dans la phrase *Filip je profesor* 'Filip est professeur', le groupe *je profesor* 'est professeur' est analysé comme un prédicat nominal, alors que le nom *profesor* 'professeur' a le rôle du prédicatif.

(9) a.

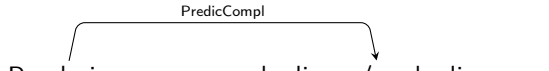


'Filip est beau/professeur/en forme.'

b.

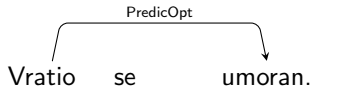
'Le film est à l'affiche.'

c.



'Il s'est proclamé roi.'

d.



'Il est rentré fatigué.'

Pnom (*predicative nominal*) dans le PDT (Hajič et al., 1999, pp. 28-37). La grammaire russe connaît également la notion de prédicat nominal, et le prédicatif introduit par le verbe *être* est typiquement appelé 'la partie copulative du prédicat nominal' (Cubberley, 2002, p. 190). Dans le treebank SynTagRus, cette relation (*Присвязочное синтаксическо отношение* 'relation syntaxique copulative') [8]) est appliquée aux constructions équivalentes du prédicatif nominal aussi bien qu'adverbial. Dans le même corpus, les exemples du prédicatif optionnel sont traités comme coprédicatifs (*копредикативное синтаксическо отношение* 'relation syntaxique coprédicative') [9]. La tradition grammaticale croate reconnaît également le prédicat nominal, en désignant le prédicatif nominal comme *predikatsko ime* 'nom prédicatif/prédicationnel' (Peti, 2005). Le prédicatif optionnel est à son tour traité comme *predikatni proširak* 'extension prédicative/du prédicat' (Peti, 1979 ; Karabalić, 2003 ; Barić et al., 1995 ; Katičić, 1986). Une terminologie comparable à celle de Stanojčić & Popović (2012) existe également dans la linguistique anglaise : le prédicatif nominal correspond aux termes de *predicative noun* et *predicative adjective*, alors que les prédicatifs complémentaire et optionnel sont traités comme *predicative complements* (Kim & Sells, 2008). Dans la section 5.2 de (Huddleston, 1984), on trouve une typologie

8. Point 1.7 dans la documentation en ligne du corpus (`http://ruscorpora.ru/instruction-syntax.html`). Dernier accès : 26/01/2016.
9. *Ibid*, points 2.5.8 et 2.5.9.

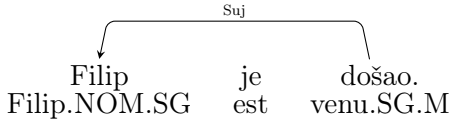
des prédicatifs bien détaillée, basée, entre autres critères, sur les propriétés aspectuelles du verbe gouverneur. Certaines de ces distinctions sont reprises dans le corpus anglais de Universal Dependencies.

Ce tour d'horizon rapide permet de voir que les 4 fonctions de Stanojčić & Popović (2012) sont reconnues à la fois dans la syntaxe théorique et dans la linguistique de corpus de différentes langues, malgré le fait que les corpus cités peuvent avoir des principes de construction très différents. Il n'y avait donc pas d'inconvénient *a priori* à ce que cette terminologie soit mise à l'épreuve dans le cadre d'une application en corpus sur le serbe. Ces fonctions ont donc été incluses dans le jeu sous les étiquettes suivantes : PredicNom pour le prédicatif nominal, PredicAdv pour le prédicatif adverbial, PredicCompl pour le prédicatif complémentaire et PredicOpt pour le prédicatif optionnel.

5.2.5 Sujet grammatical et sujet logique

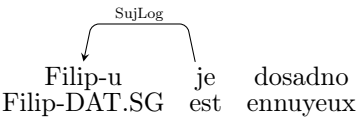
Un procédé comparable a été adopté pour examiner les fonctions du sujet grammatical et du sujet logique. En serbe, le sujet est typiquement exprimé au nominatif, il répond à la question *ko ?* 'qui-sujet-humain' ou *šta* 'quoi-sujet-non-humain' et régit l'accord du verbe (cf. exemple 10a). Cependant, un groupe de verbes exprimant un état physique ou mental ouvrent des places aux éléments qui répondent aux questions *kome ?* 'à-qui-humain', *koga ?* 'qui-objet-humain' ou 'de-qui-humain', et qui sont exprimés respectivement au datif, à l'accusatif ou au génitif (cf. exemples 10b et 10c). Ce constituant est désigné dans la littérature comme sujet logique (cf. Stanojčić & Popović, 2012, p. 263-264).

(10) a.



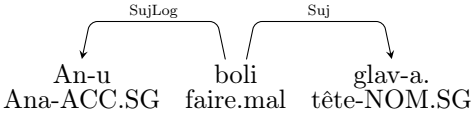
'Filip est venu.'

b.



'Filip s'ennuie.'

c.



'Ana a mal à la tête.'

Cette distinction n'est pas retenue dans PDT, HOBS ou SETimes. Elle se base pourtant sur des propriétés morphosyntaxiques nettes, accessibles aux parsers. Par ailleurs, certains

verbes, comme *boleti* ‘faire mal’, admettent les deux types de sujet (cf. exemple 10c). D’après le critère de répétabilité de la TST, s’il s’agissait de la même fonction, cette situation devrait être valide avec chaque gouverneur possible. Ceci n’est pas le cas : la présence d’un sujet au nominatif n’est pas possible dans les constructions du type montré dans l’exemple 10b. Cela confirme qu’il s’agit de deux fonctions syntaxiques différentes.

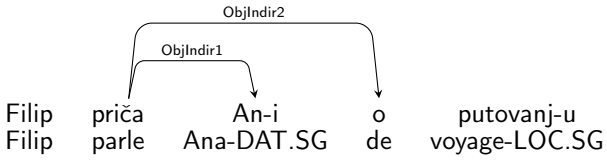
L’éventuelle difficulté pour le parsing se trouve plutôt dans la possibilité de confusion du sujet logique avec l’objet direct (typiquement à l’accusatif) et l’objet indirect (typiquement au datif). Mais malgré cette proximité formelle entre ces différents types de dépendants verbaux, une différence syntaxique existe : dans une phrase canonique, le sujet logique se positionne à gauche du verbe (cf. exemple 10b), alors que l’objet direct et l’objet indirect se trouvent à droite de leur gouverneur. Il n’est cependant pas clair si cette différence sera suffisante pour garantir l’apprentissage de cette fonction syntaxique par le parser. Nous avons donc décider de la maintenir à ce stade et d’évaluer son adaptation une fois le corpus annoté. Nous introduisons donc le label Suj pour le sujet grammatical et le label SujLog pour le sujet logique.

5.2.6 « Objet indirect »

La relation de l’objet indirect, telle que décrite dans les grammaires traditionnelles du serbe, a dû être redéfinie dans le jeu d’étiquettes. Si l’on se réfère à Stanojčić & Popović (2012, p. 247), les constituants cités dans cet ouvrage comme exemples de l’objet indirect peuvent avoir la forme d’un GN au datif, génitif ou instrumental sans préposition, mais aussi celle d’un GP, qui peut être introduit par un nombre élevé de prépositions différentes (les auteurs en notent 9), qui sont, elles, complétées par des GN à des cas différents (génitif, accusatif, instrumental, locatif). Il s’agit en effet de l’ensemble des compléments d’un verbe à nature nominale ou prépositionnelle hormis les objets directs et les prédicatifs. Cette fonction couvre donc un ensemble de constituants formellement très disparates, reliés entre eux par le seul fait qu’ils ont le statut d’argument par rapport au verbe gouverneur. Une telle diversité suggère qu’il pourrait s’agir de plusieurs fonctions différentes.

En effet, certains verbes n’admettent qu’un seul de ces compléments (cf. *najesti se* ‘manger à satiété’, *ličiti* ‘ressembler à’, *sumnjati* ‘douter de’, etc.), alors que d’autres peuvent en accueillir plusieurs (cf. l’exemple 11). D’après le critère de répétabilité de la TST, ceci signifie qu’il s’agit bien de fonctions distinctes.

(11)



‘Filip parle à Ana du voyage’

En effet, deux sous-ensembles de constituants semblent se dégager de cette agglomération de dépendants divers et variés : l'objet indirect prototypique, exprimant le bénéficiaire d'un processus verbal, et le complément des verbes de parole et de processus mentaux. Le premier est systématiquement exprimé sous forme de datif sans préposition, alors que le deuxième se réalise quasi exclusivement comme un GP introduit par la préposition *o* 'de' complétée par un nom au locatif (les deux sont instanciés dans l'exemple 11). Ce dernier se différencie de l'objet indirect prototypique aussi bien par la forme que prend le dépendant que par le fait qu'il ne peut pas être pronominalisé par un clitique (alors que l'objet indirect prototypique peut être remplacé par un clitique au datif). Tout comme le premier, le GP introduit par *o* 'de' n'est pas répétable avec le même verbe, mais les deux peuvent se combiner entre eux, certains verbes de parole permettant l'intervention d'un objet indirect typique aussi, comme nous avons pu l'observer dans l'exemple ci-dessus.

Les phénomènes correspondants dans d'autres langues ont droit à des traitements différents dans les corpus analysés : dans PDT, HOBS et SETimes, tous les compléments verbaux sont traités comme Obj. Dans SynTagRus, les compléments sont marqués par les fonctions kompl1, kompl2, ... kompl5, qui encodent la proximité du complément par rapport au verbe dans la structure argumentale de celui-ci. FTBDep met en œuvre 4 étiquettes : obj destiné typiquement à l'objet direct, a_obj pour les objets indirects introduits par *à* cliticisables par *lui* ou *leur*, de_obj pour les objets indirects introduits par *de* remplaçables par *en* ou *dont*, et p_obj pour les objets indirects introduits par d'autres prépositions. Il s'agit de compléments non effaçables ou non mobiles, correspondant typiquement au complément d'agent au passif et aux locatifs obligatoires. Les dépendants optionnels d'un verbe sont traités comme modifieurs (l'étiquette mod).

Étant donné un certain parallélisme entre le traitement proposé par FTBDep et les relations d'objet identifiées (objet direct bien circonscrit et deux types d'objet indirect), nous adoptons les étiquettes suivantes : en plus de l'étiquette ObjDir destinée à l'objet direct, l'objet indirect prototypique au datif est annoté comme ObjIndirCas, alors que le complément des verbes de parole et de processus mentaux introduits par *o* 'de' sont traités comme ObjIndirPrep. Il est bien possible que d'autres fonctions se cachent sous la relation générale d'objet indirect, mais les identifier nécessiterait une quantité importante de données linguistiques et une étude approfondie. Nous avons donc remis l'examen du reste de ces éléments pour plus tard en mettant en place un traitement de surface pour ces éléments (cf. section 5.2.7).

5.2.7 Dépendants du nom, verbe, adjectif et adverbe

Un autre point de notre schéma d'annotation qui diverge de la syntaxe théorique du serbe (et d'autres langues comme le français) est celui regroupant les étiquettes sous-

spécifiées pour les dépendants des verbes, noms, adjectifs et adverbes commençant par Dep. Au lieu d'entrer dans les distinctions fines entre les compléments et les ajouts, ces étiquettes sont basées sur le terme général de *dépendant*. Leur deuxième partie indique le type du gouverneur, qui peut être N, V, Adj ou Adv ; la troisième indique la forme morphosyntaxique du dépendant, qui peut être un nom fléchi (Cas), un groupe prépositionel (Prep), un adjectif (Adj) ou un adverbe (Adv).

Ces étiquettes ont été créées au moment où la liste des propriétés distinctives des relations syntaxiques présentée dans la section 5.2.1 s'est avérée insuffisante pour analyser l'ensemble des dépendants nominaux en serbe. Selon (Stanojčić & Popović, 2012), il existe trois types de dépendants nominaux : *kongruentni atribut* 'attribut accordé' (équivalent direct de la fonction d'épithète en français, ex. 12a), *padežni atribut* 'attribut casuel' (qui correspond *grosso modo* au complément du nom en français, ex. 12b), et *atributiv* 'attributif', une fonction syntaxique à part exprimée par un groupe nominal juxtaposé à un autre nom (ex. 12c).

L'attribut accordé se réalise sous la forme d'un adjectif qui s'accorde pleinement avec le nom (en genre, nombre et cas). La fonction de l'attribut casuel désigne un GN à un cas oblique ou un GP. Les deux réalisations sont désignées par le terme *casuel* car la tradition grammaticale serbe considère souvent que les GP sont une manifestation spécifique d'un cas. La préposition est vue comme une sorte d'élément introductif exigé par la forme fléchie du nom et les constructions impliquant une préposition et un nom dans un cas oblique sont souvent appelées des constructions prépositionelles-casuelles (*predloško-padežne konstrukcije*). L'attributif, quant à lui, désigne un dépendant nominal sous la forme d'un GN au même cas que son gouverneur.

(12) a.

Novi	student	nosi	**plavu**	torbu
nouveau.NOM.SG.M	étudiant.NOM.SG	porte	bleu.ACC.SG.F	sac.ACC.SG

'Le nouvel étudiant porte un sac bleu.'

b.

ljubav	**majke**	i	torta	**od**	**jabuka**
amour.NOM.SG	mère.GEN.SG	et	gâteau.NOM.SG	de	pomme.GEN.PL

'l'amour maternel et le gâteau aux pommes'

c.

Gospodin	Petrović	je	u	hotelu	**Slavija**
monsieur.NOM.SG	Petrovic.NOM.SG	est	dans	hôtel.LOC.SG	Slavija.NOM.SG

Monsieur Petrovic est à l'hôtel Slavija.

Si l'on essaye de contraster les fonctions présentées, on se rend compte qu'il est difficile d'opérer une distinction fiable. L'attribut accordé est exclusivement un GA, l'attribut casuel peut être soit un GN, soit un GP, alors que l'attributif est exclusivement un GN. La distinction entre l'attribut casuel et l'attributif se joue dans le fait que le premier ne

s'accorde en principe pas avec le gouverneur, contrairement à l'attributif. Cependant, il est tout à fait possible que l'attribut casuel se trouve au même cas que son gouverneur à cause des exigences du contexte plus large (cf. *od glave psa* lit. 'de tête.GEN chien.GEN', 'de la tête du chien'). Dans ce cas, la distinction se perd. Par ailleurs, si l'attributif exprime le nom de l'entité désignée par le gouverneur (cf. *hotel Slavija* 'hôtel Slavija' dans l'exemple 12c), il peut garder la forme du nominatif quel que soit le cas du gouverneur. On peut essayer d'avoir recours à l'ordre des constituants pour clarifier ces distinctions. Cependant, même si l'attribut accordé préfère la position à gauche du gouverneur, et l'attribut casuel celle à sa droite, les deux sont capables de se déplacer (ceci est discuté en profondeur dans le chapitre 10). Pour ce qui est de l'attributif, certains sous-types se positionnent à droite du gouverneur, et d'autres à sa gauche (cf. exemple 12c).

Pour éviter de regrouper des phénomènes syntaxiques disparates sous la même étiquettes, ce qui a été identifié comme défavorable pour le parsing par Mille et al. (2012), nous avons préféré entamer l'annotation manuelle du corpus avec des étiquettes génériques pour les dépendants du nom. Afin de faciliter l'analyse du comportement de différents types de dépendants nominaux une fois le corpus constitué, nous introduisons trois étiquettes exprimant 3 formes morphosyntaxiques que peut avoir un dépendant nominal : DepNAdj, DepNPrep et DepNCas. Une approche comparable est adoptée dans plusieurs corpus consultés : à l'exception de SynTagRus, qui met en place une classification très élaborée de différents types de dépendants nominaux, PDT, HOBS et SETimes couvrent ces fonctions avec une seule étiquette (Atr). FTBDep en distingue deux : l'épithète, qui s'applique aux adjectifs dépendants d'un nom, et dep, qui désigne tout dépendant nominal sous forme d'un GP.

À partir de cette analyse des dépendants nominaux, un traitement parallèle pour les dépendants des verbes, des adjectifs et des adverbes a été établi. Ces étiquettes sont sous-spécifiées : elles répercutent des informations déjà présentes au niveau de l'annotation morphosyntaxique, sans apporter davantage d'informations sur la nature du lien posé par la dépendance. On pourrait donc considérer qu'une seule étiquette Dep aurait suffi. Il nous a semblé néanmoins intéressant de garder cette distinction dans l'annotation manuelle afin d'observer de manière plus immédiate le comportement de ces différentes configurations. En effet, en serbe les groupes nominaux et prépositionnels peuvent être gouvernés aussi bien par un nom ou un verbe que par un adjectif ou un adverbe. Les cas de figure possibles avec les noms ont été montrés ci-dessus. En ce qui concerne les verbes, il peut s'agir de constructions comme *ići* ***kući*** 'aller **à la maison**' ou *boriti se* ***za napredak*** 'lutter **pour le progrès**'. L'adjectif en serbe peut être modifié par des éléments adverbiaux, mais par ailleurs, certains adjectifs admettent un dépendant nominal ou prépositionnel sous une forme spécifique : *sličan* 'semblable' peut être complété par un nom au datif (*sličan* ***ocu*** lit. 'semblable **père.DAT**', 'semblable à son père'), alors que *zaljubljen* 'amoureux' peut

avoir comme dépendant un groupe prépositionnel introduit par *u* 'dans' suivi d'un nom à l'accusatif (*zaljubljen* ***u Anu*** lit. 'amoureux **dans Ana.ACC**', 'amoureux d'Ana'). Les adverbes de quantité admettent également des compléments au génitif, comme dans *mnogo* ***ljudi*** lit. 'beaucoup **gens.GEN**', 'beaucoup de gens'. Ces éléments sont par ailleurs dotés de différents degrés de flexibilité quant à leur position par rapport à leur gouverneur.

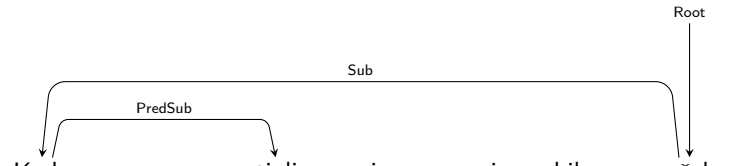
Ces faits semblaient propices à provoquer un nombre d'erreurs important dans l'identification du gouverneur des groupes nominaux et prépositionnels. En surdéfinissant ces relations, nous espérions attirer l'attention des annotateurs humains sur ces structures et pouvoir analyser plus facilement la distribution d'erreurs. Par ailleurs, cette méthode nous a permis de poser des contraintes précises dans l'interface d'annotation syntaxique, ce qui s'est montré particulièrement utile dans le cadre de l'annotation manuelle (voir la section 8.2.2). Nous avons également effectué une analyse approfondie du comportement des adjectifs dépendants d'un nom à l'aide de ces étiquettes. Elle est présentée dans le chapitre 10.

5.2.8 Traitement des subordonnées

Parmi les propositions subordonnées, on peut identifier deux types de comportements principaux. D'un côté, nous avons celles introduites par un subordonnant dont la seule fonction au plan syntaxique est d'assurer l'inclusion de la subordonnée dans la proposition principale. Il peut s'agir aussi bien des complétives, qui n'ont pas de rôle au plan sémantique non plus, que des circonstancielles, qui, quant à elles, apportent un contenu sémantique. D'un autre côté, on observe les propositions relatives et interrogatives indirectes, dont les subordonnants sont des mots en *qu-*. Bien que leur fonction sémantique puisse être analysée de différentes manières (cf. Le Goffic, 2007), ces formes ont typiquement un double rôle au niveau syntaxique : elles assurent la subordination, mais elles ont également une fonction syntaxique à l'intérieur de la subordonnée (cf. Chomsky, 1977). Elles semblent donc avoir une double dépendance dans la phrase, une au niveau de la principale, et une deuxième au niveau de la subordonnée (il s'agit souvent d'une dépendance du verbe, mais ce n'est pas le seul cas de figure possible, cf. l'exemple 14b). Cependant, d'après le principe du gouverneur unique préconisé par la TST et adopté par les systèmes de parsing (cf. section 2.2.3), un token peut avoir un seul gouverneur. Ces formes ne peuvent donc garder qu'une de ces dépendances dans le cadre d'une annotation en corpus. Nous avons fait le choix de favoriser ici le lien à l'intérieur de la subordonnée, dans le but de préserver une représentation fidèle de la structure argumentale des verbes dans la subordonnée. Nous mettons donc en place deux traitements principaux : le premier pour les propositions dont le subordonnant n'a qu'un rôle au niveau syntaxique, et le deuxième pour les relatives et les interrogatives indirectes.

Les propositions dont le subordonnant a une seule fonction sont traitées à l'aide de deux étiquettes : Sub relie le verbe de la principale et le subordonnant, et PredSub est utilisée pour établir le lien entre le subordonnant et le verbe de la subordonnée.

(13)



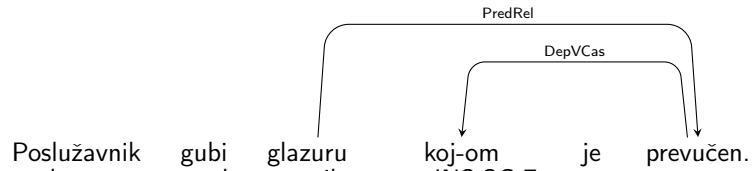
'Quand nous sommes arrivés, l'automne avait commencé.'

Il est à noter que les subordonnées en *da* 'que' complétant les verbes aspectuels ou modaux et les complétives introduites par le même subordonnant bénéficient d'un traitement légèrement différent. Les particularités sont disponibles dans le guide d'annotation (tome 2, annexe C).

Le deuxième type de traitement connaît deux variations : l'une pour les relatives et l'autre pour les interrogatives indirectes.

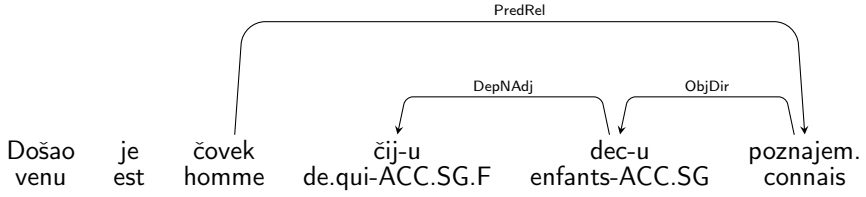
En ce qui concerne les relatives, le lien entre la proposition principale et la subordonnée s'établit en reliant l'antécédent de la relative à son prédicat par la fonction PredRel.

(14) a.



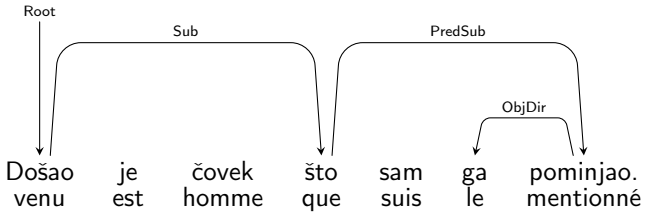
'Le plateau perd l'émail dont il est recouvert.'

b.



'L'homme dont je connais les enfants est venu.'

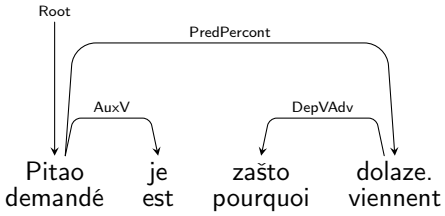
Il est à noter que le serbe admet également des relatives introduites par *što* 'que' invariable, dans lesquelles la fonction normalement dévolue au relatif est reprise par un pronom personnel. Le traitement de ces subordonnées est identique à celui des subordonnées à subordonnant simple (cf. exemple 15).



(15) 'L'homme que j'ai mentionné est venu.'

L'application du traitement prévu pour les relatives à ce cas de figure entraînerait la représentation du verbe de la relative avec deux objets directs. D'ailleurs, du point de vue sémantique, dans ces propositions, le relatif tend à perdre son sens relativisant pour ne garder qu'un rôle de subordonnant.

Les interrogatives indirectes sont traitées de la manière suivante : le verbe de la principale gouverne le verbe de la percontative *via* la relation PredPercont, alors que le subordonnant est relié au verbe de la subordonnée par l'étiquette qui exprime le mieux sa fonction par rapport au verbe.



(16) 'Il a demandé pourquoi ils venaient.'

Ce traitement s'inspire directement de celui mis en place dans le treebank FTBDep. Dans ce corpus, les propositions adverbiales sont traitées comme des modifieurs d'un verbe : le subordonnant porte l'étiquette mod, alors que la tête de la subordonnée en dépend *via* la fonction obj. En ce qui concerne les relatives, elles sont également considérées comme des modifieurs, mais leur rattachement est différent : c'est l'antécédent de la relative qui gouverne son prédicat, et le relatif dépend du prédicat de la subordonnée. Il est annoté en accord avec la fonction qu'il exerce par rapport au verbe de la relative, préservant ainsi la représentation de la structure argumentale du verbe. Un traitement comparable a été mis en place dans le schéma d'annotation utilisé pour l'annotation du corpus croate SETimes (Merkler et al., 2013).

D'autres corpus adoptent des approches différentes. PDT favorise la tête lexicale pour tout type de subordonnée. C'est donc le verbe de la subordonnée qui dépend directement du verbe de la principale *via* la relation appropriée (Adv dans le cas des circonstancielles,

Obj dans le cas des déclaratives et des percontatives, Atr dans le cas des relatives). Le subordonnant est rattaché au verbe de la subordonnée à travers la relation la plus appropriée (Adv dans le cas des circonstancielles et des percontatives, Obj ou Suj dans les relatives)(Hajič et al., 1999).

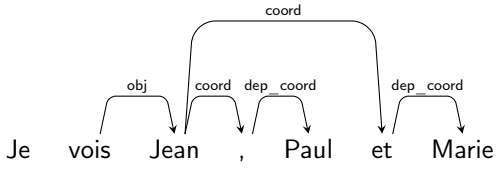
La deuxième version du treebank croate HOBS adopte la vision inverse : le verbe de la principale gouverne le subordonnant *via* la relation Sub et le subordonnant gouverne à son tour le verbe de la subordonnée à travers la relation Pred. Ce traitement concerne toutes les subordonnées, y compris les relatives. Par conséquent, la fonction syntaxique du relatif à l'intérieur de la relative est perdue.

5.2.9 Coordination

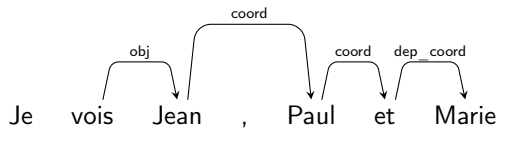
En ce qui concerne la coordination, deux approches principales sont mises en œuvre dans les ressources consultées. La première consiste à considérer la conjonction de coordination comme tête de la structure, et les conjoints comme ses dépendants (cf. PDT, SETimes.hr). La deuxième met en place un traitement qui considère que le premier conjoint gouverne la conjonction de coordination, qui à son tour gouverne le deuxième conjoint. Ce traitement est présent dans FTBDep.

Dans la version originale de FTBDep, c'est le premier conjoint qui est considéré comme la tête de la construction. Il est annoté en fonction de son rôle dans la phrase, et gouverne tous les coordonnants présents dans la structure, notamment la conjonction de coordination, mais aussi les virgules si la structure contient plus de 2 conjoints (étiquette coord). Les coordonnants, quant à eux, gouvernent chacun le conjoint qu'ils introduisent, annoté comme dep_coord (cf. exemple 17a).

(17) a.



b.

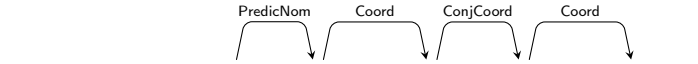


Or, Urieli (2014) démontre qu'une autre approche facilite le parsing de ces structures. Il propose la modification qui traite tous les conjoints entre le premier et la conjonction de coordination comme coord en établissant des dépendances en cascade (chaque conjoint dépend de celui qui lui précède). La conjonction porte elle aussi la même étiquette, et le dernier conjoint est annoté comme dep_coord. Les virgules ne sont plus annotées (cf.

exemple 17b). Ce traitement a apporté une réduction d'erreur dans le traitement des constructions coordonnées de 26,72 % (en f-mesure). Nous retenons donc cette deuxième approche, avec une légère modification en ce qui concerne les étiquettes : tous les conjoints (sauf le premier) sont annotés comme Coord, alors que la conjonction porte l'étiquette ConjCoord. Cependant, elle est bien gouvernée par le conjoint immédiatement précédent (cf. exemple 18).

Nous considérons également comme coordination les enchaînements de plusieurs éléments sans conjonction de coordination. Ce phénomène est décrit comme juxtaposition dans la littérature linguistique, mais il est analysé comme coordination dans plusieurs corpus consultés (FTBDep, PDT, SETimes). Une illustration du traitement proposé peut être trouvée dans l'exemple 19.

(18)



'La gouttière était vieille, cabossée et déformée.'

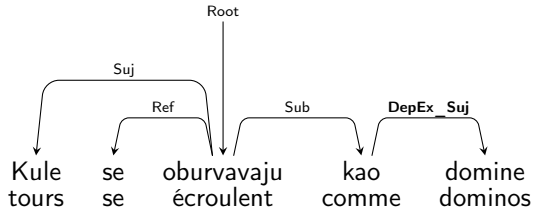
(19)

'La gouttière était vieille, cabossée, déformée.'

5.2.10 Ellipse

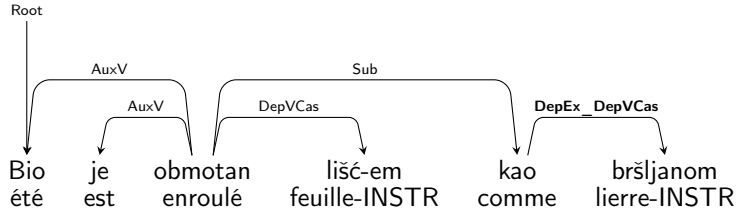
L'ellipse est un phénomène syntaxique complexe, dont le traitement ne semble pas encore circonscrit, que ce soit en linguistique théorique ou en linguistique de corpus (cf. Bos & Spenader, 2011). En cherchant à mettre en place un traitement aussi simple et aussi systématique que possible, capable cependant de préserver des informations linguistiques, nous avons décidé de retenir l'approche de PDT. Dans ce corpus, une forme dépendante d'un élément phrastique élidé est rattachée au gouverneur de l'élément élidé. Cette dépendance porte l'étiquette de la relation que la forme concernée a par rapport à son gouverneur, augmentée d'un suffixe qui indique qu'il s'agit d'une ellipse. Nous adaptons légèrement ce traitement en utilisant le préfixé de DepEx_ ('dépendance externe') pour effectuer cette modification. De nombreux cas de figure sont possibles ; quelques-uns parmi les plus fréquents sont illustrés dans la suite.

Dans l'exemple 20, la proposition élidée pourrait être restituée comme *Kule se oburvavaju kao što se oburvavaju domine* 'Les tours s'écroulent comme s'écroulent des dominos'. Cette manipulation permet d'identifier que la forme *domine* se trouve dans la position du sujet dans la proposition élidée. Par conséquent, on lui accorde l'étiquette DepEx_Suj, et



(20) 'Les tours s'écroulent comme des dominos'

on la fait gouverner par le subordonnant : c'est lui qui serait le gouverneur du verbe de la subordonnée, si celui-ci était présent dans la phrase.



(21) 'Il était enroulé avec des feuilles comme avec du lierre'

Dans l'exemple 21, la proposition complète serait *kao da je obmotan bršljanom* 'comme s'il était enroulé avec du lierre', mais le verbe, qui gouvernerait normalement la forme à l'instrumental, est élidé. Par conséquent, c'est la conjonction (qui gouvernerait le verbe de la subordonnée s'il était présent) qui est le gouverneur de la forme à l'instrumental, et comme cette forme aurait la fonction DepVCas par rapport au verbe élide, c'est l'étiquette DepEx_DepVCas qui est utilisée.

Le point faible principal de cette approche réside dans le fait que n'importe quelle étiquette peut être modifiée de cette manière, menant ainsi à une augmentation du jeu d'étiquettes difficile à contenir. Il a cependant l'avantage de préserver l'information sur la véritable place de la structure élidée dans l'arbre syntaxique. Ceci n'est pas le cas avec le traitement proposé par SETimes.hr : dans ce corpus, les éléments gouvernés par des formes omises de la phrase sont annotés avec l'étiquette Elp, qui est rattachée au token racine de la phrase, quelle que soit leur position réelle.

FTBDep, quant à lui, ne propose pas d'étiquette spécialisée qui servirait à rattacher à la phrase tout type d'élément dépendant d'un token élidé, mais se concentre plutôt sur le traitement des constructions spécifiques dans lesquelles l'ellipse peut apparaître, telles les constructions de comparaison, les subordonnées comparatives ou les relatives averbales en *dont*. Cependant, ce type de traitement exige une étude détaillée des constructions impliquées que nous n'avons pas pu effectuer dans le cadre de cette thèse.

Tout comme le guide d'annotation morphosyntaxique, la version initiale du guide d'annotation syntaxique a été mise à l'épreuve des données dans le cadre de l'annotation ma-

nuelle, pour être ensuite soumise à une évaluation de l'accord inter-annotateurs. Cette démarche est décrite en détail dans la section 7.5.2. La totalité du guide d'annotation syntaxique est disponible dans l'annexe C (tome 2).

5.3 Principes de lemmatisation adoptés

Parmi ces trois tâches d'annotation, la lemmatisation est sans doute la moins complexe. À la différence de l'étiquetage morphosyntaxique et de l'annotation syntaxique, qui font appel à des jeux d'étiquettes et des règles d'annotation complexes, la lemmatisation exige simplement la capacité de retrouver les formes canoniques de la langue traitée. Par conséquent, il suffisait de quelques règles relatives à des phénomènes précis en serbe pour définir le cadre de cette tâche. En cas de doute, les annotateurs étaient invités à se reporter au dictionnaire électronique du serbe de Simić (2005).

5.3.1 Traitement des verbes *jesam* et *biti*

Il existe en serbe deux verbes 'être' : *jesam* 'je suis' (et sa forme négative *nisam* 'je ne suis pas') et *biti* 'être'. À la différence de l'italien et de l'espagnol, par exemple, il n'y a pas de véritable distinction sémantique entre les deux. Le verbe *jesam* est un verbe défectif : il existe seulement au présent et ne dispose pas d'un infinitif, et c'est lui qui exprime le présent indicatif. Le verbe *biti* dispose d'un paradigme complet, y compris au présent. À partir de ce critère morphosyntaxique, on considère traditionnellement qu'il s'agit de deux lemmes différents. Nous préservons cette distinction dans la lemmatisation.

5.3.2 Traitement des adjectifs : forme courte ou forme longue

En serbe, c'est le nominatif singulier du masculin de l'indéfini qui est considéré comme la forme de base d'un adjectif. Cette forme se termine quasi-systématiquement par une consonne (cf. *velik* 'grand', *nov* 'nouveau', etc.). Cependant, certains adjectifs – et notamment les adjectifs relationnels – n'ont pas de formes de l'indéfini et sont donc cités au nominatif singulier du masculin du défini, qui se termine typiquement par la voyelle *-i* (cf. *seoski* 'villageois', *alfabetski* 'alphabétique'). Cependant, la frontière entre ces deux ensembles d'adjectifs n'est pas clairement déterminée : pour certains adjectifs massifs, les deux lemmes sont possibles (cf. *mermeran* et *mermerni* 'en marbre', *papiran* ou *papirni* 'en papier', *kristalan* ou *kristalni* 'en cristal'). Il en est de même pour certains adjectifs qualificatifs, rarement utilisés à l'indéfini pour des raisons sémantiques (cf. *davan/davni* 'ancien', *divalj/divlji* 'sauvage'), etc.

La consigne dans ce cas de figure était de consulter le dictionnaire mentionné ci-dessus (Simić, 2005), et si l'indéfini est reconnu comme possible, l'utiliser pour la lemmatisation

de la forme fléchie en question. Une liste de lemmes adjectivaux validés a été établie au fur et à mesure ; elle est disponible dans le guide d'annotation pour la lemmatisation (cf. tome 2, annexe B).

5.3.3 Traitement des verbes : question de lemmes doublons

En ce qui concerne les verbes, pour certains paradigmes il existe deux formes de l'infinitif reconnues : la forme *brojim* 'je compte' peut se lemmatiser comme *brojati* ou *brojiti* 'compter', *podignem* 'je soulève' peut correspondre à *podići* ou *podignuti* 'soulever', et *stojim* 'je me tiens debout' peut avoir comme infinitif *stojati* ou *stajati* 'se tenir debout', etc.

Dans le cas des verbes où on a le choix entre l'infinitif en *-ći* et celui en *-ti*, nous choisissons systématiquement celui en *-ći*. Pour les autres cas, les infinitifs retenus sont notés dans une liste ouverte qui était continuellement mise à jour lors de lemmatisation manuelle (cf. tome 2, annexe B).

5.3.4 Autres cas de figure

Les prépositions *ka* 'vers' et *sa* 'avec' disposent également des formes allomorphes respectives *k* et *s*. Pour la lemmatisation, nous utilisons systématiquement des formes longues.

Le pronom réflexif clitique *se* 'se' dispose d'une forme pleine *sebe*. Comme cette forme pleine est beaucoup moins fréquente en corpus, nous utilisons la forme brève pour la lemmatisation.

Étant donné la relative simplicité de la tâche, nous n'avons pas mesuré l'accord inter-annotateurs sur ce segment du travail. Une description du processus de la lemmatisation manuelle est disponible dans la section 8.5.

5.4 Bilan intermédiaire

Dans ce chapitre, nous avons montré comment nous avons traduit les principes d'annotation présentés dans le chapitre 4 en jeux d'étiquettes et guides d'annotation. Au niveau morphosyntaxique, la restriction du jeu aux traits impliqués dans le fonctionnement syntaxique du serbe nous a permis d'obtenir un jeu d'étiquettes détaillées plus petit que celui de Krstev et al. (2004b) (1042 étiquettes *vs* 1243). Au niveau syntaxique, notre objectif était inverse : nous avons cherché à établir un jeu plus riche que celui actuellement utilisé sur le croate (cf. Merkler et al., 2013), qui ne compte que 15 étiquettes. En nous servant de critères de distinction des relations syntaxiques de la TST, nous avons constitué un jeu

d'étiquettes qui preserve de nombreuses distinctions de la grammaire serbe (cf. les prédicatifs, les sujets), mais en abandonne d'autres (cf. les dépendants du nom, la distinction argument *vs* ajout). Les écarts par rapport à la tradition grammaticale serbe étaient motivés par un manque de critères de surface qui permettent d'identifier la fonction syntaxique en question. En évitant ce type de distinctions, nous cherchons à garantir l'adaptation des étiquettes pour une utilisation en parsing.

Cependant, avant d'être jugés adaptés à une utilisation en corpus, ces schémas d'annotation et les guides associés devaient être évalués de manière explicite. Ils ont donc d'abord été mis à l'épreuve des données par nous-mêmes, pour être ensuite soumis à une évaluation de l'accord inter-annotateurs. Nous revenons sur cette question dans le chapitre dédié à l'initialisation de notre méthode (cf. chapitre 7, section 7.5).

Chapitre 6

Création de ressources lexicales

Comme indiqué dans le chapitre 4, la création d'un lexique morphosyntaxique relève du stade de préparation de la campagne. Comme nous avons eu recours à une extraction de données à partir du Wiktionary, ce travail a fait appel aux compétences en TAL, bien que l'expertise linguistique ait été nécessaire pour évaluer la qualité du lexique obtenu.

En effet, le seul lexique serbe librement disponible au début de cette thèse était trop petit pour avoir un effet satisfaisant dans le cadre du parsing (20 000 entrées seulement) (cf. Krstev et al., 2004b). Pour assurer une couverture plus solide, nous avons constitué un lexique morphosyntaxique à partir du Wiktionnaire pour le serbo-croate (Miletic, 2017). Ce travail a été effectué en 2015 ; en 2016, un lexique serbe (ainsi qu'un lexique croate) construit manuellement a été diffusé par Ljubešić et al. (2016). À notre connaissance, il s'agit des lexiques serbes les plus importants diffusés librement (respectivement 3,1 millions et 5,3 millions d'entrées).

Pour assurer une ressource optimale pour nos expériences en TAL, nous avons effectué la fusion de ces deux lexiques et nous nous en sommes servie dans nos expériences en parsing. Notre lexique basé sur le Wiktionnaire est présenté dans la section 6.1, le lexique srLex de Ljubešić et al. (2016) est décrit dans la section 6.2, et les détails sur le processus de fusion et la ressource résultante sont donnés dans la section 6.3.

6.1 Lexique *wikimorph-sr*

Notre premier projet de constitution du lexique a été directement inspiré par les travaux de (Sajous et al., 2013 ; Sagot, 2014 ; Sennrich & Kunz, 2014). Ces chercheurs ont exploité la ressource libre de Wiktionnaire pour constituer des ressources électroniques dotées d'informations morphosyntaxiques. L'avantage principal de cette approche réside dans son coût réduit par rapport à une création manuelle. Par ailleurs, l'utilisation de Wiktionary permet d'obtenir un lexique qui n'est pas soumis à des restrictions quant à

la redistribution et aux applications possibles. Nous avons donc décidé d'explorer cette option.

Wiktionary est un projet collaboratif de création de dictionnaire démarré en 2002. Aujourd'hui, il comprend plus de 150 langues. Les entrées dictionnairiques peuvent contenir des définitions, mais aussi des informations phonétiques, flexionnelles et sémantiques, ainsi que des traductions dans d'autres langues. Cela fait du Wiktionary une ressource précieuse pour le TAL. Plusieurs travaux ont montré que ce type de ressources peut obtenir des résultats comparables ou même meilleurs que les ressources créées par des experts (voir, par exemple les travaux de Strube & Ponzetto (2006), Zesch & Gurevych (2007), Zesch & Gurevych (2010) ou Gabrilovich & Markovitch (2007)). Wiktionary a été exploité dans diverses applications du TAL : il a été utilisé pour le calcul de la proximité sémantique (Zesch et al., 2008), la création de réseaux de synonymie (Navarro et al., 2009), la constitution et l'enrichissement d'ontologies (Meyer & Gurevych, 2012 ; Pérez et al., 2011), ainsi que pour la dérivation de lexiques morphosyntaxiques (Sajous et al., 2013 ; Sagot, 2014 ; Sennrich & Kunz, 2014).

Les données du Wiktionary sont diffusées sous forme de *dump* périodique de fichiers XML. Malheureusement, seule la macrostructure des pages qui est balisée en XML, alors que le contenu de la page se présente en wikicode, un format textuel très flexible et peu documenté. L'élaboration d'un parser pour wikicode doit donc être faite à partir d'une observation méticuleuse des pages. De surcroît, comme il a déjà été noté dans (Navarro et al., 2009 ; Sajous et al., 2013), cette structure varie de manière importante entre les éditions de différentes langues, ce qui signifie qu'un parser du wikicode développé pour une langue ne peut pas être simplement transposé pour une autre. C'est pour cette raison que nous avons dû construire un nouvel extracteur. Ce processus est décrit dans la section 6.1.1.

6.1.1 Extraction de données

Notre lexique a été extrait de l'édition du Wiktionary pour le serbo-croate. En effet, deux éditions proposent du contenu serbe : l'édition serbo-croate (`sh.wiktionary.org`) et la version serbe (`sr.wiktionary.org`). Ce fait semble être dû à des facteurs extralinguistiques plutôt que linguistiques : un parcours manuel des deux éditions ne nous a pas permis de relever des différences importantes quant à la qualité du contenu. Nous avons donc sélectionné la version serbo-croate car elle contient un nombre d'entrées largement supérieur à l'édition serbe : 850 000 *vs* 45 000. L'extraction s'est focalisée sur les informations morphosyntaxiques, notamment le cas, le nombre et le genre. Elle a été basée sur le *dump* du 2 octobre 2015.

Étant donné que plusieurs standards d'encodage peuvent coexister dans la même édition, voire dans la même page du Wiktionary, un extracteur doit faire preuve d'une grande

robustesse pour maximiser la quantité de données extraites. Par exemple, dans le *dump* que nous avons utilisé, il existe deux types de pages principaux : la page dont l'entrée est un lemme et qui contient le paradigme complet du lemme en question (cf. figure 6.1), et la page dont l'entrée est une forme fléchie et qui en liste toutes les interprétations morphosyntaxiques possibles (cf. figure 6.2).

Dans le premier format, les traits morphosyntaxiques de chaque forme fléchie peuvent être indiqués par des codes (typiquement dans le cas des verbes) ou bien ils sont déduits à partir de la position de la forme dans la table du paradigme (typiquement dans le cas des noms, cf. figure 6.1). Cela est possible grâce au fait que les tables du Wiktionary suivent de manière générale la présentation des paradigmes traditionnellement acceptée pour le serbe. Par exemple, dans la figure 6.1, la première colonne représente le singulier, la deuxième le pluriel, et les cas sont présentés dans l'ordre suivant : nominatif, génitif, datif, accusatif, vocatif, instrumental et locatif. Cependant, nous avons également rencontré des pages où les formes de l'instrumental et du locatif étaient permutées. Pour éviter une extraction erronée, notre extracteur effectue une vérification basée sur la terminaison de la forme pour vérifier si le cas inféré par l'extracteur à partir de la position de la forme dans le tableau correspond au suffixe exhibé par la forme en question et corrige si nécessaire l'information du cas qui est inscrite dans le lexique.

```
====Deklinacija====
{{sh-imenica-deklinacija2
|jezik|jezici
|jezika|jezika
|jeziku|jezicima
|jezik|jezike
|jeziče|jezici
|jeziku|jezicima
|jezikom|jezicima
}}
```

FIGURE 6.1 – Modèle de page du Wiktionary basé sur le lemme : article du mot *jezik* 'langue'

Dans les articles dont l'entrée est une forme fléchie, la forme fléchie traitée est donnée en tête de l'article entre guillemets, suivie par une série de descriptions textuelles de ses traits morphosyntaxiques introduites par des dièses. Il s'agit typiquement des groupes nominaux qui doivent être décomposés et analysés pour en extraire l'information sur les valeurs des différents traits morphosyntaxiques. Par exemple, l'article présenté dans la figure 6.2 traite la forme *guvernerskim*, une forme fléchie ambiguë de l'adjectif *guvernerski* 'relatif au gouverneur'. La première ligne commençant par "#" dans la figure 6.2, à savoir *instrumental množine ženskog roda pozitiva određenog vida pridjeva*, signifie littéralement 'instrumental du pluriel du genre féminin du positif de l'aspect déterminé de l'adjectif'.

L'ordre dans lequel les informations sont présentées n'est pas fixe, et par ailleurs, certaines données peuvent être absentes de la description.

```
===Flektirani oblici===
"'guvernerskim'"

# instrumental množine ženskog roda pozitiva određenog vida pridjeva
[[guvernerski#Srpskohrvatski|guvernerski]]
# lokativ množine ženskog roda pozitiva određenog vida pridjeva
[[guvernerski#Srpskohrvatski|guvernerski]]
# dativ množine muškog roda pozitiva određenog vida pridjeva
[[guvernerski#Srpskohrvatski|guvernerski]]
# instrumental množine muškog roda pozitiva određenog vida pridjeva
[[guvernerski#Srpskohrvatski|guvernerski]]
```

FIGURE 6.2 – Modèle de page du Wiktionary basé sur la forme fléchie : article du mot *guvernerski* 'relatif au gouverneur'

De nombreuses autres variations de plus bas niveau ont également été détectées, comme différents encodages des formes verbales, et divers codes utilisés pour indiquer certains traits morphosyntaxiques. Afin de maximiser la quantité d'informations extraites, nous avons consacré une attention particulière au traitement de chacun des cas de figure relevés.

Nous avons également repéré et comblé quelques lacunes quasi-systématiques dans le traitement de certains types des lemmes. Par exemple, la majorité des entrées adverbiales contenait des adverbes au comparatif ou au superlatif, mais il n'y avait pas d'entrée correspondante au positif. Comme les articles en question contenaient néanmoins le lemme, et que la forme du positif d'un adverbe est identique au lemme, ces entrées-là ont été générées automatiquement. Il en est de même pour les formes du futur simple qui étaient omises des articles de certains verbes : ce temps ayant un schéma de flexion très régulier, il suffisait de disposer de l'infinitif d'un verbe pour pouvoir créer les formes manquantes.

Nous avons également constaté que Wiktionary était particulièrement pauvre en formes pour les classes fermées. Le résultat de l'extraction initiale a donc été enrichi en utilisant plusieurs autres sources. 107 prépositions ont été importées des listes constituées manuellement lors des travaux théoriques sur les relations spatiales de Stosic (2001), et un ensemble de 76 prépositions, 43 conjonctions, 33 interjections et 868 adverbes ont été extraits du corpus étiqueté en parties du discours de (Miletic, 2013). Toutes ces formes ont été rajoutées au résultat de l'extraction automatique.

6.1.2 Wikimorph-sr : taille et caractéristiques principales

La ressource résultante, nommée Wikimorph-sr, contient 1 226 638 formes fléchies provenant de 117 445 lemmes différents, réparties en 3 066 214 triplets uniques <*forme fléchie, lemme, étiquette morphosyntaxique détaillée*>. Ce lexique est donc nettement mieux doté

que le lexique existant du projet MultextEast Krstev et al. (2004b) (20 000 entrées).

Wikimorph-sr est stocké dans un format textuel illustré dans la figure 6.3. La première colonne contient la forme fléchie, la deuxième contient le lemme, et la troisième l'étiquette morphosyntaxique correspondante. Les étiquettes morphosyntaxiques utilisées sont celles de notre jeu d'étiquettes morphosyntaxiques, à une différence près : si l'information sur un trait morphosyntaxique spécifique n'était pas présente dans Wiktionary, on indique "0" à sa place dans l'étiquette. Cela permet de différencier les cas où l'information n'a pas pu être extraite de ceux où le trait ne s'applique pas à la forme en question, marqués à leur tour par "-" dans l'étiquette morphosyntaxique.

```
trag     trag     N_com_acc_sg_m
trag     trag     N_com_nom_sg_m_trag
traga    tragati  V_main_pres_3_sg_-_-
traga    trag     N_com_gen_sg_m
tragah   tragati  V_main_impf_1_sg_-_-
```

FIGURE 6.3 – Format textuel du lexique *Wikimorph-sr*

Il convient ici d'apporter plusieurs précisions par rapport au contenu de Wikimorph-sr. Tout d'abord, le lexique contient un certain degré de surgénération. Ceci est dû au fait qu'une partie des pages ont été créées à travers des schémas de flexion appliqués quasi systématiquement à tous les lemmes d'une catégorie. Dans le domaine adjectival, cela a généré des formes du comparatif et du superlatif même pour les lemmes qui se prêtent difficilement à la comparaison pour des raisons sémantiques (cf. *abecedniji* 'plus alphabétique', *bakteriološkiji* 'plus bactériologique'). Ce phénomène touche aussi certains verbes : on retrouve des formes de l'imparfait pour des verbes perfectifs, et des formes de l'aoriste pour les imperfectifs. Or, l'aspect lexical de ces verbes bloque normalement la génération de formes verbales exprimant l'aspect opposé (cf. section 1.1.5).

Bien que cette caractéristique puisse s'avérer problématique pour certaines applications en TAL, elle ne l'est pas (ou l'est dans une moindre mesure) pour l'usage du lexique que nous envisageons : on ne cherchera dans le lexique que les formes retrouvées dans les textes traités, par définition existantes. Il reste cependant le fait que la partie utile du lexique est plus petite que sa taille apparente.

La ressource contient également une part importante de noms propres : 355 178, soit plus de 10 % d'entrées. Quoiqu'elle semble diminuer encore la portion du lexique utile au traitement des textes généraux, cette caractéristique peut se montrer précieuse pour la reconnaissance des entités nommées dans d'autres types d'applications.

6.1.3 Wikimorph-sr : couverture et ambiguïté

Afin de mieux évaluer la valeur réelle du lexique, nous avons effectué des tests de couverture et analysé l'ambiguïté dans le lexique. Les deux expériences sont présentées dans la suite.

Nous avons mesuré la couverture du lexique sur un échantillon du volet serbe de ParCoLab de 150 000 tokens équivalent à 28 980 formes fléchies uniques. La couverture a été calculée pour toutes les formes fléchies, puis pour celles ayant au moins 2, 5 et 10 occurrences dans l'échantillon (cf. tableau 6.1).

Seuil de fréquence	No de formes fléchies uniques	Trouvées dans Wikimorph-sr	Couverture
1	28 980	20 808	71,8 %
2	10 630	8 136	76,5 %
5	2 946	2 328	79,0 %
10	1 241	990	79,8 %

TABLE 6.1 – Test de couverture de Wikimorph-sr

À titre d'illustration, nous comparons ces résultats avec ceux du GLÀFF, un lexique français dérivé du Wiktionary (Hathout et al., 2014). Ce lexique contient plus de 2 millions d'entrées. Sur un corpus journalistique de 200 millions de tokens (300 000 formes fléchies uniques), sa couverture des formes au seuil de fréquence de 10 est de 86,23 %. Nous constatons donc que la couverture de notre lexique est un solide point de départ, mais elle mérite d'être améliorée.

Un autre facteur peut affecter l'utilité d'un lexique lors de l'annotation automatique : le degré d'ambiguïté qu'il encode. Comme il a été expliqué précédemment, si un étiqueteur ou un parser fait appel à une entrée ambiguë dans un lexique, l'outil doit être capable de déterminer, en fonction du contexte, quelle est la description morphosyntaxique à retenir. La difficulté de cette tâche augmente avec le degré d'ambiguïté présent dans le lexique utilisé. Ce phénomène est souvent majeur dans les langues à morphologie flexionnelle riche, et le serbe ne fait pas exception : un syncrétisme important existe, notamment dans les paradigmes nominaux et adjectivaux (cf. section 1.1.1). Pour tenter une quantification plus précise, nous avons calculé différents indicateurs d'ambiguïté dans le lexique. Les résultats principaux sont donnés dans le tableau 6.2.

Nous remarquons que quasiment 60 % de formes fléchies représentées dans le lexique sont ambiguës et exhibent au moins deux interprétations morphosyntaxiques possibles. Par ailleurs, une portion non négligeable des formes fléchies (3 %) exhibe un degré d'ambiguïté élevé, avec 10 interprétations possibles ou plus. Il est également important de noter que parmi les formes ambiguës, 95 % relèvent des cas de syncrétisme à l'intérieur d'un même

No de formes fléchies	1 226 638
No d'entrées	3 066 214
No d'intérprétations par forme fléchie	2,50
No de formes ambiguës (%)	706 817 (57,6 %)
No de formes avec ≥10 entrées (%)	36 922 (3,0 %)

TABLE 6.2 – Analyse de l'ambiguïté dans Wikimoprh-sr

paradigme.

Malgré ces deux aspects perfectibles (la couverture et l'ambiguïté), la création de Wikimorph-sr représente un enrichissement important vis-à-vis des ressources lexicales existantes pour le serbe, aussi bien du point de vue de la taille que des conditions d'utilisation. Comparé au lexique MultextEast, contenant environ 20 000 entrées, notre lexique présente une nomenclature beaucoup plus importante. Par ailleurs, le lexique MultextEast est non seulement limité aux applications non commerciales, mais aussi soumis à la clause de non-redistribution. Il ne peut donc pas servir de base pour la dérivation de nouvelles ressources qui seraient librement diffusées. En revanche, Wikimorph-sr est adapté à tous types d'applications, y compris les applications commerciales [1].

6.2 Lexique srLex

Le lexique srLex de Ljubešić et al. (2016) contient 5,3 millions d'entrées. Il a été construit à partir des lexiques croate, serbe et bosniaque du logiciel de traduction automatique à base de règles Apertium (Forcada et al., 2011), mais il a connu des extensions importantes dans le cadre d'une campagne de création manuelle d'entrées. Le lexique est librement diffusé [2], avec une licence pour des applications non commerciales, mais qui autorise la redistribution.

6.2.1 Campagne de constitution de srLex

La ressource initiale sur laquelle srLex est basé contenait les paradigmes de 10 183 lexèmes, ainsi que les schémas flexionnels de 413 paradigmes différents. Ljubešić et al. (2016) ont mis en place une campagne d'augmentation du lexique effectuée par une équipe de 6 linguistes. La liste de lemmes à rajouter a été construite en prenant la fréquence comme critère : le lexique initial a été projeté sur le corpus srWaC (Ljubešić & Klubička,

1. Wikimorph-sr est librement disponible aux adresses suivantes : `http://redac.univ-tlse2.fr/lexiques/wikimorph-sr_fr.html` et `http://parcolab.univ-tlse2.fr/en/about/resources/`.

2. `http://nlp.ffzg.hr/resources/lexicons/srlex/`. Dernier accès : le 23 octobre 2017.

2014), et les formes fléchies non couvertes par le lexique ont été triées en fonction de la fréquence.

Tout d'abord, les données croates ont été traitées. Pour traiter les formes inconnues, les linguistes se servaient d'une interface graphique liée à un prédicteur qui leur proposait, pour chaque forme fléchie, des candidats pour le lemme et le schéma de flexion. Les linguistes pouvaient accepter un des candidats, ou bien signaler la forme comme appartenant à un schéma de flexion inexistant dans le système. À la fin de chaque cycle, les linguistes traitaient les formes signalées et créaient les schémas de flexion manquants, ce qui permettait de couvrir plus de formes fléchies dans le cycle suivant. Le processus a été effectué en 6 cycles. Le traitement des données serbes a été plus rapide : 2 cycles ont été suffisants, grâce au fait qu'il existe un recouvrement lexical important entre les deux langues.

Le lexique serbe obtenu contient 5 327 361 forme fléchie provenant de 105 358 lemmes différents. Les deux lexiques (le croate et le serbe) sont disponibles aux deux formats : celui du projet MultextEast et celui du projet Universal Dependencies. Ils sont disponibles à l'adresse `http://nlp.ffzg.hr/resources/lexicons/srlex/` [3].

Le lexique a été évalué par ses auteurs dans le cadre de l'annotation morphosyntaxique détaillée et de l'étiquetage en parties du discours. En fonction de l'outil et du corpus d'évaluation, srLex apporte une amélioration de l'exactitude de 1,38 % à 5,65 % en étiquetage détaillé, et de 0,71 % à 3,23 % points en étiquetage en parties du discours (Ljubešić et al., 2016).

6.3 Lexique combiné ParCoLex

La licence permissive de srLex nous a permis de fusionner les deux lexiques afin de maximiser leur utilité. Pour évaluer la pertinence de cette idée, nous avons d'abord comparé la couverture mutuelle des lexiques, ainsi que leur couverture sur un échantillon de ParCoLab. L'échantillon utilisé contient 16 389 tokens (sans signes de ponctuation), correspondant à 6 301 formes fléchies uniques. Pour chacun des deux lexiques, nous avons calculé le pourcentage des formes fléchies uniques ainsi que le pourcentage des tokens de l'échantillon (des occurrences des formes fléchies uniques) qu'il couvrait. Les résultats sont présentés dans le tableau 6.3.

SrLex étant plus large, les chiffres obtenus étaient prévisibles : Wikimorph-sr couvre seulement 20 % des entrées de srLex, alors que srLex contient plus de 41 % des entrées de Wikimorph-sr. Il existe également une différence importante en faveur de srLex en ce qui concerne la couverture de l'échantillon, que ce soit au niveau des formes fléchies uniques (92,8 % *vs* 63,3 %) ou des occurrences (93,8 % *vs* 73,2 %). Ceci s'explique sans doute par la méthode adoptée dans la création de srLex : les lemmes ajoutés durant l'étape

3. Dernier accès : le 23 octobre 2017

d'enrichissement manuel ont été choisis à partir des taux de fréquence calculés à partir d'un corpus web de taille importante.

Il est néanmoins intéressant de noter qu'une part importante d'entrées de Wikimorph-sr ne figurent pas dans srLex. Il nous a donc semblé pertinent de fusionner ces deux ressources et de vérifier si cet ajout augmentait encore l'utilité du lexique. La dernière ligne du tableau 6.3 présente le résultat de cette manipulation. Le lexique fusionné, nommé ParCoLex, contient au total 7 180 665 entrées uniques <*forme fléchie, lemme, étiquette morphosyntaxique détaillée*>, qui représentent 1 956 094 formes fléchies uniques provenant de 157 886 lemmes. La fusion a effectivement permis d'augmenter encore l'excellente couverture de srLex : nous observons un gain de 2,4 % sur les formes fléchies uniques, et de 4 % sur le nombre total d'occurrences des formes fléchies dans l'échantillon.

Lexique	Entrées	Lemmes	Couverture de l'autre lexique	Couverture échantillon	
				formes fléchies	occurrences
Wikimorph-sr	3 066 214	117 445	20,8 %	63,3 %	73,2 %
srLex	5 327 361	105 358	41,1 %	92,8 %	93,8 %
ParCoLex	7 180 665	157 886	NA	95,2 %	97,8 %

TABLE 6.3 – Tests de couverture avec les 3 lexiques

Par conséquent, nous avons retenu ParCoLex comme outil de travail pour la suite. Il a été intégré dans la méthode globale de *bootstrapping* récursif et a également été exploité plus tard, dans les expériences de parsing. Plus de détails sur ce point seront donnés dans le chapitre 9. Le lexique est également disponible à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Lexicons/ParCoLex`.

Chapitre 7

Mise en œuvre de la méthode adoptée

Après avoir défini les guides d'annotation (cf. chapitre 5) et créé le lexique morphosyntaxique (cf. chapitre 6), il nous reste encore un aspect de la préparation matérielle de la campagne d'annotation à traiter : l'initialisation de la méthode. Plus particulièrement, il nous faut identifier le corpus à annoter, optimiser les guides d'annotation, et mettre en œuvre les outils chargés de la préannotation automatique. Notre choix de corpus est présenté dans la section 7.1. Vu la complémentarité des besoins en ce qui concerne l'optimisation des guides et l'initialisation des outils, nous combinons ces deux points : les guides sont utilisés pour créer des ressources d'entraînement initiales pour les outils. L'entraînement initial de l'étiqueteur morphosyntaxique est décrit dans la section 7.2, celui du lemmatiseur dans la section 7.3, et celui du parser dans la section 7.4. Enfin, nous effectuons également une évaluation formelle des guides d'annotation à travers une évaluation de l'accord-inter-annotateurs (cf. section 7.5).

L'application des guides d'annotation et la constitution des ressources d'entraînement initiales sont faites par les annotateurs expérimentés, alors que l'initialisation des outils et la mise en place des évaluations de l'accord inter-annotateurs sont effectués par le taliste.

7.1 Corpus sélectionné pour l'annotation

Au moment du choix du contenu pour le treebank à constituer, nous avions à notre disposition un échantillon de texte dont le traitement avait déjà été entamé dans le cadre de l'un de nos travaux antérieurs (Miletic, 2013). Il s'agit du corpus ParCoTrain, présenté dans la section 1.2.1. Pour rappel, le corpus contient 150 000 tokens, il est doté d'une annotation en parties du discours et partiellement lemmatisé. Comme ces couches d'annotation existantes pouvaient se montrer utiles dans le présent travail, nous avons décidé de

baser notre corpus annoté syntaxiquement sur le même échantillon de texte. Nous avons pourtant limité la taille du treebank visée à 100 000 tokens : premièrement, cette taille est comparable à celle du corpus croate utilisé par Agić et al. (2013b) et Agić & Ljubešić (2015), ce qui facilite la comparaison des résultats obtenus sur ces corpus ; deuxièmement, nous avons considéré que ce chiffre était plus réaliste étant donné nos contraintes de temps. Par conséquent, nous avons repris du corpus ParCoTrain le contenu de deux ouvrages : *Bašta, pepeo* de Danilo Kiš (échantillon *basta*) et *Testament* de Vidosav Stevanović (échantillon *testament*) [1]. La structure du corpus sélectionné et l'état de l'annotation au démarrage de la constitution du treebank sont présentés dans le tableau 7.1.

Sous-échantillon	Tokens	Annotation POS	Lemmatisation
basta	55 783	Oui	Non
testament	45 642	Oui	Oui
Total	101 425		

TABLE 7.1 – Structure et état de l'annotation de l'échantillon sélectionné

Le genre textuel du corpus choisi appelle un commentaire. En effet, il s'agit d'un corpus littéraire. Or, une pratique plus courante est d'utiliser des textes journalistiques pour constituer un treebank (cf. Marcus et al., 1993 ; Abeillé et al., 2003 ; Agić & Ljubešić, 2014) : les textes journalistiques sont plus souvent libres de droits, ce qui facilite la diffusion de la ressource créée, et ils sont considérés comme moins complexes du point de vue syntaxique que les textes littéraires, ce qui simplifie l'apprentissage des parsers.

Le premier point n'était pas problématique dans notre cas : nous avions déjà l'accord des ayant droits pour la diffusion du corpus à des fins non commerciales [2]. En revanche, la question de la difficulté intrinsèque du texte était pertinente. Il était cependant difficile d'évaluer l'ampleur de l'effet que ce paramètre pouvait avoir sur l'apprentissage du parsing. Le fait d'exploiter un corpus déjà mis en place avait également un avantage pratique : cela nous évitait de passer par l'étape importante de récolte et de prétraitement des textes.

Le corpus retenu a été divisé en cinq échantillons d'environ 20 000 tokens chacun. Ce sont donc ces échantillons qui ont été traités tour à tour en suivant la méthode proposée dans la section 4.2. Pour faciliter leur identification dans la suite de ce document, nous nommons le corpus retenu ParCoTrain-Synt, et indiquons les différents échantillons comme 1_20, 2_20, 3_20, et ainsi de suite.

1. Kiš, Danilo. *Bašta, pepeo*, 2010. Podgorica : Narodna knjiga.
Stevanović, Vidosav. *Testament*, 1986. Beograd : SKZ.
2. Sous la licence CC BY-NC-SA 3.0 (`https://creativecommons.org/licenses/by-nc-sa/3.0/`).

7.2 Étiquetage morphosyntaxique

Comme il a été indiqué dans la section 3.2.7, nous avons retenu l'étiqueteur HunPos (Halácsy et al., 2007) pour l'étiquetage morphosyntaxique de notre corpus. HunPos est un outil basé sur le modèle HMM (*Hidden Markov models* (cf. section 3.2.1). Il s'agit en effet d'une nouvelle implémentation de l'étiqueteur TnT de Brants (2000b) qui apporte quelques améliorations par rapport à l'outil original.

Tout d'abord, HunPos rend possible l'exploitation d'un contexte d'apprentissage plus large. Dans le calcul de probabilités, au niveau des étiquettes TnT exploite seulement l'étiquette du token courant, tandis que HunPos utilise également l'étiquette précédente. Cet élargissement du contexte au niveau des étiquettes a apporté jusqu'à 10 % de réduction d'erreur (Halácsy et al., 2007, p. 210). En réalité, la taille du contexte exploité n'est pas limitée par HunPos, mais les auteurs indiquent que les fenêtres au delà de 2 tokens n'ont pas donné d'améliorations supplémentaires.

HunPos tente également d'améliorer le traitement des formes inconnues, le point faible principal des modèles HMM. Pour pallier ce problème, TnT dispose d'un module d'analyse de suffixes qui, durant l'apprentissage, créé une base de suffixes de tokens peu fréquents dans le corpus d'entraînement et mémorise la distribution des tags pour chaque suffixe. Ces informations sont ensuite utilisées dans l'étiquetage pour traiter les formes inconnues en fonction de leur terminaison. Cependant, ce module peut proposer un nombre d'étiquettes très élevé, dont certaines ne sont pas plausibles du tout. Pour limiter cet effet, HunPos permet l'utilisation d'un lexique, dont le rôle est de proposer les labels possibles pour la forme inconnue, alors que l'analyseur de suffixes leur attribue des poids et choisit le label final.

Dans la suite, nous décrivons la démarche que nous avons suivie pour préparer HunPos à l'utilisation dans le cadre de notre méthode de *bootstrapping* itératif.

7.2.1 Utilisation d'un modèle HunPos entraîné sur le croate

D'après le schéma de notre méthode globale (cf. chapitre 4), le premier entraînement des outils automatiques devrait être exécuté sur un échantillon annoté manuellement dans sa totalité : le corpus que nous avons retenu dispose d'une annotation morphosyntaxique, mais elle ne contient que l'indication des parties du discours (cf. section 7.1), alors que nous visons ici une annotation en traits morphosyntaxiques fins. Dans son état actuel, le corpus n'est donc pas adapté à un premier entraînement de HunPos. Une solution possible serait d'enrichir le corpus actuel en traits morphosyntaxiques de manière manuelle. Cependant, les détails des résultats de HunPos rapportés dans (Agić et al., 2013a) suggèrent qu'il pourrait être possible d'utiliser le modèle existant entraîné sur le croate pour faire une

préannotation automatique sur notre échantillon initial en serbe.

Comme mentionné ci-dessus, le modèle de HunPos testé dans ce travail a été entraîné exclusivement sur des données croates. Néanmoins, il a pu annoter des textes serbes avec une exactitude très proche de celle obtenue sur le croate : 87 % pour le croate *vs* 85 % pour le serbe. Les évaluations ont été effectuées aussi bien sur des textes journalistiques (du même domaine que le corpus d'apprentissage) que sur des textes issus de Wikipédia, pour les deux langues. Les taux d'exactitude rapportés dans (Agić et al., 2013a) sont repris dans le tableau 7.2.

croate		serbe	
presse	Wikipedia	presse	Wikipédia
87,72 %	81,52 %	85,56 %	82,79 %

TABLE 7.2 – Exactitude de HunPos dans (Agić et al., 2013a)

Comme il a déjà été observé sur la syntaxe (cf. section 3.4.3), le modèle croate semble plus affecté par le changement de domaine (journalistique *vs* encyclopédique) que par le changement de langue (croate *vs* serbe). Ce fait nous permettait d'envisager une utilisation du modèle croate sur nos données en serbe. Cependant, deux questions importantes se posaient. Tout d'abord, nous avons remarqué que le modèle est déstabilisé par le changement de domaine. Lors de son application sur ParCoTrain-Synt, l'écart entre le genre textuel du corpus d'apprentissage et du corpus à annoter risquait d'être encore plus marqué (journalistique *vs* littéraire). Il était difficile d'estimer à quel point les performances du modèle seraient affectées par cette transposition. La deuxième question concernait les différences entre le jeu d'étiquettes et le schéma d'annotation sur lesquels le modèle a été entraîné et les nôtres. Le jeu d'étiquettes intégré au modèle était le jeu croate du projet MultextEast (Erjavec, 2012) [3], fondé sur les mêmes principes de base que le jeu d'étiquettes serbe du même projet. Comme nous l'avons vu dans la section 5.1, notre jeu d'étiquettes, et notamment le schéma d'annotation associé, diffèrent en plusieurs points de celui de MultextEast. Qui plus est, les divergences entre les deux schémas d'annotation ne peuvent pas être éliminées par une correction automatique, le traitement correct de certains cas de figure étant fortement dépendant du contexte. La correction manuelle de la sortie du modèle croate devrait donc inclure non seulement la correction des erreurs d'annotation proprement dites, mais aussi les interventions nécessaires pour faire converger le schéma d'annotation du modèle vers celui de ParCoTrain-Synt. Il était tout à fait justifié de se demander si cela n'éliminerait pas l'avantage présupposé d'une préannotation automatique. Pour mieux estimer l'ampleur de cet effet, une première évaluation de

3. Le jeu d'étiquettes croate est présenté en détail à l'adresse suivante : `http://nl.ijs.si/ME/V4/msd/html/msd-hr.html`

la sortie de HunPOS sur notre corpus a été effectuée.

7.2.2 Évaluation du modèle croate sur un échantillon de ParCoTrain-Synt

Cette évaluation se fonde sur une correction manuelle en deux temps : d'abord, la sortie de l'outil a été corrigée en accord avec le schéma d'annotation de MultextEast, et dans un deuxième temps, la correction a été effectuée selon le schéma d'annotation de ParCoTrain-Synt. La première étape nous a permis d'évaluer la part d'erreurs d'étiquetage proprement dites dans la sortie de l'outil, alors que la deuxième nous a servi à évaluer l'effort global nécessaire pour adapter la sortie du modèle croate aux exigences du schéma d'annotation de ParCoTrain-Synt. Cette expérience a été effectuée sur un échantillon de 2122 tokens.

Le taux d'erreur et la précision ont été calculés à trois niveaux : au niveau des étiquettes des parties du discours, des étiquettes détaillées globales, mais aussi des traits morphosyntaxiques individuels. Pour rappel, les étiquettes du projet MultextEast sont des étiquettes positionnelles, encodant la partie du discours, mais aussi de nombreuses autres propriétés morphosyntaxiques, comme le genre, le nombre, le cas, la forme verbale, etc. Bien qu'une étiquette soit considérée comme incorrecte dès qu'elle contient un trait erroné, dans la perspective d'une correction manuelle, une étiquette contenant plusieurs attributs incorrects est plus lourde à corriger que celle qui n'en comporte qu'un seul : l'annotateur humain doit vérifier et, si nécessaire, corriger chacun de ces traits. Évaluer le taux d'erreur à ce niveau permet donc d'estimer le nombre d'interventions nécessaires de la part de l'annotateur humain. Les résultats sont présentés dans le tableau 7.3. Nous utilisons le taux d'erreur comme métrique : il s'agit du pourcentage d'unités qui ont été mal annotées.

Niveau d'évaluation	Schéma d'annotation	
	MultextEast	ParCoLab
Parties du discours	8,2 %	12,06 %
Étiquette détaillée	22,05 %	26,20 %
Traits individuels	11,89 %	16,01 %

TABLE 7.3 – Évaluation initiale du modèle croate appliqué au contenu de ParCoTrain-Synt

Quand on observe les résultats selon le schéma MultextEast, on constate une baisse d'exactitude d'environ 4 % au niveau des parties du discours et d'environ 8 % pour les étiquettes détaillées par rapport aux résultats de (Agić et al., 2013a). Cela semble confirmer notre hypothèse que le changement de genre entraîne une détérioration de performances. Quand la sortie du modèle est confrontée au schéma d'annotation de ParCoTrain-Synt, le taux d'erreur sur les étiquettes complètes monte de 22,05 % à 26,20 %. Au niveau

des traits individuels, ces 26 % d'étiquettes incorrectes représentent 16,01 % de traits qui nécessitent une intervention de l'annotateur humain.

Afin d'estimer l'effet exact de ce taux d'erreur sur le processus de correction, nous avons évalué la vitesse de l'annotation manuelle à partir de cette préannotation et l'avons comparée à la vitesse d'annotation manuelle intégrale (à partir du texte nu). Sans préannotation automatique, un annotateur expérimenté traite en moyenne 500 tokens/h, alors qu'il atteint la vitesse moyenne de 620 tokens/h en corrigeant la sortie du modèle croate. On réalise donc un gain d'environ 24 % de tokens par heure. Nous avons donc exploité cette préannotation pour valider le premier échantillon de 20 000 tokens de ParCoTrain-Synt.

L'accélération observée est cependant moins importante qu'on le souhaiterait. Les annotateurs ont indiqué que les interventions les plus chronophages concernaient les corrections dues aux différences entre les schémas d'annotation. Nous avons donc utilisé les 20 000 tokens validés pour ré-entraîner HunPos et obtenir ainsi un modèle intégrant le schéma d'annotation de notre corpus. Notre inquiétude principale était qu'un échantillon de cette taille ne serait pas suffisant pour entraîner un modèle aussi performant que le modèle croate existant (qui a été développé sur un corpus de 87 000 tokens). Nous avons donc évalué le modèle ré-entraîné aussi bien du point de vue de ses performances (évaluation quantitative), que de ses effets sur les temps de correction.

7.2.3 Ré-entraînement de HunPos sur le premier échantillon de ParCoTrain-Synt

Nous avons effectué une validation croisée à 10 itérations en utilisant comme corpus d'évaluation les 20 000 tokens issus de l'annotation avec HunPos après correction manuelle. L'apprentissage a été effectué dans les mêmes conditions que pour le modèle croate : aucune ressource externe n'a été utilisée. La variation de l'exactitude est assez importante entre différentes itérations, entre 70 % et 83 % (cf. tableau 7.4). Ceci était attendu étant donné la taille très limitée du corpus d'entraînement. Néanmoins, malgré la différence de taille entre les corpus d'entraînement pour le modèle croate (87 000 tokens) et le modèle ré-entraîné (20 000 tokens), les performances de base des deux modèles sur ParCoTrain-Synt sont très proches (respectivement 77,95 % et 78,82 % d'exactitude).

Exact.	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
	76,23	82,97	**83,52**	79,83	80,68	*70,91*	83,28	78,47	74,52	77,73
Moyenne	**78,82**									
Taille moyenne du corpus d'entraîn. :					18 370 tokens					
Taille moyenne du corpus d'éval. :					2 040 tokens					

TABLE 7.4 – Évaluation de HunPos sur un échantillon de 20 K tokens

Nous avons également observé une augmentation de la vitesse de correction manuelle

beaucoup plus importante par rapport à l'annotation manuelle intégrale. Nous avons utilisé le modèle ré-entraîné de HunPos pour annoter un nouvel échantillon de 20 000 tokens, non compris dans le corpus d'entraînement. Cette fois-ci, l'annotateur expert traite en moyenne 800 tokens/h, et l'annotateur novice atteint la vitesse de 325 tokens/h. Il s'agit donc d'une augmentation du nombre de tokens traités de respectivement 60 % et plus de 300 % par rapport à l'annotation manuelle intégrale (cf. tableau 7.5).

Scénario	Annotateur	Vitesse d'annotation
Annot. manuelle	Expert	500 tok/h
	Novice	80 tok/h
Préannot. modèle croate	Expert	620 tok/h (+24 %)
	Novice	non mesuré (-)
Préannot. modèle ré-entraîné	Expert	800 tok/h (+60 %)
	Novice	325 tok/h (+300 %)

TABLE 7.5 – Vitesse d'annotation manuelle en fonction du modèle de préannotation

Ayant jugé ces résultats concluants, nous avons retenu le modèle ré-entraîné comme outil de travail et l'avons utilisé pour effectuer la préannotation du deuxième échantillon de 20K tokens. La suite du travail sur l'annotation morphosyntaxique de notre corpus est décrite dans la section 8.4.

7.3 Lemmatisation

Comme indiqué dans la section 3.3.4, pour la lemmatisation, notre choix s'est arrêté sur l'outil CST (Jongejan & Dalianis, 2009) (cf. section 3.3.2). Depuis sa création, le lemmatiseur CST a été entraîné sur un ensemble de 24 langues, dont le serbe [4]. C'est également le cas du modèle croate développé par Agić et al. (2013a) [5]. Afin de vérifier l'adaptation de ces modèles à nos données, nous avons procédé à une évaluation initiale.

7.3.1 Entraînement initial de CST

Une partie du contenu sélectionné pour ParCoTrain-Synt avait déjà été lemmatisée dans le cadre du travail de (Miletic, 2013). Un échantillon de 10 000 tokens a été extrait de cet ensemble de données (dorénavant, *enctest-eval*) pour servir de corpus d'évaluation initiale. Nous avons testé les deux modèles disponibles. Les résultats sont indiqués dans le tableau 7.6.

4. La liste des langues, ainsi que tous les modèles, sont disponibles à la page de l'outil : `http://cst.dk/download/uk/index.html#lemmatiser`. Dernier accès : le 20 octobre 2017.

5. `http://nlp.ffzg.hr/resources/models/tagging/`. Dernier accès : le 20 octobre 2017

Modèle	Ressource d'entraînement	Sur *enctest-eval*
Fourni par CST	lexique MULTEXT-East (20K entrées)	76 %
Fourni par Agić et al. (2013a)	corpus SETimes.hr (87K tokens)	50,8 %
Modèle ré-entraîné	*enctest-train-unique*	86,2 %

TABLE 7.6 – Résultats de différents modèles de CST sur ParCoLab

Les résultats des deux modèles sur cet échantillon n'étaient pas satisfaisants : le modèle fourni avec CST avait atteint 76 % d'exactitude, alors que celui de Agić et al. (2013a) n'avait pas pu dépasser 50,8 %. Cette dégradation des performances est sans doute due au fait que les modèles ont été évalués sur un échantillon dépourvu d'annotation morphosyntaxique. En revanche, les deux modèles ont été entraînés sur des ressources étiquetées [6]. Bien qu'un modèle de CST entraîné avec des informations morphosyntaxiques puisse être appliqué à des données qui n'en sont pas dotées, les auteurs indiquent que les performances du modèle en seraient affectées.

Comme ces résultats n'étaient pas prometteurs en vue d'une accélération de la lemmatisation manuelle, nous avons effectué un entraînement initial de CST sur la partie de notre échantillon déjà lemmatisée. Autrement dit, nous avons converti 85 000 tokens lemmatisés en une ressource contenant des triplets uniques *forme fléchie – POS – lemme*. Le lexique résultant (dorénavant *enctest-train-unique*) contient 20 964 entrées correspondant à 10 197 lemmes différents.

Le modèle entraîné sur ce lexique minimal a atteint une exactitude de 86,2 % sur *enctest-eval* (cf. tableau 7.6), dépassant de manière nette les deux modèles préexistants. L'utilité de la préannotation a également été évaluée en mesurant la vitesse d'annotation sur un échantillon de 2000 tokens. En faisant la lemmatisation manuelle du texte nu, un annotateur humain expérimenté traitait en moyenne 825 tokens/h, alors qu'en corrigeant la sortie du modèle ré-entraîné, sa vitesse moyenne atteignait 1400 tokens/h. Par conséquent, nous avons retenu ce modèle et l'avons utilisé pour compléter la lemmatisation du premier échantillon de ParCoTrain-Synt.

D'après la méthode globale stipulée, l'outil aurait dû être ré-entraîné sur la totalité du texte lemmatisé désormais disponible (100 000 tokens). Cependant, nous nous sommes procuré le lexique morphosyntaxique ParCoLex, qui contient plus de 7 millions d'entrées (cf. chapitre 6). Il nous a semblé judicieux de dériver un nouveau modèle de lemmatisation pour CST en utilisant ce lexique. Sa couverture importante devait garantir une meilleure généralisation des patrons de flexion et, par conséquent, le développement d'un modèle

6. Le modèle croate a été entraîné à partir du corpus SETimes, annoté avec le jeu d'étiquettes croate du projet MultextEast. Le modèle fourni par CST a été fait à partir du lexique serbe du projet MultextEast (environ 20 000 entrées) exploitant le jeu d'étiquettes serbe du même projet (cf. le fichier *readme* disponible sur la page du modèle).

plus robuste. Cela permettrait de se passer des ré-entraînements répétés.

7.3.2 Ré-entraînement de CST sur le lexique combiné ParCoLex

Afin de faciliter l'entraînement du lemmatiseur CST, nous avons utilisé le lexique ParCoLex (cf. section 6.3). Certaines adaptations ont été effectuées pour optimiser son exploitation par CST. Notamment, CST génère un fichier de règles pour chaque étiquette rencontrée dans le corpus d'apprentissage. Pour éviter un éclatement du modèle, les étiquettes détaillées du lexique ont été remplacées par la seule indication de la partie du discours, à l'exception des noms : pour cette catégorie, l'information du genre a été également retenue, étant donné qu'elle peut permettre de désambiguïser entre différents schémas de flexion. Par exemple, la forme fléchie d'un nom féminin qui se termine par *-nci* aura très probablement un lemme en *-nka* (cf. *senci* (dat.sg.) > *senka* (nom.sg.) 'ombre'), alors que la forme fléchie d'un nom masculin exhibant la même terminaison peut avoir un lemme en *-nac* (cf. *lonci* (nom.pl.) > *lonac* (nom.sg.) 'casserole') ou en *-nak* (cf. *proplanci* (nom.pl.) > *proplanak* (nom.sg.) 'clairière'). Les étiquettes morphosyntaxiques retenues sont listées dans le tableau 7.7. Cette simplification des étiquettes a réduit le nombre d'entrées uniques de ParCoLex à 2 millions.

Étiquette	Catégorie
A	adjectif
Abr	abréviation
Adv	adverbe
C	conjonction
I	interjection
N	nom sans genre [a]
N_f	nom féminin
N_m	nom masculin
N_n	nom neutre
Num	numéral
P	pronom
Part	particule
Prep	préposition
V	verbe

TABLE 7.7 – Étiquettes morphosyntaxiques retenues pour l'entraînement de CST sur ParCoLex

a. D'après notre schéma d'annotation, certains noms propres (les noms de famille des femmes) ne portent pas la marque du genre. Voir le guide d'annotation morphosyntaxique (cf. tome 2, annexe A).

La documentation de CST indique que l'apprentissage de l'outil peut être facilité si les formes d'un lemme sont regroupés ensemble. Autrement dit, il faut créer des *clusters*

d'entrées en fonction du lemme, en séparant les entrées des lemmes différents par une ligne vide. Un extrait du contenu de ParCoLex modifié de sorte à correspondre à cette condition est donné dans la figure 7.1.

Gabonaca	Gabonac	N_m
Gabonac	Gabonac	N_m
Gabonca	Gabonac	N_m
Gabonce	Gabonac	N_m
Gabonče	Gabonac	N_m
Gaboncem	Gabonac	N_m
Gabonci	Gabonac	N_m
Gaboncima	Gabonac	N_m
Gaboncu	Gabonac	N_m
gabonska	gabonski	A
gabonske	gabonski	A
gabonski	gabonski	A
gabonskih	gabonski	A
gabonskima	gabonski	A
gabonskim	gabonski	A
gabonsko	gabonski	A
gabonskoga	gabonski	A
gabonskog	gabonski	A
gabonskoj	gabonski	A
gabonskome	gabonski	A
gabonskom	gabonski	A
gabonskomu	gabonski	A
gabonsku	gabonski	A

FIGURE 7.1 – Extrait du contenu de ParCoLex adapté à l'entraînement de CST

Une fois le ré-entraînement terminé, le nouveau modèle a été testé sur le premier échantillon de 20 000 tokens doté d'une annotation morphosyntaxique détaillée. Cet échantillon disposait déjà d'une lemmatisation manuelle, ce qui a permis d'exécuter une évaluation automatique. Le nouveau modèle a atteint une exactitude globale de 96,5 %, s'approchant ainsi des résultats obtenus par Agić et al. (2013a).

Nous avons également inspecté les performances du modèle sur les mots hors vocabulaire. Des 20 917 tokens de l'échantillon de test, 17 % (3 618) étaient inconnus de l'outil. Il s'agissait majoritairement des tokens de ponctuation : vu que l'entraînement de CST se base sur un lexique et non pas sur un véritable corpus, l'outil n'avait pas été exposé aux symboles de ponctuation. Seules 314 formes étaient de véritables mots inconnus du modèle. De cet ensemble, seules 60 formes étaient mal annotées. Ceci correspond à une part de 1,5 % de formes inconnues et à un taux d'exactitude de 80,9 % dans leur traitement (cf. tableau 7.8).

Étant donné le nombre très réduit d'interventions nécessaires, la vitesse d'annota-

	No de tokens	Exactitude	Taux d'erreur
Total	20 917	96,5 %	3,5 %
Inconnus	314 (1,5 %)	80,9 %	19,1 %

TABLE 7.8 – Résultats du modèle de CST ré-entraîné sur ParCoLex

tion manuelle a connu une augmentation très importante : elle a atteint 3400 tok/h (cf. section 8.5). Vu ces résultats plus que satisfaisants, il n'a plus été nécessaire de réitérer les entraînements de l'outil. Le modèle présenté ici a été utilisé sur la totalité du corpus ParCoTrain-Synt. Il est disponible à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Models/Lemmatisation`.

7.4 Parsing

Comme indiqué dans la section 3.4.3, nous avons retenu Talismane (Urieli, 2013) comme outil de parsing. Nous rappelons ici les propriétés principales et les fonctionnalités de cet outil.

Talismane intègre une chaîne de traitement complète, capable d'effectuer la tokénisation, l'étiquetage morphosyntaxique et le parsing, ce qui permet de partir d'un texte brut et d'aboutir à un corpus parsé (sous condition de disposer des modèles pour les trois modules). Par ailleurs, l'outil est doté des fonctionnalités d'évaluation produisant des résultats détaillés, ainsi que des modules d'analyse statistique du corpus. Quant au traitement des relations non projectives, l'outil exploite une méthode proche du parsing pseudo-projectif de Nivre & Nilsson (2005) (cf. section 3.4.2).

Un autre atout de Talismane est essentiel pour notre cadre de travail. Il s'agit de son système d'exploitation des traits morphosyntaxiques. Talismane permet une prise en main détaillée sur cet aspect de son fonctionnement : il est possible de définir l'ensemble des traits d'apprentissage, en faisant appel à différentes positions dans la pile et dans le *buffer* (cf. section 3.4.2). L'utilisateur a la liberté de créer des traits simples ou des traits complexes construits à partir de ces premiers.

Par ailleurs, Talismane ne puise pas les informations morphosyntaxiques du corpus d'entraînement durant l'apprentissage. Les seules informations qu'il en récupère sont les paires *token - POS*, qui lui permettent ensuite de trouver les traits morphosyntaxiques correspondants dans le lexique. Si les traits trouvés sont ambigus, toutes les valeurs possibles sont retenues. Ainsi, le parser prend en compte l'incertitude autour des traits morphosyntaxiques d'une forme ambiguë. Ceci devrait lui permettre de faire des généralisations plus fiables lors du traitement d'un texte inconnu que s'il avait été entraîné sur l'annotation désambiguïsée du corpus d'entraînement. Ce fait est particulièrement intéressant

pour nous du fait que le serbe exhibe un degré d'ambiguïté élevé (cf. section 6.1.3).

7.4.1 Entraînement initial

L'entraînement initial en parsing a été effectué sur les deux premiers échantillons du corpus (40 000 tokens), annotés manuellement dans le cadre de l'optimisation du guide d'annotation syntaxique. Il faut préciser que l'annotation de ces échantillons au niveau syntaxique a été entièrement manuelle : aucune préannotation n'a été utilisée. Or, une méthode comparable à celle utilisée avec HunPos (cf. section 7.2.1) aurait pu être envisagée. Comme il a déjà été fait mention, les travaux de Agić et al. (2013b) et de Agić & Ljubešić (2015) montrent qu'un modèle de parsing entraîné sur le croate peut être transposé sur le serbe sans grosses pertes en performances. On pouvait donc exploiter l'un de ces modèles – librement diffusés – pour préannoter nos échantillons. Cependant, cela aurait exigé un effort considérable. Tout d'abord, ces modèles ont été entraînés sur des jeux d'étiquettes morphosyntaxiques différents du nôtre. Pour qu'ils soient capables de pré-traiter notre corpus, il aurait fallu convertir notre annotation morphosyntaxique de sorte à coïncider avec celle intégrée aux outils. Deuxièmement, les modèles en question se basent également sur des jeux d'étiquettes syntaxiques nettement différents par rapport au nôtre. Il aurait donc été difficile d'exploiter une préannotation qu'ils auraient produite. Par conséquent, nous avons préféré concentrer notre effort sur l'annotation manuelle, qui nous a permis par ailleurs de combler les lacunes dans notre guide d'annotation.

La pertinence du choix de Talismane s'est confirmée dès les premiers tests. Suite à l'annotation syntaxique manuelle des deux premiers échantillons de notre corpus, Talismane a été évalué sur un corpus de 40 000 tokens. Dans ce premier entraînement, nous avons exploité seulement les parties du discours et les lemmes pour la définition des traits d'apprentissage. Une validation croisée à 10 itérations a montré que Talismane atteignait un score LAS de 76,3 points et un score UAS de 80,6 points. Il faut cependant préciser que dans ces premières expériences nous n'avons pas fait appel au module de Talismane chargé du parsing pseudo-projectif. L'évaluation a donc été faite sur des données artificiellement projectivisées. Les résultats réels sont donc plus bas, mais la différence n'est pas grande : comme indiqué dans le chapitre 11, notre corpus contient moins d'1 % de relations non projectives.

Ils sont néanmoins tout à fait satisfaisants comparé aux résultats précédents sur le serbe (cf. tableau 7.9).

Ces scores ne sont pas directement comparables vu que les conditions d'apprentissage diffèrent de manière importante d'un travail à l'autre. Ils permettent tout de même de constater que les résultats de Talismane sont prometteurs, étant donné qu'il a eu à sa disposition un corpus deux fois plus petit que celui utilisé dans les deux autres travaux.

Outil	Eval.	Taille corpus	Jeu MS	Jeu Synt	LAS	UAS
Mate (Bohnet, 2010)	(Agić & Ljubešić, 2015)	87 K	14	39	75,8	82,4
MST (McDonald et al., 2006)	(Agić et al., 2013b)	87 K	662	15	73,9	80,6
Talismane (Urieli, 2013)	le présent travail	40 K	12	60	76,3	80,6

TABLE 7.9 – Premiers résultats de Talismane comparés aux résultats existants

Le modèle constitué a été exploité pour la préannotation automatique dans le cadre de l'annotation manuelle. Son utilisation a permis d'accélérer l'annotation syntaxique de 40 % par rapport à l'annotation manuelle sans préannotation. L'utilisation du modèle et les résultats obtenus sont décrits en détail dans la section 8.6.

7.5 Mise au point des guides d'annotation

Comme indiqué dans les sections précédentes, l'initialisation des trois outils a nécessité l'annotation ou la correction manuelle d'une certaine quantité de texte. Pour l'étiquetage morphosyntaxique, il s'agit de 20 000 tokens, pour la lemmatisation 15 000, et pour l'annotation syntaxique environ 40 000. Cette annotation initiale a été effectuée par nous-mêmes, en interaction fréquente avec un deuxième annotateur expert. Ce travail nous a permis d'identifier les lacunes dans les guides d'annotation et de les compléter. Nous avons ensuite testé la capacité de ces documents à assurer une annotation manuelle cohérente. Ceci a été fait à travers l'évaluation de l'accord inter-annotateurs.

7.5.1 Mise au point du guide d'annotation morphosyntaxique

Cette tâche a été confiée à 4 étudiants serbophones, inscrits au programme LLCE - Français à l'Université de Belgrade, qui effectuaient un séjour à l'Université Toulouse - Jean Jaurès dans le cadre d'un échange Erasmus au moment de l'évaluation. 2 étudiants étaient de niveau M1, et 2 de niveau L3. Leur formation de base comporte des modules d'apprentissage du français, mais aussi de linguistique française et générale, et ils disposaient également de connaissances solides en linguistique serbe.

En tant qu'annotateur expérimenté et l'auteure des guides d'annotation, nous avons organisé une formation pour les annotateurs. Cette formation a été constituée de 2 sessions d'entraînement de 3h chacune. La première session a été consacrée à la présentation du guide d'annotation morphosyntaxique et des spécificités de certains traitements adoptés (notamment ceux qui ne coïncident pas avec la tradition grammaticale serbe). Lors de la deuxième rencontre, un entraînement pratique a été mis en place : les étudiants ont annoté ensemble un échantillon de texte authentique sous notre supervision. Toutes les décisions ont été discutées et argumentées en temps réel par les annotateurs novices et validées par

	Paire 1	Paire 2
kappa sur étiq. complètes	0,90	0,91
kappa sur POS	0,96	0,97
kappa sur traits	0,95	0,96

TABLE 7.10 – *kappa* de Cohen à différents niveaux par paire d'annotateurs

nous-mêmes.

Les paires d'annotateurs ont été constituées par tirage au sort. Chacune des paires, qui comprenait un étudiant de niveau M1 et un de niveau L3, s'est vue confier un échantillon d'environ 2000 tokens. Les deux échantillons avaient été préannotés automatiquement en utilisant HunPOS (Halácsy et al., 2007) et le modèle d'annotation développé sur le premier échantillon de 20 000 tokens (cf. section 7.2). La tâche des annotateurs a donc consisté à corriger l'annotation automatique proposée par l'outil.

Le travail a été effectué en environ 2 semaines. Les annotations produites ont été utilisées pour calculer la mesure standard de l'accord inter-annotateurs, le *kappa* de Cohen (cf. section 2.4.3). Le calcul a été fait à l'aide du langage de programmation R. Nous mesurons l'accord au niveau des étiquettes complètes, des étiquettes des parties du discours, ainsi qu'au niveau des traits morphosyntaxiques individuels. Les résultats par paire d'annotateurs sont donnés dans le tableau 7.10.

D'après l'échelle d'interprétation des valeurs de *kappa* proposée par Landis & Koch (1977) et présentée dans la section 2.4.3, ces résultats représentent un accord quasi-parfait.

Une analyse qualitative de la production des annotateurs a également été effectuée. Nous avons en particulier examiné la confusion au niveau des parties du discours. La matrice de confusion montrait que les distinctions les plus problématiques étaient celle entre les verbes et les adjectifs, mais aussi celle entre les particules, les conjonctions et les adverbes. Le premier problème était dû à la confusion entre le participe passé et l'adjectif déverbal dérivé du participe actif. Le guide d'annotation indiquait un traitement non ambigu pour ce cas de figure (cf. le traitement des participes décrit dans la section 5.1.2), mais l'un des annotateurs sentait que le traitement proposé ne correspondait pas aux faits linguistiques. Par conséquent, l'argumentation faite dans le guide d'annotation a été étendue et explicitée, et une attention particulière a été accordée à ce point dans la formation des nouveaux annotateurs.

Le deuxième point problématique provenait du fait que certains adverbes et conjonctions en serbe peuvent également se comporter comme des particules. La version initiale du guide proposait de suivre la classification donnée par Mrazović (2009), mais cette classification s'est montrée difficile à appliquer d'après les annotateurs. Nous avons donc cherché à simplifier le traitement de ces classes en prônant une distinction plus nettement basée

	Paire 1		Paire 2	
	Annot.1 (JMa)	Annot.2 (DT)	Annot.1 (JMi)	Annot.2 (II)
kappa (étiq.complètes)	0,91	0,96	0,94	0,92
kappa (traits)	0,96	0,97	0,96	0,96

TABLE 7.11 – Accord des annotateurs avec l'annotation de référence

sur des critères syntaxiques.

Suite à ce travail d'évaluation, l'annotation de référence a été constituée pour les deux échantillons. Pour ce faire, nous avons effectué une adjudication des points de divergence entre les annotateurs, et l'annotation ainsi produite a été validée par un deuxième expert linguiste. Ceci nous a permis de calculer également le taux d'accord de chacun des annotateurs par rapport à l'annotation de référence. L'accord a été calculé au niveau des étiquettes et au niveau des traits, cf. tableau 7.11.

Ces résultats étaient indicatifs de la qualité du guide d'annotation, mais aussi de la grande qualité du travail fourni par les annotateurs. En se basant sur ces mesures, mais aussi sur les échanges qui ont eu lieu avec ce groupe d'étudiants, nous avons conclu qu'ils étaient capables d'aborder l'annotation manuelle du corpus en autonomie. Nous reviendrons sur ce point dans le chapitre 8.

7.5.2 Mise au point du guide d'annotation syntaxique

Cette tâche a été confiée à deux annotatrices serbophones de l'Université de Belgrade. L'une d'entre elles faisait partie du groupe qui avait participé à l'évaluation de l'accord inter-annotateurs pour la morphosyntaxe (niveau M1), alors que l'autre avait été recrutée au Département d'études romanes à l'Université de Belgrade (niveau L3). Les deux annotatrices avaient été formées à la tâche et avaient déjà eu l'occasion de se familiariser avec l'annotation syntaxique sur corpus en utilisant notre guide d'annotation. Elles avaient effectué un minimum de 10 h d'annotation chacune avant le début de l'évaluation.

L'évaluation a été effectuée sur un échantillon de 3000 tokens. Le contenu avait été préannoté en utilisant le parser par transitions Talismane (Urieli, 2013) selon les modalités décrites dans la section 7.4. Les annotatrices avaient donc à corriger et compléter l'analyse partielle produite par l'outil.

Le travail a été effectué en environ 3 semaines. Pour estimer le taux d'accord, nous avons de nouveau utilisé la mesure de *kappa* de Cohen (cf. section 2.4). Nous nous sommes servie de la fonctionalité *evaluate* du parser Talismane pour calculer l'accord entre les deux annotatrices, mais aussi entre chacune des annotatrices et l'annotation de référence. Le calcul a été fait sur le score LAS. Les résultats sont donnés dans le tableau 7.12.

Encore une fois, les taux d'accord obtenus peuvent être considérés comme excellents

Scénario	*kappa* de Cohen
Annot1 *vs* Annot2	0,94
Annot1 *vs* Réf	0,94
Annot2 *vs* Réf	0,93

TABLE 7.12 – Accord inter-annotateurs en annotation syntaxique

selon l'échelle d'interprétation proposée par Landis & Koch (1977). Le taux d'accord des annotatrices par rapport à l'annotation de référence est également remarquable.

Afin d'identifier les points problématiques, nous avons analysé les matrices de confusion pour chacune des annotatrices. Les mêmes groupes d'étiquettes se sont avérés difficiles pour les deux.

Les étiquettes dédiées au traitement de l'ellipse ont généré le plus d'erreurs. Pour essayer de contrer cet effet, nous avons introduit des tests dans le guide d'annotation pour faciliter l'identification du gouverneur d'une forme dépendante d'un élément élidé.

Quant à l'identification des étiquettes, les confusions les plus nombreuses concernent le traitement des subordonnées complétives et du sujet logique. Les erreurs sur les complétives étaient dues à une conjonction serbe polysémique, *da*. Elle peut être équivalente de *que* en français et introduire des complétives (*Filip traži* ***da*** *Ana dođe* 'Filip demande **qu**'Ana vienne'), mais elle peut également signifier *si* dans les phrases hypothétiques irréelles (***Da*** *si došao, sve bi video* 'Si tu étais venu, tu aurais tout vu'), ou encore avoir un sens final (*Filip se sakrio* ***da*** *ga Ana ne vidi* 'Filip s'est caché **pour qu'**Ana ne le voie pas). Afin de faciliter la distinction de ces cas de figure, de nouveaux exemples ont été rajoutés au guide, ainsi que des rappels explicites de la nature problématique de la conjonction *da*.

En ce qui concerne le sujet logique, cette fonction a été confondue avec l'objet direct et l'objet indirect casuel. Ceci reflète les problèmes d'ambiguïté de la forme du sujet logique évoquées dans la section 5.2.5. Il faut souligner cependant qu'il n'y avait que 3 occurrences de cette fonction dans l'échantillon de test. Il est donc relativement difficile d'évaluer l'ampleur de ce problème. Nous avons néanmoins inclus des exemples supplémentaires dans le guide d'annotation pour illustrer plus clairement la distinction entre ces fonctions.

7.6 Bilan intermédiaire

Grâce aux démarches décrites dans ce chapitre, nous avons complété le premier stade de notre méthode : le travail de préparation pour la campagne d'annotation. Plus particulièrement, nous avons complété et évalué nos guides d'annotation, d'abord à travers une première utilisation sur les données, et ensuite dans le cadre des évaluations de l'accord

inter-annotateurs. Nous avons également effectué l'entraînement initial des trois outils et les avons ainsi préparés pour la préannotation automatique. Les différentes activités liées à ce premier entraînement nous ont menée à annoter ou valider manuellement une certaine quantité de données ; l'annotation du corpus ParCoTrain-Synt a donc également été entamée.

Tous ces facteurs réunis, nous pouvions passer au stade de la pré-campagne, dédiée au recrutement et à la formation des annotateurs, puis à la campagne d'annotation elle-même. Cette partie de notre travail est décrite dans le chapitre 8.

Chapitre 8

Campagnes d'annotation manuelle

La majorité de l'annotation manuelle de notre corpus d'apprentissage et d'évaluation a été effectuée dans le cadre de deux campagnes d'annotation avec des annotateurs étudiants. Le projet ParCoLab entretient des liens forts avec le Département d'études romanes à la Faculté de Philologie de l'Université de Belgrade. De nombreux étudiants avaient déjà participé au projet, notamment dans le cadre de la vérification manuelle des alignements et dans la préparation de différents types de contenu pour l'intégration au corpus. Ce groupe d'étudiants a servi de vivier principal pour l'identification des candidats à la tâche d'annotation manuelle.

Les deux campagnes d'annotation ont été organisées à l'Université de Toulouse - Jean Jaurès. La première campagne a duré 2 semaines, et la deuxième 3. Les étudiants retenus ont été accueillis par le laboratoire CLLE, où la majorité de leur travail s'est déroulée. Ces séjours ont été possibles grâce à un financement fourni par le projet ParCoLab, obtenu dans le cadre d'un projet bilatéral franco-serbe PHC « Pavle Savic », alloué par Campus France.

La suite de ce chapitre est dédiée aux aspects pratiques de ces campagnes. Tout d'abord, nous décrivons le stade de la pré-campagne : la sélection des interfaces d'annotation, aussi bien pour l'annotation morphosyntaxique et la lemmatisation (section 8.1) que pour l'annotation syntaxique (cf. section 8.2), alors que la section 8.3 décrit la démarche de sélection et de formation des annotateurs. Ensuite, nous présentons les conditions de travail et les résultats obtenus dans les campagnes d'annotation par niveau d'annotation : le travail en morphosyntaxe est décrit dans la section 8.4, celui en lemmatisation dans la section 8.5, et celui en syntaxe dans la section 8.6, suivi d'un bilan dans la section 8.9. La dernière phase de ce projet, la finalisation du corpus, est abordée dans la section 8.8. Enfin, la section 8.9 résume les apports des campagnes, mais aussi du processus de la constitution de ParCoTrain-Synt.

8.1 Interface d'annotation manuelle pour l'étiquetage morphosyntaxique et la lemmatisation

La relative simplicité des tâches d'annotation morphosyntaxique et de lemmatisation nous a permis d'avoir recours à un outil de base, ne nécessitant pas d'apprentissage : un tableur. En effet, la sortie de l'étiqueteur et du lemmatiseur se présente dans un format verticalisé, où une ligne correspond à un token. Chaque ligne contient le token lui-même et l'annotation qui lui a été attribuée par l'outil de traitement. Il était donc simple de traiter ces fichiers de sortie comme des fichiers .csv et de les importer dans un tableur.

Des aménagements ont été faits pour garantir un certain degré d'ergonomie, notamment pour l'annotation morphosyntaxique. Pour rappel, HunPos reproduit les étiquettes morphosyntaxiques du corpus d'apprentissage. Il s'agit d'étiquettes complexes exprimant la partie du discours, mais aussi une série de traits morphosyntaxiques sous forme de codes. Cela rend la correction de l'annotation difficile.

À titre d'illustration, un extrait de la sortie de HunPos est donné dans la figure 8.1. Dans cet exemple, le token *noćne* (forme fléchie de *noćni* 'nocturne') est annoté comme adjectif à l'accusatif pluriel du féminin au positif (A=adjectif, qual=qualificatif, acc=accusatif, pl=pluriel, f=féminin, pos=positif), et le token *hladnoće* (forme fléchie de *hladnoća* 'froid') comme nom commun féminin au nominatif pluriel (N=nom, com=commun, nom=nominatif, pl=pluriel, f=féminin). Or, les deux formes sont au génitif singulier, imposé par la préposition *od* 'de'. L'étiquette correcte pour l'adjectif serait donc A_qual_gen_sg_f_pos, et celle du nom N_com_gen_sg_f.

od	Prep
noćne	A_qual_acc_pl_f_pos
hladnoće	N_com_nom_pl_f
je	V_aux_pres_3_sg_-_-
kašljao	V_main_partact_-_sg_m_-

FIGURE 8.1 – Illustration de la sortie de HunPos

Afin de faciliter la détection des erreurs et la correction des étiquettes, nous avons mis en place un autre format, dans lequel chaque trait occupe une colonne, et les valeurs des traits sont données en mots pleins. L'extrait correspondant à celui ci-dessus est donné dans la figure 8.2.

Ce format facilite l'accès à chaque type d'information, et pour effectuer une correction, il suffit de modifier la valeur du trait erroné sans se soucier du reste de l'étiquette. Cependant, on remarque que la même colonne correspond à différents traits en fonction de la partie du discours. Pour simplifier l'identification des traits, nous avons créé un en-tête dynamique : les intitulés des colonnes, qui correspondaient aux noms des traits représentés

od	Prep						
noćne	A	opsti	akuzativ	mnozina	zenski rod	pozitiv	
hladnoće	N	opsta	nominativ	mnozina	zenski rod		
je	V	pomocni	prezent	trece lice	jednina	—	—
kašljao	V	glavni	particip_radni	—	jednina	muski rod	—

FIGURE 8.2 – Illustration de la transformation explicite de la sortie de HunPos

dans les colonnes, changent en fonction de la valeur de la colonne contenant la partie du discours. Par conséquent, dès que l'annotateur apporte une correction à la partie du discours, l'en-tête change pour indiquer les traits appropriés. Deux en-têtes différents (celui du pronom et celui du verbe) sont montrés dans la figure 8.3.

	A	B	C	D	E	F	G	H
1	**Token**	**POS**	**Lice**	**Broj**	**Rod**	**Padez**	**Kategorija**	
3	Ko	P	---	---	---	nominativ	upitna	
4	to	P	---	jednina	srednji rod	nominativ	pokazna	
5	ometa	V	prezent	trece lice	jednina	---	---	glavni
6	san	N	akuzativ	jednina	muski rod	opsta		
7	pravednika	N	genitiv	jednina	muski rod	opsta		

	A	B	C	D	E	F	G	H
1	**Token**	**POS**	**Vreme**	**Lice**	**Broj**	**Rod**	**Negacija**	**Kategorija**
3	Ko	P	---	---	---	nominativ	upitna	
4	to	P	---	jednina	srednji rod	nominativ	pokazna	
5	ometa	V	prezent	trece lice	jednina	---	---	glavni
6	san	N	akuzativ	jednina	muski rod	opsta		
7	pravednika	N	genitiv	jednina	muski rod	opsta		

FIGURE 8.3 – En-tête dynamique pour l'annotation morphosyntaxique

8.2 Interface d'annotation manuelle pour la syntaxe

Même si l'annotation en dépendances peut être représentée dans un format textuel tabulé (ce qui est en effet le cas avec les corpus d'apprentissage de parsing), ce format n'offre pas le minimum d'ergonomie nécessaire pour une annotation manuelle efficace. En effet, pour pouvoir tenir compte de toutes les interactions syntaxiques dans la phrase, il est indispensable que l'annotateur humain puisse visualiser l'arbre syntaxique qu'il est en train de créer et intervenir de manière aussi directe que possible. Nous avons considéré deux éditeurs qui permettent ce type d'interaction : TrEd de Pajas & Štěpánek (2008) [1] et brat de Stenetorp et al. (2012) [2].

1. https ://ufal.mff.cuni.cz/tred/
2. http ://brat.nlplab.org/

8.2.1 Éditeur d'arbres TrEd

TrEd est un éditeur graphique pour les structures arborescentes. L'outil est écrit en perl et il est disponible pour toutes les plateformes d'exploitation. Outre l'interface graphique, illustrée dans la figure 8.4, il dispose également d'un éditeur en ligne de commande pour l'application des macros aux fichiers et cette version de l'outil peut également être installée sur un server. TrEd a été utilisé dans la création du corpus tchèque PDT.

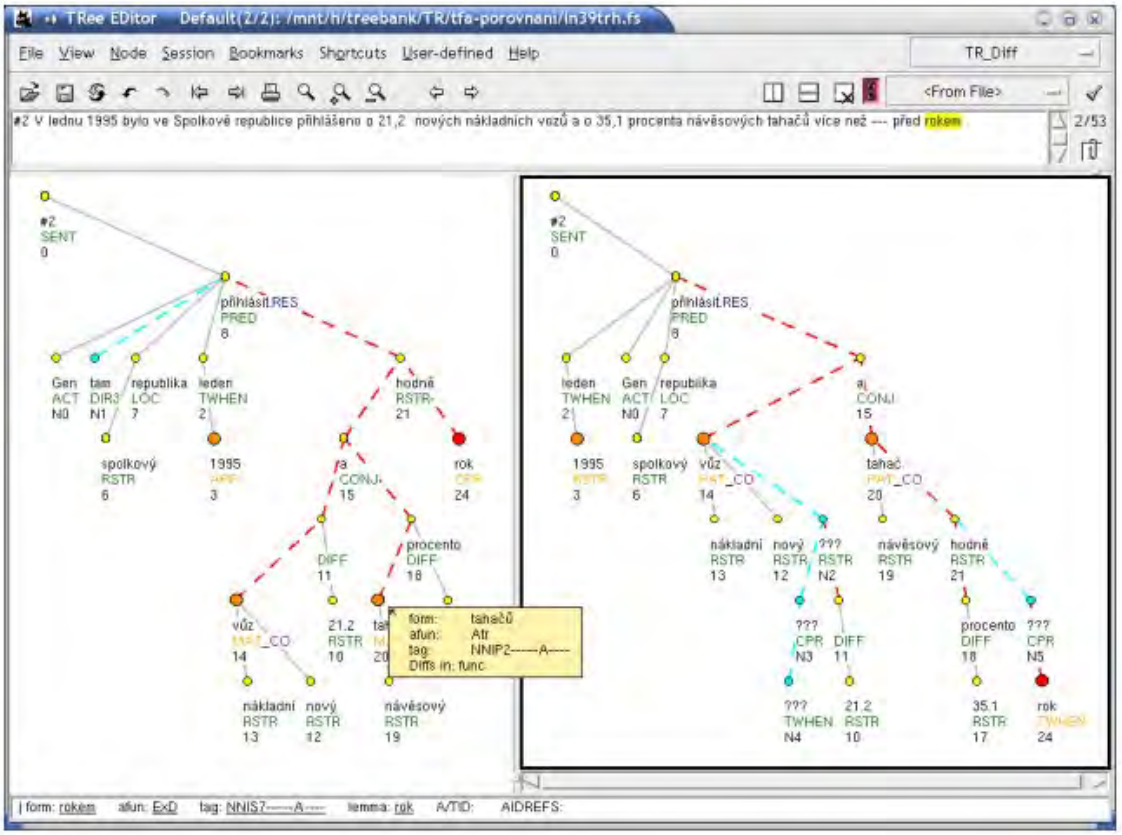


FIGURE 8.4 – Interface graphique de TrEd

Son format d'entrée/sortie par défaut est un format XML nommé PML (Prague Markup Language). Il s'agit d'un format complexe mettant en place des annotations à plusieurs niveaux. Les dépendances syntaxiques sont exprimées de manière suivante : chaque dépendant est représenté comme un nœud-fille de son gouverneur. Des éléments et attributs dédiés indiquent la forme des tokens, leur position linéaire dans la phrase de départ et leur fonction syntaxique. Un exemple représentant l'annotation des phrases *John loves Mary. He told her this Friday.* est donné dans la figure 8.5 [3]. Une extension pour la conversion

3. XML repris de la documentation du PML à l'adresse suivante : `http://ufal.mff.cuni.cz/jazz/PML/doc/pml_doc.html#id2534176`. Dernier accès : le 24 octobre 2017.

du format CoNLL2009 vers PML est proposée par la communauté TrEd [4].

```
<?xml version="1.0"?>
<annotation xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
  <head>
    <schema href="example1_schema.xml"/>
  </head>
  <meta>
    <annotator>Jan Novak</annotator>
    <datetime>Sun May 1 18:56:55 2005</datetime>
  </meta>
  <trees>
    <LM ord="2">
      <func>Pred</func>
      <form>loves</form>
      <governs>
        <LM ord="1">
          <func>Subj</func>
          <form>John</form>
        </LM>
        <LM ord="3">
          <func>Obj</func>
          <form>Mary</form>
        </LM>
      </governs>
    </LM>
```

FIGURE 8.5 – Exemple du format PML de TrEd

8.2.2 Éditeur de dépendances brat

Brat est un éditeur d'annotations implémenté en python. Il est destiné en premier lieu à une utilisation sur serveur, mais une installation locale en mode *stand-alone* est également possible. D'après la documentation de l'outil, brat est adapté à l'annotation dans le cadre des tâches comme la reconnaissance d'entités nommées, l'extraction d'événements, la résolution de la coréférence, etc. Il propose une interface graphique agréable et intuitive (cf. figure 8.6) et il a été exploité pour l'annotation de plusieurs corpus, notamment dans les domaines médical et biomédical : Anatomical Entity Mention (Ohta et al., 2012), CellFinder (Neves et al., 2012) et Multi-Level Event Extraction corpus (Pyysalo et al., 2012).

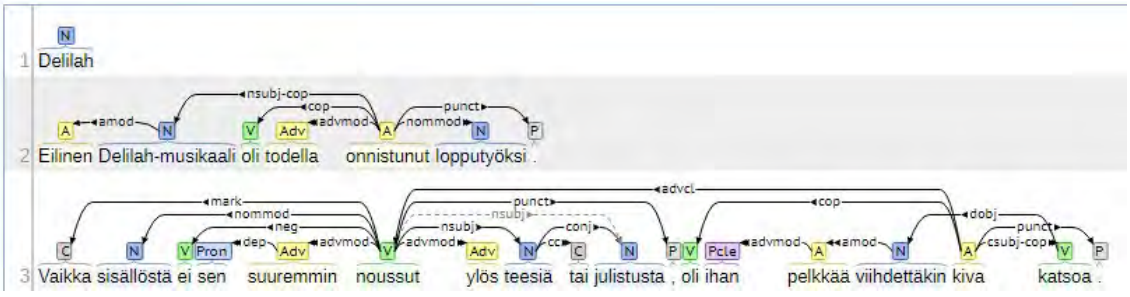


FIGURE 8.6 – Interface graphique de brat

En entrée et sortie, brat exploite un format textuel, avec des annotations déportées (cf. figure 8.7). Dans ce format, la première colonne représente l'identifiant de l'annotation, la deuxième donne l'étiquette et les *offset* du/des token(s) qui la portent, et la dernière

4. `http://ufal.mff.cuni.cz/tred/extensions/core/conll2009/documentation/index.html`

colonne indique le token lui-même. Des convertisseurs du format CoNLL-X vers le format brat et *vice versa* sont disponibles.

```
T1      Organization 0 4                Sony
T2      MERGE-ORG 14 27                 joint venture
T3      Organization 33 41              Ericsson
E1      MERGE-ORG:T2 Org1:T1 Org2:T3
T4      Country 75 81                   Sweden
R1      Origin Arg1:T3 Arg2:T4
```

FIGURE 8.7 – Exemple du format *standoff* de brat

L'avantage principal de brat par rapport à TrEd est son mode de fonctionnement serveur. Pour rappel, TrEd propose le fonctionnement sur serveur seulement pour sa version en ligne de commande, adaptée à l'application des transformations à de grandes quantités de données, mais pas à l'annotation manuelle. Brat, en revanche, permet de faire des annotations en utilisant le serveur. Il est donc possible d'apporter des annotations à distance et d'avoir plusieurs annotateurs qui travaillent sur le même fichier en temps réel. Qui plus est, les annotateurs intervenant sur un projet sont dispensés d'installer l'outil : il suffit de se rendre sur le serveur en utilisant le navigateur Google Chrome [5]. Par ailleurs, cette fonctionalité de l'outil avait déjà été utilisée par des membres du laboratoire CLLE, si bien que l'infrastructure nécessaire était déjà en place. Pour toutes ces raisons, nous avons retenu brat comme outil d'annotation syntaxique manuelle.

L'interface graphique est simple et intuitive. Les annotations morphosyntaxiques sont représentées sous forme d'étiquettes en dessous des tokens. Pour les modifier, il suffit de double-cliquer sur l'étiquette existante, ce qui ouvre une fenêtre pop-up, qui affiche la totalité du jeu d'étiquettes et permet de sélectionner l'étiquette correcte. Les relations se posent en cliquant sur le gouverneur et en tirant la flèche de la souris jusqu'au dépendant, ce qui permet de poser un arc de dépendance. En lâchant le bouton de la souris, une fenêtre pop-up s'ouvre dans laquelle on peut sélectionner la relation souhaitée. Les étiquettes syntaxiques déjà en place peuvent être modifiées de la même manière que les étiquettes morphosyntaxiques [6].

Une caractéristique particulièrement intéressante de brat réside dans sa fonctionnalité de définition du schéma d'annotation. En effet, avant de démarrer le travail, il est nécessaire de définir les règles d'annotation par le biais du fichier de configuration d'annotation *annotation.conf*. C'est un fichier textuel dans lequel on peut définir les étiquettes pour tous les niveaux d'annotation. Les seules étiquettes admises par l'outil sont celles qui

5. La documentation de l'outil précise que brat est optimisé pour ce navigateur spécifique.

6. Brat propose également la possibilité de paramétrer l'affichage à travers le fichier de configuration visuelle (*visual.conf*), où il est possible de définir la couleur des nuages portant les annotations morphosyntaxiques, la forme et la couleur des flèches des relations, ainsi que les abréviations pour les étiquettes. Nous n'avons pas exploré en détail ces possibilités, nous contentant des paramètres par défaut de l'outil.

figurent dans ce fichier. Si le texte à traiter contient des étiquettes non comprises dans la configuration, brat signale une erreur.

Brat propose 4 types d'annotation : entités (annotations liées aux segments de texte), relations (relations entre les entités), évènements (annotations impliquant plusieurs relations) et attributs (annotations associées aux entités). Dans ce travail, nous exploitons les entités, qui correspondent à l'annotation morphosyntaxique, et les relations, qui correspondent aux dépendances syntaxiques. Pour les entités, la configuration du schéma d'annotation est relativement simple : il suffit de lister les étiquettes morphosyntaxiques valides dans *annotation.conf*. Quant aux relations, la situation est plus complexe : pour chaque label de relation, le fichier de configuration d'annotation doit contenir l'indication des étiquettes des entités qui peuvent avoir le rôle du gouverneur et du dépendant de la relation en question. À partir de ces définitions, brat sélectionne les étiquettes syntaxiques licites dans la fenêtre pop-up lorsqu'un arc de dépendance est posé.

Ce mode de fonctionnement présente à la fois des avantages et des inconvénients. D'un côté, cette contrainte peut aider à minimiser les erreurs humaines : la liste des relations proposées pour chaque arc dessiné ne contient qu'une portion du jeu d'étiquettes syntaxiques, ce qui diminue la probabilité de sélectionner la mauvaise fonction par mégarde. Mais ce besoin de fournir une définition en extension du gouverneur et dépendant de chaque relation est problématique car il présuppose une connaissance parfaite de tous les cas de figure possibles pour chaque relation. C'est notamment très contraignant dans les premières étapes de l'annotation syntaxique manuelle, qui sont faites avec l'objectif d'observer les données et de raffiner le schéma d'annotation. Ce processus nous a permis d'identifier de nombreuses modifications nécessaires. Or, le fichier *annotation.conf* n'est accessible que sur le serveur qui héberge brat. Par conséquent, seul un administrateur de serveur pouvait modifier la configuration. Ce fait rendait les ajustements du schéma relativement coûteux, menant à la situation où l'on attendait que plusieurs changements soient signalés comme nécessaires pour effectuer la modification. En attendant, les annotateurs devaient continuer à travailler avec la configuration existante, tout en marquant les cas problématiques pour y revenir une fois la configuration mise à jour.

Cette particularité mise à part, brat est un outil ergonomique et pratique. Plusieurs de ses fonctionnalités ont été appréciées durant ce travail, notamment la possibilité de laisser des commentaires sur chaque annotation posée, ainsi que son interface de recherche, qui permet de formuler des requêtes sur le texte, les annotations morphosyntaxiques ou syntaxiques, ou bien les commentaires, et même de croiser ces paramètres. La propriété qui a été le plus mise en avant par les annotateurs humains était le fait que l'outil n'exigeait pas de sauvegardes explicites : chaque modification apportée aux fichiers était instantanément enregistrée sur le serveur. Le risque de perte de données était ainsi réduit au minimum. De manière générale, l'avis des annotateurs par rapport à l'outil a été très positif.

8.3 Sélection et formation des annotateurs

Le recrutement et la formation des annotateurs qui seront chargés de l'annotation manuelle du corpus sont d'une importance cruciale pour la réussite de la campagne. Ils représentent donc la tâche la plus importante dans le stade de la pré-campagne. Alors que l'aspect pratique de ce travail peut être pris en charge par le gestionnaire, l'essentiel revient aux annotateurs expérimentés : c'est à eux, en tant qu'experts linguistes, d'évaluer les compétences des candidats, de définir le contenu de la formation et de la dispenser. Dans le cadre de cette thèse, les deux rôles ont été pris en charge par nous-mêmes.

En recrutant les annotateurs qui seront chargés de l'annotation manuelle, nous cherchions plusieurs qualités spécifiques : tout d'abord, les étudiants en question devaient avoir une excellente maîtrise des notions morphosyntaxiques et syntaxiques générales, mais aussi une connaissance approfondie de la grammaire traditionnelle du serbe. Ce deuxième point était important pour la raison suivante : la majorité des traitements que nous mettons en place se définissent par rapport à la grammaire serbe, soit en la reprenant, soit en la remplaçant par une autre approche. Une bonne maîtrise de la grammaire serbe facilite donc la compréhension de nos schémas d'annotation. Ensuite, les étudiants devaient également avoir une capacité d'apprentissage et d'adaptation importante, afin de pouvoir assimiler rapidement les guides et prendre en compte les modifications apportées aux traitements traditionnels. Enfin, il était également fortement souhaitable qu'ils soient motivés et constants, capables de maintenir un niveau élevé de concentration sur des tâches qui peuvent paraître répétitives.

Afin de permettre l'évaluation des deux groupes de critères, une sélection en deux temps a été organisée. La première étape comportait une évaluation sur dossier, prenant en compte l'excellence académique des candidats, mais aussi l'évaluation de leurs qualités personnelles. Ce travail a été confié à S. Marjanović, un enseignant-chercheur au Département d'études romanes à l'Université de Belgrade. S. Marjanović est en effet le coordinateur du côté serbe du projet ParCoLab. Il connaissait donc les candidats comme participants du projet ParCoLab, mais aussi en tant qu'étudiants. Le deuxième volet de la sélection consistait en un test de morphosyntaxe et de syntaxe du serbe. Le test portait majoritairement sur la grammaire traditionnelle du serbe, mais il contenait également des questions relevant de points problématiques souvent ignorés par les grammaires serbes, qui permettaient d'évaluer le raisonnement linguistique des candidats. Les candidats finaux ont été sélectionnés en prenant en compte les résultats combinés des deux étapes.

Pour les deux campagnes d'annotation, l'appel à candidatures a été diffusé au Département d'études romanes de l'Université de Belgrade, mais il a également été transmis à des candidats d'autres institutions repérés par les membres seniors du projet ParCoLab. Pour l'annotation morphosyntaxique, un groupe de 4 annotateurs a été retenu, dont 3 de

l'Université de Belgrade (L3 en LLCE français), et un de l'Université de Gênes en Italie (M1 en LEA français et anglais, avec une forte composante en linguistique théorique). La campagne d'annotation syntaxique a été réalisée par 2 annotatrices de l'Université de Belgrade, dont une était du niveau L3, et l'autre du niveau M1, les deux inscrites au parcours LLCE français au Département d'études romanes.

Avant le début de la campagne d'annotation morphosyntaxique, nous avons mis en place et dispensé une formation pour les annotateurs à l'Université de Belgrade. Un total de 5h a été consacré à l'annotation morphosyntaxique, alors que 4h ont été dédiées à l'annotation syntaxique. Le contenu et l'organisation de la formation sont présentés dans le tableau 8.2.

Séance	Durée	Contenu
Séance 1	2 h	Analyse du test de sélection, présentation du jeu d'étiquettes et du schéma d'annotation
Séance 2	3 h	Entraînement en annotation morphosyntaxique fine sur un extrait du corpus
Séance 3	1 h 30	Introduction à la syntaxe de dépendances, analyse des exemples de base en termes généraux (sans faire appel au jeu d'étiquettes syntaxiques)
Séance 4	1 h 30	Présentation du jeu d'étiquettes de ParCoLab, notamment des phénomènes syntaxiques complexes (coordination, subordination, juxtaposition, ellipse)
Séance 5	1 h	Familiarisation avec l'interface d'annotation à travers l'annotation d'un échantillon du corpus

TABLE 8.2 – Contenu et organisation de la formation des annotateurs pour la première campagne d'annotation

Il était initialement prévu de faire travailler le même groupe d'annotateurs sur les trois couches d'annotation (morphosyntaxe, lemmatisation, syntaxe). Cependant, à l'issue de cette initiation, nous avons constaté que les étudiants avaient une maîtrise tout à fait solide en morphosyntaxe, mais que leurs compétences en analyse syntaxique étaient moins bonnes. Les étudiants avaient également fait part de leur malaise face à cette tâche, en citant notamment leur manque de familiarité avec la syntaxe en dépendances. En accord avec ces observations, nous avons limité la tâche des annotateurs de la première campagne à l'annotation morphosyntaxique et à la lemmatisation. L'annotation syntaxique a fait l'objet d'une deuxième campagne (cf. section 8.6)

Ceci nous a permis de mieux cibler les étudiants avec des compétences fortes en syntaxe pour la deuxième campagne. Cependant, comme cette deuxième étape du travail a été organisée dans des délais plus courts que la première, il n'a pas été possible de proposer

une formation initiale à Belgrade. Nous avons essayé de compenser ce fait en consacrant les deux premières journées de travail à Toulouse à une présentation détaillée du guide et de l'interface d'annotation. Le guide d'annotation avait été transmis aux annotatrices avant leur arrivée et elles avaient eu l'occasion d'explorer son contenu. Par conséquent, la partie de la formation consacrée au guide a été orientée par leurs questions et a majoritairement porté sur les structures complexes.

Outre cette entrée en matière par la maîtrise des guides d'annotation, les deux groupes d'annotateurs ont été sensibilisés à l'intérêt des tâches qu'ils avaient à exécuter. La chaîne de traitement leur a été présentée dans sa totalité, et l'effet des erreurs introduites à tout niveau d'analyse présenté de manière détaillée. Il leur a également été expliqué que leurs efforts allaient contribuer à améliorer les performances des outils automatiques utilisés. Après la fin de chaque cycle de validation manuelle, les nouveaux résultats des outils réentraînés leur ont été communiqués ; ainsi, ils avaient un indicateur objectif de l'impact de leur travail. Nous estimons que cette approche a permis de renforcer le sentiment d'appartenance au projet et de maintenir le niveau d'implication des étudiants.

8.4 Campagne 1 : annotation manuelle au niveau morphosyntaxique

La première campagne d'annotation a eu lieu au département SDL de l'Université de Toulouse - Jean Jaurès du 5 au 16 décembre 2016 avec un groupe de 4 annotateurs serbophones, dont 3 étudiantes de niveau L3 inscrites au parcours LLCE français à l'Université de Belgrade, et un étudiant de niveau M1 dans un parcours de LEA français-anglais à l'Université de Gênes en Italie. Comme mentionné dans la section précédente, cette première campagne a été limitée à l'annotation morphosyntaxique et à la lemmatisation. Pour dégager plus clairement les apports de la campagne à chacun des deux niveaux de traitement, nous les présentons tour à tour. Le travail en annotation morphosyntaxique est détaillé dans la présente section, alors que le travail en lemmatisation est abordé dans la section 8.5. Les conditions globales de travail sont donc les mêmes dans les deux cas.

Étant donné la durée relativement courte de la campagne, une véritable évaluation de l'accord inter-annotateurs n'a pas été effectuée avec ce groupe d'étudiants. En revanche, la première journée de travail a été consacrée à un test de maîtrise du guide d'annotation : les annotateurs ont effectué, indépendamment les uns des autres, l'annotation d'un échantillon minimal de 200 tokens, pour lequel une annotation de référence avait déjà été établie. Les erreurs relevées dans leurs productions respectives étant rares ($<$4 % pour tous les annotateurs), nous avons estimé que les annotateurs étaient prêts à aborder l'annotation du corpus proprement dite. Il faut tout de même souligner que le travail était effectué en notre présence, et que les annotateurs nous présentaient tous les points problématiques

rencontrés dans les textes. Ainsi, toute difficulté était immédiatement discutée et résolue. Ce dispositif visait à minimiser les divergences entre les annotateurs. Il a par ailleurs mené à des échanges intéressants et stimulants sur certains points du schéma d'annotation.

8.4.1 Déroulement et résultats de l'annotation manuelle

Pour rappel, le corpus ParCoTrain-Synt a été divisé en 5 échantillons d'environ 20 000 tokens chacun. L'état de l'avancement de l'annotation morphosyntaxique au démarrage de cette campagne est donné dans le tableau 8.4.

Échantillon	Taille	Annot. morphosynt.	Méthode d'annotation
1_20	20 918	Oui	Préannotation avec modèle croate de HunPos ; validation par l'annotateur expérimenté
2_20	20 619	Oui pour 4 K tokens	Préannotation avec modèle de HunPos ré-entraîné sur 1_20 ; validation de 4 K tokens par les annotateurs UT2J dans le cadre des évaluations de l'accord inter-annotateurs
3_20	19 339	Non	-
4_20	20 668	Non	-
5_20	20 796	Non	-

TABLE 8.4 – L'état de l'annotation morphosyntaxique avant le début de la campagne

Le travail des annotateurs était organisé en deux séances de 2 h par jour. La première tâche de la campagne a consisté à compléter l'annotation de l'échantillon 2_20, entamée dans le cadre de l'évaluation de l'accord inter-annotateurs (cf. section 7.5.1). Il s'agissait donc d'une préannotation effectuée avec HunPos utilisant le modèle entraîné sur les 20 000 tokens de l'échantillon 1_20 (cf. section 7.2). Comme décrit dans la section 8.1, ce travail a été réalisé dans un tableur Excel, en exploitant une version explicitée des étiquettes morphosyntaxiques produites par HunPos. Chacun des quatre annotateurs s'est vu attribuer un échantillon d'environ 3 000 tokens. Ce premier objectif a été atteint en environ 9 h, correspondant à un total de 36 h de travail.

La validation de la totalité de l'échantillon 2_20 a permis d'augmenter le corpus d'entraînement existant et de ré-entraîner HunPos, cette fois sur une ressource de 40 000 tokens. Le nouveau modèle a été utilisé pour préannoter l'échantillon 3_20, qui a ensuite été divisé en portions de 1000 tokens et réparti entre les annotateurs. La correction manuelle a été effectuée en environ 15 h, soit 60 h de travail au total.

Au moment où ce travail s'est terminé, le séjour des annotateurs était arrivé à son

terme [7]. Cependant, les annotateurs ont exprimé le souhait de continuer à contribuer au projet et de poursuivre l'annotation à distance. Durant les 6 semaines après le retour des annotateurs, la validation des échantillons 4_20 et 5_20 a été réalisée. La totalité du corpus a donc été dotée d'une annotation morphosyntaxique détaillée dans le cadre de cette campagne.

8.4.2 Performances de l'étiqueteur et vitesse des annotateurs humains

Ce travail nous a également permis d'observer deux indicateurs relatifs à la pertinence de la méthode globale adoptée, à savoir la courbe d'apprentissage de l'outil HunPos et l'évolution de la vitesse d'annotation chez les annotateurs novices. L'exactitude de l'outil a été évaluée lors de chaque ré-entraînement par le biais d'une validation croisée à 10 itérations, nous permettant de calculer l'exactitude moyenne de HunPos pour chaque taille de corpus d'entraînement. En ce qui concerne la vitesse d'annotation, les annotateurs notaient le nombre de tokens traités pour chaque heure d'annotation et la valeur moyenne a été calculée pour chaque cycle d'annotation. Les résultats obtenus pour ces deux paramètres sont donnés dans le tableau 8.5.

Échantillon	2_20	3_20	4_20	5_20
Exactitue moyenne de HunPos	78,82 %	82,37 %	83,95 %	85,00 %
Taille du corpus d'entraînement	20 K	40 K	60 K	80 K
Vitesse d'annot. moyenne	410 tok/h	520 tok/h	650 tok/h	710 tok/h

TABLE 8.5 – Exactitude de HunPos et vitesse d'annotation manuelle durant la campagne 1

On observe que les performances de l'outil ainsi que celles des annotateurs se sont améliorées de façon stable et continue tout au long de cette expérience. Il est également remarquable qu'à la fin de la campagne les annotateurs novices s'étaient rapprochés de la vitesse d'annotation de l'annotateur expérimenté, qui traitait en moyenne 800 tokens/h lors de la validation de l'échantillon 1_20 (cf. section 7.2.3).

De manière plus précise, pour HunPos, nous remarquons que le passage d'un corpus d'entraînement de 20 000 tokens à un corpus de 40 000 apporte un gain important de 3,55 %. L'augmentation suivante ramène +1,58 % supplémentaires, et la courbe semble se stabiliser avec la dernière, rajoutant +1,05 % d'exactitude [8].

Quant à la vitesse d'annotation manuelle, les gains respectifs sont de 27 %, 25 % et 10 % par rapport à l'étape précédente. Il est justifié de se demander si ces résultats sont dus

7. Les annotateurs ont également effectué la lemmatisation des mêmes échantillons durant cette campagne. Plus de détails sur ce point seront proposés dans la section 8.5.

8. Le dernier modèle est disponible à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Models/Tagging/HunPos`.

à l'effet de l'apprentissage et à la maîtrise croissante de la tâche par les annotateurs au fur et à mesure du déroulement de la campagne. Cela est surtout plausible pour les premiers cycles d'annotation, durant lesquels les annotateurs étaient effectivement encore en train d'établir leurs mécanismes de travail. Rappelons cependant qu'à la fin de l'annotation de l'échantillon 3_20, les annotateurs avaient traité environ 10 000 tokens chacun. Il semble raisonnable de supposer que cette quantité de travail est suffisante pour stabiliser la maîtrise de la tâche. Même si un effet d'apprentissage continue à exister, on peut faire l'hypothèse que son amplitude est moindre dans les deux derniers cycles de travail. Cela signifie que les améliorations observées sur les échantillons 4_20 et 5_20, de 25 % et 10 % respectivement, sont au moins en partie dues à l'amélioration des performances de l'outil. Bien qu'il reste délicat d'évaluer de manière précise la valeur ajoutée par les performances augmentées de l'outil, ces observations valident les suppositions de départ sur lesquelles est basée la méthode de constitution de corpus adoptée : les augmentations successives du corpus d'entraînement permettent d'améliorer les outils de prétraitement automatique, ce qui à son tour facilite la tâche des annotateurs humains, menant à une création de corpus plus aisée et plus rapide.

8.5 Campagne 1 : lemmatisation manuelle

La lemmatisation manuelle du corpus a été effectuée lors de la première campagne d'annotation manuelle, par le même groupe d'annotateurs que l'annotation morphosyntaxique (voir la section 8.4). Comme mentionné précédemment, les annotateurs se servaient d'un tableur Excel pour ce niveau d'annotation : le texte était verticalisé, avec chaque token et son annotation donnés sur une ligne du tableur. La tâche des annotateurs consistait donc à vérifier et à corriger si nécessaire le lemme indiqué pour chaque forme fléchie. Comme stipulé par l'organisation en cascades du prétraitement automatique (cf. section 4.3), la lemmatisation suivait l'étiquetage morphosyntaxique : après la validation de l'étiquetage automatique d'un échantillon, l'échantillon était préannoté par CST et ensuite validé par les annotateurs. L'organisation exacte du travail est présentée dans la suite.

8.5.1 Déroulement et résultats de la lemmatisation manuelle

La lemmatisation des cinq échantillons du corpus d'apprentissage ne s'est pas déroulée dans le même ordre que l'annotation morphosyntaxique. Cela est dû en premier lieu au fait qu'une partie du contenu de notre corpus avait déjà été lemmatisée manuellement dans le cadre du travail de Miletic (2013). Ce contenu correspond à une partie de l'échantillon 3_20 et à la totalité des échantillons 4_20 et 5_20. De plus, nous avions nous-mêmes traité l'échantillon 1_20 lors de l'initialisation de CST (cf. section 7.3.1). Par conséquent, au moment du démarrage de la campagne d'annotation, il restait encore environ 34 000

tokens en attente de lemmatisation : la totalité de l'échantillon 2_20 et environ 14 000 tokens de l'échantillon 3_30 (cf. tableau 8.7).

Échantillon	Taille	Lemmatisation	Méthode d'annotation
1_20	20 918	Oui	Préannotation avec premier modèle de CST ; validation manuelle par A.M.
2_20	20 619	Non	-
3_20	19 339	Oui pour 5 K tokens	Annotation manuelle (cf. Miletic, 2013)
4_20	20 668	Oui	*idem*
5_20	20 796	Oui	*idem*

TABLE 8.7 – État de la lemmatisation au démarrage de la campagne d'annotation

Entre la validation de l'échantillon 1_20 et le début de cette campagne d'annotation, un nouveau modèle de lemmatisation a été entraîné en utilisant le lexique combiné ParCoLex. Ce nouveau modèle ayant des performances élevées (cf. section 7.3.2), il a été utilisé pour préannoter la totalité des 34 000 tokens restants. Les 20 000 tokens provenant de l'échantillon 2_20 ont été validés durant le séjour des annotateurs à Toulouse, alors que les 14 000 tokens de l'échantillon 3_20 ont été traités par un annotateur à distance après son retour. La lemmatisation a donc été effectuée sur la totalité du corpus.

8.5.2 Performances de CST et vitesse des annotateurs humains

Comme cette partie de l'annotation n'a pas suivi le schéma global des ré-entraînements réitérés, il n'était pas possible d'observer d'aussi près l'évolution des performances de l'outil et la vitesse de correction manuelle. Quelques informations ont tout de même pu être recueillies, cf. le tableau 8.8.

Mode d'annotation	Exactitude de l'outil	Vitesse de correction manuelle
Manuelle	-	825 tok/h
Préannotation avec modèle 1 de CST	86,2 %	1400 tok/h
Préannotation avec modèle 2 de CST	96,5 %	3200 tok/h

TABLE 8.8 – Exactitude de CST et vitesse de validation manuelle durant la campagne 1

Les résultats dans les deux premières lignes du tableau 8.8 ont été obtenus par l'annotateur expérimenté. En revanche, les résultats obtenus sur la partie du corpus préannotée avec le modèle 2 de CST sont comparables pour l'annotateur expérimenté et les annotateurs novices, le premier ayant effectué la correction d'un échantillon de 1000 tokens dans l'objectif même d'évaluer sa vitesse. On peut en déduire que ce niveau de qualité de la pré-

annotation permet d'éliminer les différences d'efficacité entre les annotateurs expérimentés et les annotateurs novices. Il est remarquable que l'annotation de l'échantillon 2_20 n'ait pris que 2 h avec quatre annotateurs travaillant en parallèle, autrement dit 8 h de travail au total. L'échantillon restant de 14 000 tokens a exigé 4 h de travail supplémentaires.

L'augmentation de la vitesse paraît ici encore plus nette que dans le cas de l'annotation morphosyntaxique, mais elle va de pair avec l'amélioration importante des performances de la qualité de la préannotation. Par ailleurs, s'agissant d'une tâche plus simple que l'annotation morphosyntaxique, on peut faire l'hypothèse que l'effet d'apprentissage est moins prononcé ici. Ces résultats semblent donc montrer plus clairement l'impact que peut avoir la qualité de la préannotation sur les performances des annotateurs humains.

8.6 Campagne 2 : annotation syntaxique manuelle

La deuxième campagne d'annotation a eu lieu du 20 mars au 7 avril 2017. Elle a été réalisée par deux annotatrices recrutées à l'Université de Belgrade, l'une de niveau L3, et l'autre de niveau M1, les deux inscrites au parcours LLCE - français au Département des études romanes. L'organisation du travail était comparable à celle durant la première campagne : les annotatrices effectuaient 5 heures d'annotation par jour divisées en deux parties, sous la surveillance de l'annotateur expérimenté. Comme pour l'annotation morphosyntaxique, les annotatrices travaillaient indépendamment l'une de l'autre. Encore une fois, la décision de ne pas mettre en place une annotation en double a été motivée par le séjour relativement court des annotateurs et l'impératif de maximiser leur rendement afin de faire avancer l'annotation du corpus.

8.6.1 Déroulement du travail et résultats de l'annotation syntaxique manuelle

Au début de la campagne, les deux premiers échantillons du corpus avaient déjà été annotés en dépendances syntaxiques de manière manuelle dans le cadre de la mise au point et de l'évaluation du guide d'annotation syntaxique et de l'initialisation de Talismane (cf. sections 7.5.2 et 7.4). Il restait donc 60 000 tokens à traiter (cf. tableau 8.10).

Le parser Talismane avait été entraîné sur le corpus manuellement annoté de 40 000 tokens. Cet apprentissage initial a été effectué avec les paramètres d'apprentissage par défaut, en exploitant seulement la partie du discours et le lemme en tant que trait morphosyntaxique pour l'entraînement [9]. L'outil a été évalué par une validation croisée à 10 itérations. Les valeurs moyennes de LAS et de UAS obtenues étaient respectivement de

9. Des tests détaillés exploitant les différentes propriétés morphosyntaxiques ont été conduits depuis la finalisation du corpus d'apprentissage et sont présentés dans le chapitre 9.

Échantillon	Taille	Annot. synt.	Mode d'annotation
1_20	20 918	Oui	Annotation manuelle par l'annotateur expérimenté
2_20	20 619	Oui	Annotation manuelle par l'annotateur expérimenté et par les annotateurs UT2J dans le cadre de la mise au point du guide d'annotation
3_20	19 339	Non	-
4_20	20 668	Non	-
5_20	20 796	Non	-

TABLE 8.10 – Avancement de l'annotation syntaxique au début de la campagne d'annotation

76,34 % et de 84,06 % (voir la section 7.4.1 pour les détails). Ces résultats ont été jugés satisfaisants et le modèle obtenu a été utilisé pour préannoter l'échantillon 3_20.

Dans cette étape du travail, une fonctionnalité de Talismane s'est montrée particulièrement utile. Il s'agit de la possibilité d'obtenir dans la sortie du parser les probabilités pour chaque étiquette émise. Nous avons exploité cette possibilité pour trier l'annotation produite de sorte à ne garder que les annotations avec une valeur de probabilité supérieure à 0,85. Nous espérions ainsi permettre aux annotateurs de bénéficier des analyses les plus fiables, en leur épargnant le besoin d'analyser et rectifier des dépendances mal posées. Une fois le filtre appliqué, 11 363 tokens gardaient leur annotation, ce qui correspond à 59 % de l'échantillon. C'est donc cette annotation partielle qui a été importée dans brat et qui a été corrigée et complétée manuellement par les annotateurs (cf. figure 8.8).

FIGURE 8.8 – Annotation partielle importée dans brat

L'échantillon 3_20 a été validé en 7 jours, correspondant à un total de 70 h de travail. L'échantillon validé devait alors être joint au corpus d'entraînement existant pour permettre un nouvel entraînement de Talismane. Or, ce processus est relativement complexe et exige de convertir les données du format brat vers le format CoNLL, mettre à jour les fichiers de configuration d'apprentissage de Talismane, ré-entraîner l'outil, utiliser le nouveau modèle pour annoter l'échantillon 4_20, filtrer l'annotation produite, convertir les fichiers au format brat et les importer sur le serveur d'annotation. Ces activités auraient pris une demi-journée, alors qu'il ne restait que 6 jours avant la fin de la campagne. Comme

les annotateurs avaient exprimé leur satisfaction quant à la qualité de la préannotation, il a été décidé de ne pas passer par ce processus, mais d'annoter plutôt l'échantillon 4_20 avec le modèle existant. Le travail a donc été poursuivi sur le nouvel échantillon et la validation a pu être finalisée avant le départ des annotateurs, en 60 h de travail.

Après leur retour à Belgrade, l'une des annotatrices a continué le travail sur l'échantillon 5_20. Comme le temps n'était plus un facteur limitant, un ré-entraînement a été fait sur le corpus validé jusque-là, contenant 80 000 tokens. L'entraînement a été effectué dans les mêmes conditions que le premier. Les scores de l'outil se sont améliorés de manière importante : Talismane a obtenu 84,20 % en LAS et 89,70 % en UAS. Sur l'échantillon 5_20, 14 980 tokens (72 %) ont reçu des annotations avec un niveau de confiance supérieur à 0,85. La validation manuelle a été effectuée en 55 h de travail réparties sur 6 semaines. L'annotation syntaxique du corpus dans sa totalité a donc également été finalisée.

Nous n'avons pas effectué d'analyse d'erreur systématique ; néanmoins, dans les phases de discussion dédiées au retour d'expérience, les annotatrices nous ont indiqué que la préannotation était la plus fiable sur les relations intra-propositionnelles, notamment sur les dépendants directs du verbe et sur les relations à l'intérieur du groupe nominal. En revanche, les relations liées à la coordination et à la subordination étaient souvent problématiques.

8.6.2 Performances de Talismane et vitesse des annotateurs humains

Même s'il y a eu moins de cycles d'entraînement pour l'annotation syntaxique que pour l'annotation morphosyntaxique, nous avons tout de même pu observer les résultats du parser et la vitesse d'annotation des annotateurs humains en fonction de la taille du corpus d'entraînement utilisé pour la création du modèle de préannotation. La vitesse des annotatrices en annotation manuelle intégrale a été évaluée au démarrage de la campagne, sur des échantillons non annotés d'environ 2 000 tokens.

Le tableau 8.12 indique que l'annotation syntaxique est la tâche la plus lente parmi les trois considérées. Les mêmes observations globales restent par ailleurs valides ici : l'augmentation du corpus d'apprentissage mène à une amélioration des scores de l'outil automatique, et la vitesse d'annotation manuelle suit cette tendance. Bien qu'il soit difficile d'estimer l'effet d'apprentissage sur la vitesse d'annotation manuelle, ces améliorations importantes suggèrent encore une fois que la méthode de travail choisie était pertinente et efficace.

8.7 Bilan des campagnes

Si ce travail a pu être réalisé dans des délais aussi brefs, ceci est en grande partie dû à l'efficacité de nos annotateurs. Le tableau 8.13 récapitule les résultats des deux campagnes.

Échantillon(s)	Mode d'annot.	Performances de l'outil LAS	UAS	Annot. retenues	Vitesse d'annot.man.
Échantillon initial	Manuelle	-	-	-	220 tok/h
3_20 et 4_20	Préannotation avec modèle 1 (40 K tok)	76,34	84,06	59 %	310 tok/h (+40 %)
5_20	Préannotation avec modèle 2 (80 K tok)	84,20	89,70	72 %	380 tok/h (+72 %)

TABLE 8.12 – Performances de Talismane et vitesse d'annotation manuelle

Camp.	Tâche	Annotateurs	Durée	Endroit	Rendement	Pers.-heures
C1	morphosyntaxe	3 L3 + 1 M1	2 semaines	Toulouse	30 000 tok.	60 h
			6 semaines	Belgrade	30 000 tok.	50 h
	lemmatisation	3 L3 + 1 M1	2 semaines	Toulouse	35 000 tok.	25 h
C2	syntaxe	1 L3 + 1 M1	3 semaines	Toulouse	40 000 tok.	150 h
			6 semaines	Belgrade	20 000 tok.	60 h

TABLE 8.13 – Travail réalisé dans les campagnes d'annotation manuelle

Nous pensons que ces résultats ont été favorisés par notre méthode globale : le fait de consacrer systématiquement un temps au retour d'expérience des annotateurs a été d'une grande importance. En effet, les annotateurs ont particulièrement apprécié la possibilité de faire remonter les problèmes rencontrés en annotation et de voir les guides d'annotation évoluer en fonction de leurs observations. Cette démarche leur a assuré une annotation plus aisée, mais elle leur a surtout permis de valoriser leur contribution, ce qui a renforcé leur sentiment d'appartenance au projet. Un autre point a contribué à ce sentiment : le fait de communiquer aux annotateurs les améliorations réalisées par les outils suite à la validation d'un échantillon. Cette forme de retour sur leur travail a toujours eu un accueil très positif.

8.8 Finalisation du corpus et améliorations possibles

Comme nous l'avons noté à la fin des sections 8.4, 8.5 et 8.6, ces deux campagnes d'annotation nous ont permis de valider l'annotation du corpus ParCoTrain-Synt aux trois niveaux d'annotation retenus. Nous avons donc passé à la dernière phase de ce projet : la

finalisation du corpus. Dans cette étape, nous avons, en tant qu'annotateur expérimenté, effectué l'harmonisation des annotations dans le corpus. Pour ce travail, nous nous sommes servie des versions les plus récentes des guides d'annotation. Durant les campagnes, nous avions établi un inventaire des changements qui ont eu lieu ; l'harmonisation a donc porté sur ces aspects-là.

Une fois ce travail finalisé, nous avons procédé à la préparation du corpus pour la diffusion. Le corpus est diffusé dans le format CoNLL-X. Il s'agit d'un standard *de facto* pour le stockage des treebanks adaptés au parsing en dépendances. Une ligne correspond à un token, et les différentes colonnes contiennent les informations suivantes : 1) l'ID du token, 2) le token, 3) le lemme, 4) l'étiquette POS gros grain, 5) l'étiquette POS plus spécifique, 6) les traits morphosyntaxiques, 7) l'ID du gouverneur du token et 8) l'étiquette syntaxique du token. Notons que nous exploitons la colonne 5 pour indiquer les étiquettes morphosyntaxiques détaillées. Un extrait du corpus est montré dans la figure 8.9.

```
1       Sad     sad     Adv     Adv_gen_pos     _       2       DepVAdv
2       idemo   ići     V_main  V_main_pres_1_pl_-_-    n=pl|t=pres|r=1 0       Root
3       da      da      C_sub   C_sub   _       2       Sub
4       nešto   nešto   P       P_indef_-_-_-_acc       c=acc   5       ObjDir
5       pitamo  pitati  V_main  V_main_pres_1_pl_-_-    n=pl|t=pres|r=1 3       PredSub
6       onu     onaj    A       A_dem_acc_sg_f_-        c=acc|g=f|n=sg  7       DepNAdj
7       budalu  budala  N       N_com_acc_sg_f  c=acc|g=f|n=sg  5       ObjDir
8       od      od      Prep    Prep    _       7       DepNPrep
9       kmeta   kmet    N       N_com_gen_sg_m  c=gen|g=m|n=sg  8       ComplPrep
10      !       !       Z       Z       _       9       Ponct
```

FIGURE 8.9 – Extrait du corpus ParCoTrain-Synt

Suivant une pratique courante dans le TAL, nous l'avons divisé en trois sections : *train* (destinée à l'entraînement des parsers), *dev* (dédiée au paramétrage fin) et *test* (réservée à l'évaluation). Afin d'éviter le biais quant à la longueur des phrases dans différents segments du corpus, les phrases ont été réordonnées de manière aléatoire avant la segmentation du corpus en sections. Quelques informations sur les propriétés du corpus annoté sont données dans le tableau 8.14.

On constate que les étiquettes syntaxiques et les étiquettes des parties du discours sont bien représentées dans toutes les sections, mais c'est moins vrai pour les étiquettes morphosyntaxiques détaillées : sur les 1042 étiquettes détaillées possibles définies par notre jeu d'étiquettes (cf. section 5.1.1), 679 sont instanciées dans le corpus complet, et ce taux est plus bas dans les sections individuelles. Ce fait pourrait se montrer problématique lors de l'utilisation du corpus pour l'entraînement des étiqueteurs morphosyntaxiques sur cette couche d'annotation.

Nous donnons également des indications de la complexité intrinsèque du texte au niveaux syntaxique à travers les mesures de longueur moyenne de la phrase et de profondeur maximale moyenne d'arbre syntaxique. La profondeur maximale est calculée comme la distance en nœuds entre le token le plus profond dans l'arbre et la racine de l'arbre. Ces

deux mesures sont stables à travers les sections du corpus. On peut donc considérer que les différentes sections sont d'une difficulté intrinsèque proche.

Section	Tokens	Phrases	Formes fléchies	Lemmes	Étiq. POS	Étiq. détail.	Traits MS	Étiq. synt.	$\text{Long}_{(\text{ph})}$	$\text{Prof}_{(\text{a})}$
all	101 029	3861	22 739	11 251	16	679	165	67	27,16	6,98
train	80 869	3116	19 598	10 120	16	643	159	67	26,95	6,98
test	10 162	367	4033	2802	15	432	134	60	28,69	7,11
dev	9 998	379	3959	2722	16	424	126	58	27,38	6,88

TABLE 8.14 – ParCoTrain-Synt : statistiques de base. $\text{Long}_{(\text{ph})}$ = longueur moyenne de phrase en tokens ; $\text{Prof}_{(\text{a})}$ = valeur moyenne de la profondeur d'arbre maximale

Le corpus est téléchargeable à partir de l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/ParCoTrain-Synt`.

Dans l'avenir, nous souhaitons examiner plusieurs pistes d'amélioration du corpus. Tout d'abord, nous reprendrons certains points de l'annotation syntaxique. Il s'agit en premier lieu des phénomènes volontairement mis de côté au moment de la création du schéma d'annotation (cf. le traitement des datifs, le statut des réflexifs, l'annotation détaillée des dépendants verbaux, nominaux et adjectivaux), mais aussi de ceux qui se sont montrés comme non optimaux *a posteriori*, comme l'ellipse. Des analyses plus poussées basées sur le corpus existant peuvent permettre d'identifier une meilleure manière d'encoder ces phénomènes en corpus.

Nous souhaitons également ré-examiner le traitement des unités polylexicales. Pour le moment, elles ne sont pas regroupées au niveau de la tokénisation, bien que certaines d'entre elles (notamment les locutions grammaticales) sont traitées au niveau syntaxique. Nous envisageons donc de traiter ces expressions comme des tokens uniques et d'évaluer l'effet de cette modification sur le parsing. Ce travail pourrait être basé sur les listes des locutions grammaticales fournies dans (Mrazović, 2009).

Étant donné l'utilité du schéma UD pour différentes applications en TAL (parsing multilingue, comparaisons des parsers, etc.), nous envisageons également une conversion de ParCoTrain-Synt vers ce formalisme. Nous précisons cependant que cette nouvelle annotation ne prendra pas la place de l'annotation existante, mais sera ajoutée comme une couche d'information supplémentaire.

La première piste d'amélioration à être poursuivie concerne le caractère mono-genre du corpus. Afin de combler cette lacune, un projet d'extension et de diversification de ParCoTrain-Synt a été entamé : l'une des annotatrices qui ont travaillé sur l'annotation manuelle a entrepris la constitution d'un échantillon journalistique de 30 000 tokens qui viendra enrichir le corpus existant. Ce travail est effectué dans le cadre de son mémoire de Master 2 à l'Université de Belgrade, que nous co-encadrons avec D. Stosic et V. Stanojevic.

Bien que ces différents aspects du corpus puissent être réexaminés et optimisés, ParCoTrain-Synt s'est déjà montré utile dans plusieurs applications différentes. Dans le chapitre suivant, nous optimisons l'exploitation des traits morphosyntaxiques en parsing à partir de ParCoTrain-Synt, alors que dans la partie III, nous illustrons l'utilité globale du corpus en TAL, mais surtout en linguistique théorique.

8.9 Bilan intermédiaire : retour d'expérience sur les campagnes et sur la méthode adoptée

Le fait même que le corpus ParCoTrain-Synt a pu être constitué dans les délais impartis démontre la validité de la méthode adoptée. Nous avons notamment montré qu'une telle démarche n'exige pas une équipe étendue, surtout en ce qui concerne le nombre de personnes ayant des responsabilités de gestion et d'expertise scientifique : nous avons assuré nous-mêmes les rôles du gestionnaire des campagnes, de l'annotateur expérimenté, et du taliste. Tout de même, il nous semble que le processus peut être plus ergonome si ces rôles-là sont distribués sur plusieurs personnes.

Nous avons également constaté que même une méthode réfléchie en détail doit se conformer finalement aux contraintes pratiques. Dans notre cas, la durée limitée du séjour des annotateurs à Toulouse, et surtout les compétences des annotateurs recrutés, nous ont obligée à abandonner l'idée d'effectuer toutes les couches d'annotation d'un échantillon à la fois : l'étiquetage et la lemmatisation ont été faits lors de la première campagne, alors que le parsing a été réalisé dans la deuxième.

Certains aspects de notre méthode méritent également d'être améliorés. Il s'agit notamment du temps consacré à la création d'une première version des guides d'annotation et à la formation des annotateurs. En effet, en constituant les guides d'annotation, nous avons investi un effort important dans une étude théorique des phénomènes problématiques potentiels. Or, une fois les guides confrontés aux données, certains d'entre eux n'ont pas été confirmés par l'usage, alors que de nouvelles lacunes ont émergé. Cette expérience indique qu'une démarche plus optimale consisterait à mettre en place un squelette pour les guides et entamer rapidement leur évaluation sur les données. Dans cet aspect, nous rejoignons les recommandations de Fort (2012).

En ce qui concerne la formation des annotateurs, dans le cadre de cette thèse, cette étape a été contrainte par la durée limitée des campagnes d'annotation. Par conséquent, la formation des annotateurs a porté seulement sur la maîtrise des guides d'annotation, alors que l'acquisition des tâches d'annotation en elles-mêmes s'est faite durant les campagnes. Or, Marcus et al. (1993) montrent que les effets de l'apprentissage sur la vitesse d'annotation peuvent persister jusqu'à un mois de travail. Il est donc essentiel de dégager au moins une partie du temps nécessaire pour cet apprentissage avant le début des cam-

pagnes proprement dites. Pour déterminer la durée minimale de cette étape de rodage, nous envisageons des expériences d'annotation plus longues, qui permettront de tracer la courbe d'apprentissage des annotateurs d'une manière plus suivie.

Quant à la mise en pratique de l'aspect outillé de notre méthode, elle a été réalisée à travers différents procédés. Nous les résumons ici.

Exploitation des ressources existantes pour une langue proche. Nous avons exploité avec succès un modèle d'étiquetage entraîné sur le croate pour faciliter la première phase d'étiquetage morphosyntaxique manuel de notre corpus (cf. section 7.2.1). Malheureusement, la même démarche n'a pas été fructueuse pour la lemmatisation : le modèle croate de CST avait été entraîné sur des données étiquetées avec un jeu d'étiquettes morphosyntaxiques différent du nôtre ; par conséquent, ses performances sur nos données ont été compromises (cf. section 7.3.1). Quant au modèle du parsing, au-delà du fait d'avoir été entraîné sur un jeu d'étiquettes morphosyntaxiques différent, il intégrait également un schéma d'annotation syntaxique très élgoiné du nôtre. Nous avons donc jugé qu'une préannotation avec ce modèle n'était pas la manière la plus économe de procéder et avons favorisé une annotation manuelle.

Grâce à la relation particulière entre le croate et le serbe décrite dans la section 1.1.6, nous avons pu transposer le modèle d'étiquetage directement sur notre corpus, sans devoir faire appel à des techniques de traitement inter-langues. Cependant, une approche comparable peut également être envisagée pour des langues plus éloignées. À titre d'illustration, le travail de Vergez-Couret & Urieli (2015) montre que l'étiquetage de l'occitan bénéficie de l'ajout d'un grand corpus catalan à un corpus minimal de l'occitan lors de l'apprentissage.

Exploitation de ressources d'entraînement minimales. Pour assurer un entraînement initial du lemmatiseur CST, nous avons exploité la lemmatisation présente dans le corpus au démarrage des campagnes d'annotation, réalisée dans le cadre de nos travaux antérieurs (Miletic, 2013). Ces données nous ont permis de constituer un lexique d'entraînement d'à peine 10 000 lemmes. Vu la richesse morphologique du serbe, un entraînement sur une ressource aussi petite n'était pas prometteur. Néanmoins, malgré les performances moyennes de l'outil (86,2 % d'exactitude), la vitesse d'annotation manuelle a quasiment doublé (cf. section 7.3.1).

Exploitation de ressources collaboratives *open source* : Wiktionary. Nous avons exploité ce dictionnaire électronique pour en extraire un premier lexique morphosyntaxique ; nous avons ensuite combiné le résultat de ce travail avec un autre lexique serbe (Ljubešić et al., 2016) (cf. chapitre 6). Cette démarche s'est avérée particulièrement bénéfique pour la lemmatisation : suite à un entraînement sur cette ressource, les résultats du lemmatiseur CST étaient suffisamment élevés pour que l'on se passe des réentraînements itératifs stipulés par notre méthode (cf. section 7.3.2).

Exploitation des fonctionnalités auxiliaires des outils automatiques. Nous faisons ici notamment référence à la capacité de Talismane d'accompagner les étiquettes syntaxiques produites par les taux de probabilité associés. Ce fait nous a permis de trier sa sortie et de retenir une préannotation incomplète, mais fiable (cf. section 8.6.2). Cette démarche est notamment importante lors d'une préannotation avec un outil relativement peu performant au niveau global, autrement dit, durant les premières itérations de la démarche.

Bien qu'utilisés ici dans un contexte spécifique, ces procédés peuvent être transposés à d'autres langues. Ceci est également le cas de l'annotation agile basée sur le *bootstrapping* itératif que nous avons proposée. En effet, le caractère général de notre méthode sera évalué dans un avenir immédiat : elle sera appliquée sur l'occitan dans le but de constituer un treebank pour cette langue. Ce travail sera effectué dans le cadre du projet ParCoLaF, qui vise l'ouverture du corpus ParCoLab aux langues de France, en commençant par l'occitan. Cette initiative est soutenue par un projet DGLFLF dans le cadre de l'APN « Langues et numérique » 2017.

Chapitre 9

Parsing du serbe : définition des conditions d'apprentissage optimales

Ce chapitre présente les expériences que nous avons menées sur le parsing du serbe à partir du corpus ParCoTrain-Synt. Notre objectif est ici d'évaluer l'utilité globale du corpus dans le cadre du parsing, mais aussi d'identifier les conditions d'apprentissage les mieux adaptées au serbe afin de proposer un modèle de parsing optimisé. Tout d'abord, nous évaluons l'impact sur les performances du parser des informations morphosyntaxiques. Ceci est fait de deux façons : en variant la taille et la structure du jeu d'étiquettes morphosyntaxiques et en exploitant les traits morphosyntaxiques fins en tant que traits d'apprentissage individuels passés au parser. Nous explorons également l'interaction de ces méthodes avec la granularité du jeu d'étiquettes syntaxiques. Dans un deuxième temps, nous déterminons les valeurs optimales de différents paramètres d'apprentissage statistiques. Ces optimisations nous permettent d'améliorer les performances du modèle initial de 4,2 points en LAS et de 1,8 points en UAS. Le modèle optimisé est librement disponible à l'adresse `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Models/Parsing/Talismane`.

9.1 Exploitation des traits morphosyntaxiques fins dans le parsing des langues à morphologie flexionnelle riche

Comme il a déjà été mentionné dans la section 1.3, le marquage des fonctions syntaxiques dans les langues à morphologie flexionnelle riche et à ordre des mots flexible implique souvent un marquage formel précis de différentes catégories grammaticales, tels le cas, le nombre ou le genre. De nombreux travaux examinent la manière optimale d'exploiter ce type d'informations morphosyntaxiques dans le cadre du parsing. On peut distinguer deux méthodes principales : les traits morphosyntaxiques fins peuvent être intégrées aux

étiquettes morphosyntaxiques, ou bien elles peuvent être fournies en tant que traits d'apprentissage (angl. *features*) à l'algorithme. Dans le premier cas de figure, l'inclusion des traits fins dans les étiquettes morphosyntaxiques augmente la taille du jeu d'étiquettes morphosyntaxiques et, par conséquent, le nombre de classes à partir desquelles le parser effectue l'apprentissage. Étant donné la taille souvent limitée des corpus d'entraînement, cela peut rendre le parsing plus difficile. La deuxième approche permet d'éviter ce problème en gardant le jeu d'étiquettes morphosyntaxiques initial et en déléguant l'usage des traits morphosyntaxiques à un autre niveau d'annotation. Nous détaillons les deux approches dans la suite.

9.1.1 Inclusion des traits morphosyntaxiques fins dans les étiquettes catégorielles

Collins et al. (1999) évaluent le parser de Collins (1997) sur le corpus tchèque PDT. Vu la granularité importante du jeu d'étiquettes morphosyntaxiques original de ce corpus (au-delà de 1000 étiquettes attestées en corpus), les auteurs réduisent le jeu à la seule indication de la partie du discours. Ceci donne un jeu morphosyntaxique de 13 étiquettes. Afin de récupérer une partie des informations morphosyntaxiques ainsi éliminées, les auteurs ajoutent l'indication du cas aux étiquettes des classes qui exhibent ce trait morphosyntaxique. Cela aboutit à un jeu augmenté de 58 étiquettes. L'utilisation de ce jeu plus informatif apporte une amélioration de l'exactitude du parsing de 1,1 % par rapport au jeu morphosyntaxique minimal.

Des effets comparables ont été observés sur le français : Seddah et al. (2009) montrent que des jeux plus détaillés favorisent aussi bien le parsing en dépendances qu'en constituants. En revanche, ici les informations morphosyntaxiques supplémentaires relèvent plutôt des sous-catégories distributionnelles des parties du discours (par exemple, pronom *vs* pronom interrogatif) que des traits flexionnels.

La taille des jeux détaillés dans les travaux cités ci-dessus reste cependant relativement restreinte. Des augmentations plus importantes peuvent se montrer contre-productives, notamment lorsque les informations morphosyntaxiques sont fournies par un étiqueteur automatique. Dans leur travail sur l'arabe standard moderne, Marton et al. (2013) évaluent l'apport de différents jeux d'étiquettes morphosyntaxiques, contenant entre 6 et 430 étiquettes. Si le parsing est fait à partir de l'annotation morphosyntaxique *gold*, le jeu d'étiquettes le plus détaillé donne les meilleurs résultats en parsing. En revanche, si l'étiquetage morphosyntaxique est fourni de manière automatique, le même jeu d'étiquettes est le moins favorable pour le parsing. Dans ce deuxième scénario, le meilleur score LAS est réalisé avec un jeu morphosyntaxique de 44 étiquettes, qui obtient également le meilleur score en étiquetage morphosyntaxique (97,7 % d'exactitude).

Des observations comparables ont été faites par Maier et al. (2014) sur l'allemand. Ces auteurs évaluent l'apport de 3 jeux d'étiquettes morphosyntaxiques différents, contenant respectivement 12, 54 et plus de 500 étiquettes. Le parsing est évalué sur 2 corpus selon 3 modalités : avec l'annotation morphosyntaxique *gold*, avec l'annotation morphosyntaxique fournie par le parser choisi (Petrov et al., 2006), et avec l'annotation morphosyntaxique fournie par TnT (Brants, 2000b), qui s'était avéré l'étiqueteur le plus stable lors d'une évaluation de 6 outils différents. Indépendamment du corpus et du scénario d'évaluation, le tagset de 54 étiquettes a mené aux meilleurs résultats en parsing. En revanche, le tagset le plus détaillé s'est systématiquement montré le moins adapté, même dans le scénario avec les annotations morphosyntaxiques *gold*. À partir de ces observations, les auteurs concluent que trop d'informations morphosyntaxiques défavorisent les performances des parsers sur l'allemand. Il faut noter cependant que dans le cadre de ce travail les traits morphosyntaxiques fins sont fournis au parser en tant que blocs d'informations collés aux étiquettes des parties du discours. Comme mentionné ci-dessus, cela donne un jeu morphosyntaxique de plusieurs centaines d'étiquettes. Les résultats obtenus indiquent effectivement que ce niveau de granularité affecte négativement les résultats du parsing. Cependant, si ces données avaient été exploitées en tant que traits atomiques, indépendamment du jeu d'étiquettes morphosyntaxiques, il est possible qu'elles aient eu un effet positif sur le traitement syntaxique.

9.1.2 Traits morphosyntaxiques fins en tant que traits d'apprentissage automatique

Cette deuxième méthode d'utilisation des informations morphosyntaxiques est explorée en détail par Marton et al. (2013) sur l'arabe standard moderne. Les auteurs évaluent l'apport de 9 traits morphosyntaxiques différents [1] pour le parsing selon plusieurs modalités : (modalité 1) en utilisant tous les traits disponibles ensemble ; (modalité 2) en utilisant chaque trait individuellement ; (modalité 3) en créant des combinaisons des traits par un ajout progressif des traits les plus utiles de la modalité 2 tout en gardant seulement ceux qui apportaient une amélioration significative aux résultats. Les trois modalités sont utilisées selon deux scénarios : avec des annotations morphosyntaxiques *gold* et avec des annotations automatiques, et les résultats sont comparés avec une *baseline* établie avec les indications des parties du discours seules.

Avec les annotations *gold*, la modalité 1 permet d'améliorer les résultats par rapport à la *baseline*. Au niveau des traits individuels, c'est le trait du cas qui apporte le gain le plus important, suivi par l'état et la présence du déterminant. Dans la modalité 3, la combinaison la plus efficace est celle du cas et de l'état. L'ajout de la présence du

1. Présence du déterminant, état (s'applique aux formes nominales ; valeurs possibles : défini, indéfini ou tête de la construction *idafa*), cas, personne, nombre, genre, aspect verbal, voix verbale, mode verbal.

déterminant n'apporte pas de gain, ce qui montre une redondance par rapport aux deux premiers traits.

En revanche, avec une annotation morphosyntaxique automatique, les résultats étaient bien différents : la modalité 1 apportait une perte par rapport à la *baseline*, le cas était le trait individuel le moins utile, et la meilleure combinaison des traits selon la modalité 3 comprenait la présence du déterminant, la personne, le nombre et le genre. Les auteurs expliquent ces différences par les notions de pertinence et d'exactitude de différents traits : les résultats du scénario *gold* montrent que le trait du cas est pertinent pour l'analyse syntaxique de l'arabe ; en revanche, son taux d'exactitude dans le cadre d'une annotation automatique n'est pas suffisamment élevé pour le rendre exploitable dans le deuxième scénario. L'utilité d'un trait pour le parsing dépend donc à la fois de sa pertinence et de son exactitude, mais aussi de sa redondance : un trait n'est pas utile s'il fournit le même type d'information qu'un autre trait. Toutefois, la démarche d'ajout progressif de différents traits a permis aux auteurs d'identifier des ensembles de traits qui restent utiles même s'ils sont fournis de manière automatique. La combinaison la plus performante permet d'améliorer les résultats de manière statistiquement significative par rapport à la *baseline* (+1,4 % pour le score LAS).

L'utilité des traits morphosyntaxiques atomiques a également été démontrée dans le cadre du parsing en constituants : Szántó & Farkas (2014) montrent que l'inclusion de ce type d'information mène à une amélioration des résultats sur le basque, le français, l'hébreu et le hongrois. Les auteurs attribuent cet effet au fait que ces données permettent de capter les phénomènes d'accord grammatical d'une manière plus explicite.

9.1.3 Méthode adoptée

Compte tenu des résultats mis en avant par les travaux présentés ci-dessus, nous souhaitons évaluer les deux modes d'exploitation d'informations morphosyntaxiques sur le serbe. Dans nos expériences de parsing initiales, nous avons entraîné le parser Talismane sur un jeu d'étiquettes morphosyntaxiques de 15 étiquettes [2], le jeu d'étiquettes syntaxiques plein (67 fonctions syntaxiques), et un ensemble de 6 traits morphosyntaxiques : le cas, le nombre, le genre, la personne, la forme verbale, et la morphologie complète, ce dernier étant une concaténation des cinq premiers. Comme le montrent les travaux présentés ci-dessus, manipuler ces éléments peut avoir un effet sur le processus d'apprentissage et modifier ainsi les résultats du parsing. Nous adoptons donc l'approche suivante : dans un premier temps, nous examinons la variation de granularité du jeu d'étiquettes morphosyntaxiques. Étant donné la corrélation forte entre les annotations morphosyntaxique et syntaxique

2. Il s'agit des catégories suivantes : adjectif, adverbe, nom, numéral, pronom, particule, préposition, verbe principal, verbe auxiliaire, conjonction de coordination, conjonction de subordination, interjection, abréviation, mot étranger, ponctuation.

dans notre corpus (cf. section 5.2.7), nous évaluons également plusieurs variantes du jeu syntaxique. Dans un deuxième temps, nous accordons une attention particulière à l'évaluation de l'apport des informations morphosyntaxiques fines utilisées en tant que traits d'apprentissage. En effet, Agić et al. (2013b) ont exploré cette question sur le croate, mais dans une perspective inverse : à partir d'un jeu d'étiquettes morphosyntaxiques détaillé, ils éliminaient des traits individuels afin d'identifier ceux qui nuisaient au parsing. Notre objectif est en revanche d'établir quels traits sont les plus utiles et qui devraient, par conséquent, être une priorité dans l'élaboration des corpus d'entraînement pour le serbe. Par ailleurs, dans ce volet de l'étude, nous exploitons les données morphosyntaxiques sous forme de traits atomiques, et non pas comme des éléments du jeu d'étiquettes morphosyntaxiques, ce qui était le cas dans (Agić et al., 2013b). Dans un dernier temps, nous identifions les paramètres d'apprentissage automatique qui permettent d'atteindre les résultats les plus élevés et analysons brièvement les performances du modèle optimisé.

La suite du chapitre s'organise donc de la manière suivante : dans la section 9.2, nous proposons une description rapide du corpus, du parser et du lexique utilisés dans nos expériences ; la section 9.3 est dédiée aux expériences sur la granularité des jeux d'étiquettes morphosyntaxiques et syntaxique ; l'exploitation des informations morphosyntaxiques en tant que traits d'apprentissage est détaillée dans la section 9.4 ; dans la section 9.5, nous présentons les expériences sur les paramètres d'apprentissage automatique et les résultats du modèle final ; enfin, la section 9.6 propose des conclusions et des pistes pour la suite.

9.2 Ressources et outils utilisés

Ici nous présentons les outils et ressources utilisés dans l'ensemble des expériences qui font l'objet de ce chapitre. Il s'agit notamment du corpus d'entraînement et d'évaluation ParCoTrain-Synt, du parser Talismane et du lexique ParCoLex. Comme ces trois éléments ont déjà été décrits en détail dans des sections dédiées (respectivement, 8.8, 7.4 et 6.3), nous nous limitons ici à quelques précisions et à un rappel rapide de quelques-unes de leurs propriétés, essentielles pour la présente étude.

9.2.1 Corpus de travail : 81 000 tokens de ParCoTrain-Synt

Les expériences décrites dans ce chapitre ont été entamées avant la fin de l'annotation manuelle du corpus ParCoTrain-Synt. En effet, les premières d'entre elles ont été effectuées sur les 4 premiers échantillons validés du corpus (environ 81 000 tokens). Afin de préserver la comparabilité des résultats, nous nous sommes servie du même échantillon en tant que corpus *gold* par la suite. Pour les besoins de nos expériences, le corpus *gold* a été réparti en sections *train* (dédiée à l'entraînement), *dev* (destinée au paramétrage fin du parsing) et *test* (destinée à l'évaluation finale).

Des informations de base sur le corpus *gold* dans sa totalité ainsi que sur les différentes sections sont données dans le tableau 9.1.

Section	Tokens	Phrases	Formes fléchies	Lemmes	Étiq. POS	Étiq. détail.	Traits MS	Étiq. synt.	$\text{Long}_{(\text{ph})}$	$\text{Prof}_{(\text{a})}$
all	81 204	2949	19 680	10 232	15	646	116	67	27,53	7,23
train	69 350	2544	17 826	9556	15	623	104	67	27,26	7,17
test	7803	298	3026	2095	14	387	88	56	26,19	7,01
dev	4051	107	1693	1245	14	314	91	56	37,86	9,12

TABLE 9.1 – Données sur le corpus d'évaluation. $\text{Long}_{(\text{ph})}$ = longueur moyenne de phrase en tokens ; $\text{Prof}_{(\text{a})}$ = valeur moyenne de la profondeur d'arbre maximale

Comme indiqué dans la section 8.8, l'annotation morphosyntaxique dans la version finale du corpus est réalisée à trois niveaux : le premier contient les seules indications des parties du discours (étiquettes POS), le deuxième comprend les étiquettes détaillées, et le troisième les traits morphosyntaxiques individuels. Sur les 1042 étiquettes morphosyntaxiques détaillées possibles selon notre schéma d'annotation, 646 sont attestées dans notre corpus *gold*. La section *train* a une bonne couverture de ces étiquettes (623 étiquettes détaillées), mais c'est moins le cas des sections *test* et *dev* (respectivement 387 et 314 étiquettes détaillées). Il est également clair que le rapport tag-token est faible dans toutes les sections. La situation est meilleure quant aux combinaisons des traits morphosyntaxiques : sur 116 combinaisons de la totalité du corpus, 104 sont présentes dans le *train*, 88 dans le *test* et 91 dans le *dev*. Dans les expériences décrites ci-dessous, nous utilisons systématiquement les étiquettes POS en tant que classes d'apprentissage pour le parser, alors que les informations morphosyntaxiques sont exploitées en tant que traits. Cependant, vu les particularités du parser Talismane, les traits sont dérivés du lexique ParCoLex et ne sont pas puisés dans le corpus (cf. section 9.2.2).

Quant à l'annotation syntaxique, on trouve dans le corpus *gold* 50 étiquettes principales et 17 étiquettes pour l'ellipse (cf. section 5.2.10 pour une explication détaillée du traitement de l'ellipse). Elles sont bien représentées dans les différentes sections (les 67 sont présentes dans *train*, et 56 dans *test* et *dev*).

En ce qui concerne la longueur de la phrase et la profondeur maximale de l'arbre syntaxique, les deux mesures sont relativement proches entre la totalité du corpus *gold* et les sections *train* et *test*. En revanche, elles sont plus élevées pour le *dev*, ce qui pourrait indiquer que cette section est intrinsèquement plus difficile à parser que les autres.

Précisons que les résultats présentés dans la suite de ce chapitre sont obtenus sur la section *dev* sauf indication contraire. Cela est dû au fait que la majorité de nos expériences relèvent des procédés d'optimisation. En revanche, une fois la configuration optimale identifiée, une évaluation globale sur la section *test* est effectuée.

9.2.2 Outil : Talismane, un parser à base de transitions

Étant donné les résultats initiaux encourageants obtenus avec le parser Talismane (Urieli, 2013) dans le cadre de la constitution de ParCoTrain-Synt (cf. section 8.6.2), nous poursuivons nos expériences avec cet outil. Une description détaillée de Talismane est disponible dans la section 7.4 ; nous rappelons ici les caractéristiques utiles à cette partie de l'étude.

Tout d'abord, Talismane intègre une chaîne de traitement complète, capable d'effectuer également la tokénisation et l'étiquetage morphosyntaxique. Nous nous en sommes servie ici pour comparer les résultats du parsing basés sur des annotations morphosyntaxiques manuelles avec ceux basés sur un étiquetage morphosyntaxique automatique.

Nous avons également exploité la fonctionnalité de Talismane pour la définition des traits d'apprentissage fournis au parser. Ceci est fait à l'aide d'un fichier dédié qui fait partie de l'ensemble des fichiers nécessaires au parsing (*languagePack*). La syntaxe mise en place permet d'indiquer quels sont les traits à utiliser pour différentes positions dans la pile et dans le *buffer*[3], mais aussi dans l'ordre linéaire de la phrase, ainsi que dans l'arbre en construction.

Dans nos expériences, nous limitons l'utilisation des traits morphosyntaxiques fins aux tokens en haut de la pile et en tête du *buffer*. Afin de modéliser explicitement l'accord, nous combinons les valeurs du même trait pour les deux positions (par exemple, le genre du token en haut de la pile et le genre de celui en tête du *buffer*). Quant aux positions dans la structure de l'arbre en construction, nous exploitons la forme verbale et le cas pour la tête, les dépendants gauches et les dépendants droits du token en haut de la pile, ainsi que pour les dépendants gauche du token en tête du *buffer*.

9.2.3 Lexique : ParCoLex

Talismane s'appuie seulement sur le lexique pour les traits morphosyntaxiques, préservant toute l'ambiguïté rencontrée. Ses performances sont donc affectées par la qualité des ressources lexicales à sa disposition. Dans les expériences décrites ici, nous utilisons un grand lexique généraliste et un petit lexique de classes fermées complémentaire.

Le lexique généraliste est ParCoLex, présenté en détail dans la section 6.3. Nous rappelons ici la composition du lexique combiné ParCoLex, et indiquons par ailleurs le contenu du lexique des classes fermées, ParCoLex-closed (cf. tableau 9.2).

ParCoLex-closed contient 1143 entrées de 702 formes fléchies uniques provenant de 350 lemmes différents. Il a été compilé à partir du corpus ParCoTrain-Synt, et les lacunes les plus évidentes ont été complétées manuellement (par exemple, les paradigmes des verbes

3. Voir la section 3.4.2 pour la définition de ces termes.

Lexique	Entrées	Formes fléchies	Lemmes
srLex	5 327 361	1 436 966	105 358
Wikimorph-sr	3 066 214	1 226 638	117 445
ParCoLex	6 767 039	1 959 212	157 887
ParCoLex-closed	1143	702	350

TABLE 9.2 – Ressources lexicales utilisées

auxiliaires). Ce lexique est utilisé seulement si aucune entrée n'est trouvée dans ParCoLex pour une forme fléchie donnée.

No d'entrées	% des formes fléchies
0	8,2%
1	32,8%
2-5	50,9%
6-10	5,4%
>10	2,7%

TABLE 9.3 – Couverture du corpus *gold* par les lexiques

La combinaison de ces deux lexiques a une bonne couverture sur notre corpus *gold* : seulement 8,2 % des formes fléchies uniques trouvées en corpus ne sont pas présentes dans ces ressources (cf. tableau 9.3). Seules 32,8 % de formes fléchies uniques du corpus ont une entrée unique dans le lexique. Le tableau 9.3 montre également que plus de 50 % des formes fléchies ont entre 2 et 5 descriptions morphosyntaxiques possibles. Cela souligne encore l'importance d'utiliser un système robuste face à l'ambiguïté morphosyntaxique dans le traitement du serbe.

9.3 Variations de granularité des jeux d'étiquettes

Nos premières expériences orientées vers l'optimisation du parsing ont porté sur la structure et la taille des jeux d'étiquettes morphosyntaxiques et syntaxiques. Dans un premier temps, nous avons suivi une démarche proche de celle de (Collins et al., 1999) : nous avons enrichi les indications des parties du discours par l'information du cas. Ensuite, nous avons testé plusieurs réductions du jeu d'étiquettes syntaxiques concentrées sur les étiquettes sous-spécifiées en Dep (cf. section 5.2.7) afin d'examiner le lien entre les annotations morphosyntaxique et syntaxique dans notre corpus. La première expérience est décrite dans la section 9.3.1, et la deuxième dans la section 9.3.2.

9.3.1 Ajout du cas aux étiquettes morphosyntaxiques

En serbe, quatre parties du discours portent les marques du cas : le nom, le pronom, l'adjectif, et certains numéraux. Pour cette première expérience, dans le corpus d'entraînement, nous avons remplacé les étiquettes POS de base par des étiquettes complexes contenant la partie du discours et un trait indiquant le cas. La valeur du suffixe a été extraite de l'annotation morphosyntaxique manuelle. Comme il existe 7 cas différents en serbe (nominatif, génitif, datif, accusatif, vocatif, instrumental et locatif), chacune des 4 étiquettes de base a été remplacée par 7 étiquettes complexes. Autrement dit, l'étiquette N a été remplacée par N_nom, N_gen, N_dat, etc. Le jeu d'étiquettes ainsi dérivé contient 44 étiquettes au total, contre 15 étiquettes dans le jeu de départ.

Nous avons déjà évoqué le fait qu'une telle intervention peut doublement augmenter la difficulté du parsing : premièrement, elle augmente le nombre des classes d'apprentissage sur lesquelles le parser se base ; deuxièmement, dans le cadre d'un parsing fait à partir d'une annotation morphosyntaxique automatique, cette modification rend également l'étiquetage plus difficile. Comme les erreurs introduites à ce niveau d'annotation ont tendance à se propager, le parsing en est probablement d'autant plus perturbé dans ce scénario d'évaluation. Pour essayer de mesurer ces différents effets, nous évaluons Talismane sur les deux jeux d'étiquettes dans deux scénarios de test : à partir des annotations morphosyntaxiques *gold vs* automatiques. Les autres conditions d'apprentissage restent inchangées : nous utilisons le jeu d'étiquettes syntaxiques plein (67 étiquettes) et fournissons les 6 traits morphosyntaxiques en tant que traits d'apprentissage au parser [4]. Pour l'étiquetage automatique, nous utilisons le module dédié de Talismane entraîné sur la section *train* du corpus *gold*. Les résultats sont donnés dans le tableau 9.4 : nous indiquons les scores LAS et UAS par jeu d'étiquettes et par scénario. Étant donné l'importance de la qualité de l'étiquetage morphosyntaxique pour le deuxième scénario, nous donnons également l'exactitude globale pour cette composante d'analyse (cf. colonne *Étiquetage*).

Jeu d'étiq.	POS *gold*		POS automatiques		
morphosynt.	LAS	UAS	LAS	UAS	Étiquetage
De base	87,51	90,89	**79,49**	**83,39**	**94,52**
+cas	**88,69**	**91,21**	69,59	75,27	86,05

TABLE 9.4 – Résultats du parsing avec les jeux d'étiquettes morphosyntaxiques de base et augmenté, sur les étiquettes *gold* et annotées automatiquement. *Étiquetage* = exactitude de l'étiquetage morphosyntaxique dans le scénario basé sur un étiquetage automatique.

Comme on peut voir, le jeu d'étiquettes augmenté est plus utile que le jeu de base sur l'étiquetage *gold*, mais il donne des résultats inférieurs si l'étiquetage est automatique.

4. Cas, genre, nombre, personne, forme verbale, morphologie complète.

Dans les deux cas, la différence est statistiquement très significative ($p < 0{,}005$ dans le test de McNemar). Ceci n'est pas surprenant : dans le premier scénario, le parser a à sa disposition l'information du cas désambiguïsée provenant de l'annotation manuelle, alors que dans le deuxième il doit la récupérer du lexique et gérer toute ambiguïté rencontrée. En revanche, ce jeu d'étiquettes à grain plus fin rend l'étiquetage automatique nettement plus difficile (cf. la colonne Étiquetage dans le tableau 9.4), ce qui a un effet néfaste sur le parsing.

Nous pouvons également constater que nos résultats sont en accord avec les observations faites dans les travaux existants. Comme pour le tchèque (Collins et al., 1999), l'ajout du cas aux étiquettes morphosyntaxiques facilite le parsing dans le scénario *gold*, mais le parsing basé sur un étiquetage automatique en pâtit, vu que cette modification a un impact négatif sur l'étiquetage morphosyntaxique. Dans les termes de Marton et al. (2013), le trait du cas s'est avéré pertinent pour le parsing du serbe, mais l'exactitude avec laquelle on est capable de le prédire de manière automatique est insuffisante pour le rendre utile dans un cadre supposant un traitement automatique au niveau morphosyntaxique. Comme c'est toutefois ce deuxième type d'utilisation qui est plus fréquent, nous poursuivons nos tests en utilisant le jeu d'étiquettes morphosyntaxiques de base.

Notons enfin une différence entre nos observations et celles d'Agić et al. (2013b). Dans notre cas, l'étiquetage morphosyntaxique automatique affecte fortement les performances quel que soit le jeu d'étiquettes : environ -8 points en LAS et -7 points en UAS avec l'étiquetage *gold*, et environ -19 points en LAS et -15 points en UAS avec l'étiquetage automatique. En revanche, Agić et al. (2013b) constatent des chutes beaucoup plus faibles en passant à un prétraitement automatique sur le croate : de -3 à -4 points en LAS et -2 points en UAS. Cette différence peut être due au fait que la définition de certaines étiquettes de notre jeu syntaxique s'appuie fortement sur les propriétés morphosyntaxiques des tokens. Cette observation nous a amenée à examiner le lien entre le jeu d'étiquettes syntaxiques et les performances du parsing.

9.3.2 Variation de la granularité du jeu d'étiquettes syntaxiques

Cette deuxième série de tests porte sur l'effet des modifications au niveau du jeu d'étiquettes syntaxiques. Plus particulièrement, nous avons examiné plusieurs réductions possibles focalisées sur les étiquettes sous-spécifiées en Dep. Pour rappel, ces étiquettes identifient un élément en tant que dépendant et donnent la nature morphosyntaxique du gouverneur et du dépendant selon le patron suivant : Dep(V|N|Adj|Adv)(Cas|Prep|Adj|Adv), où Cas désigne un nom dans un cas oblique. À titre d'illustration, un dépendant d'un verbe sous forme d'un groupe prépositionnel est annoté comme DepVPrep, alors qu'un dépendant d'un nom sous forme d'un nom dans un cas oblique porte l'étiquette DepN-

Cas. Ces étiquettes sont utilisées pour tous les dépendants des noms, des adjectifs et des adverbes, ainsi que pour les dépendants des verbes autres que les sujets, les objets et les prédicatifs (cf. section 5.2.7 pour une description détaillée). Il est évident que ces étiquettes reflètent seulement la nature morphosyntaxique du gouverneur et du dépendant. Le même type d'information étant disponible au niveau morphosyntaxique, il est possible de faire des réductions du jeu d'étiquettes syntaxiques sans entraîner une perte d'information. En revanche, une telle manipulation pourrait avoir un effet bénéfique pour le parsing en réduisant le nombre de classes d'apprentissage. Pour tester cette hypothèse, nous utilisons trois réductions progressives du jeu d'étiquettes syntaxiques, présentées dans le tableau 9.5.

<table>
<tr><th>Complet</th><th>Réduct.1</th><th>Réduct.2</th><th>Réduct.3</th></tr>
<tr><td>DepAdjAdj</td><td rowspan="2">DepAdjMod</td><td rowspan="4">DepAdj</td><td rowspan="20">Dep</td></tr>
<tr><td>DepAdjAdv</td></tr>
<tr><td>DepAdjCas</td><td rowspan="2">DepAdjFlect</td></tr>
<tr><td>DepAdjPrep</td></tr>
<tr><td>DepAdvAdv</td><td rowspan="2">DepAdvMod</td><td rowspan="3">DepAdv</td></tr>
<tr><td>DepAdvPrep</td></tr>
<tr><td>DepAdvCas</td><td>DepAdvCas</td></tr>
<tr><td>DepNAdj</td><td>DepNAdj</td><td rowspan="3">DepN</td></tr>
<tr><td>DepNCas</td><td rowspan="2">DepNFlect</td></tr>
<tr><td>DepNPrep</td></tr>
<tr><td>DepVAdv</td><td>DepVAdv</td><td rowspan="5">DepV</td></tr>
<tr><td>DepVCas</td><td rowspan="2">DepVFlect</td></tr>
<tr><td>DepVPrep</td></tr>
<tr><td>DepVInf</td><td>DepVInf</td></tr>
<tr><td>DepVPart</td><td>DepVPart</td></tr>
</table>

Table 9.5 – Réductions progressives du jeu d'étiquettes syntaxiques

La première réduction apporte quelques simplifications basées sur le comportement syntaxique des dépendants : les dépendants adjectivaux ou adverbiaux d'un adjectif sont le plus souvent des modifieurs de leur gouverneur et sont typiquement positionnés à sa gauche. Ils sont par conséquent groupés sous l'étiquette DepAdjMod. Les dépendants d'un nom, d'un adjectif ou d'un verbe sous forme d'un GP (Prep) ou d'un GN dans un cas oblique (Cas) peuvent avoir des fonctions syntaxiques différentes, mais ils partagent des règles de linéarisation par rapport à leur gouverneur ; ils sont donc réunis respectivement sous les étiquettes DepNFlect, DepAdjFlect et DepVFlect. La deuxième réduction garde seulement la distinction relative au gouverneur, alors que la troisième réduit toutes les étiquettes à une seule. Dans tous les cas, le reste du jeu d'étiquettes syntaxiques est inchangé.

Nous avons évalué Talismane sur les quatre versions du jeu d'étiquettes syntaxiques,

en utilisant le jeu d'étiquettes morphosyntaxiques de base, avec l'ensemble des 6 traits morphosyntaxiques disponibles. Les évaluations ont été faites selon deux scénarios : avec l'étiquetage morphosyntaxique manuel et avec celui fourni par Talismane. Les résultats sont donnés dans le tableau 9.6.

Jeu d'étiq. synt.	POS *gold*		POS automatique	
	LAS	UAS	LAS	UAS
De base	87,51	90,89	79,49	83,39
Réduction 1	87,58	90,92	79,46	83,26
Réduction 2	**87,95**	**91,41**	79,76	**83,56**
Réduction 3	87,56	90,97	**79,86**	83,24

TABLE 9.6 – Résultats de parsing sur 4 niveaux de granularité du jeu d'étiquettes syntaxiques

Comme on peut voir, les réductions du jeu d'étiquettes syntaxiques ont tendance à donner des résultats légèrement plus élevés, les gains étant plus marqués dans le scénario avec l'étiquetage morphosyntaxique automatique. Cela semble confirmer notre hypothèse que le lien entre les étiquettes morphosyntaxiques et syntaxiques s'avère problématique dans le cadre d'un étiquetage morphosyntaxique automatique. C'est probablement la raison pour laquelle la Réduction 3, qui fait appel à l'étiquette la plus générale, obtient le score LAS le plus élevé avec un étiquetage automatique. Il est néanmoins important de noter qu'aucune des différences entre les scores n'est statistiquement significative ($p > 0{,}1$ dans le test de McNemar).

Encore une fois, nous orientons notre choix par les résultats obtenus dans les expériences basées sur un étiquetage morphosyntaxique automatique, vu que ce cadre applicatif correspond à l'utilisation réelle du parser sur un texte brut. Nous retenons donc la Réduction 3 pour son score au niveau du LAS.

9.4 Apport des traits morphosyntaxiques

Afin d'évaluer l'utilité propre aux différents traits morphosyntaxiques utilisés et d'identifier la combinaison la plus intéressante pour le parsing, nous effectuons deux séries d'expériences. Dans la première, nous exploitons chacun de ces traits de manière individuelle (cf. section 9.4.1). Dans la deuxième, nous nous appuyons sur des combinaisons des traits formées par l'ajout progressif de traits individuels (cf. section 9.4.2). Dans les deux séries de tests, les évaluations sont faites aussi bien sur l'étiquetage morphosyntaxique *gold* que sur l'étiquetage automatique. Le jeu d'étiquettes morphosyntaxiques est le jeu de base (étiquettes catégorielles sans ajout de traits morphosyntaxiques), et le jeu d'étiquettes syntaxiques est celui de la Réduction 3 de la section précédente.

9.4.1 Évaluation de l'utilisation des traits individuels

Dans le cadre de cette expérience, nous considérons comme *baseline* les résultats de parsing obtenus sur les seules étiquettes POS, sans utilisation de traits morphosyntaxiques. Ensuite, chacun des 6 traits morphosyntaxiques disponibles est transmis au parser à son tour comme un trait d'apprentissage. Les résultats selon les deux scénarios d'évaluation sont donnés dans le tableau 9.7.

Les lignes sont triées d'après les valeurs de la colonne LAS pour les étiquettes *gold* ; dans les autres colonnes, le nombre entre parenthèses après le score indique le rang du trait pour le score et le scénario donnés. Pour tous les scores, nous examinons s'il existe une différence statistiquement significative par rapport à la *baseline*. Pour ce faire, nous utilisons le test de McNemar ; les scores pour lesquels $p < 0{,}05$ sont indiqués par "+", et ceux pour lesquels $p < 0{,}01$ par "++".

Trait	POS *gold*			POS automatiques		
	LAS	UAS		LAS		UAS	
Baseline	84,05	89,88		77,41		83,41	
personne	84,25	89,93	(6)	$77{,}88^{+}$	(6)	83,61	(5)
forme verb.	$84{,}57^{++}$	$90{,}25^{+}$	(4)	$77{,}98^{++}$	(5)	83,63	(4)
genre	$84{,}87^{++}$	90,08	(5)	$78{,}20^{++}$	(4)	83,51	(6)
nombre	$84{,}99^{++}$	$90{,}32^{+}$	(3)	$78{,}72^{++}$	(3)	$83{,}91^{+}$	(3)
morph. complète	$85{,}86^{++}$	$90{,}69^{++}$	(2)	$79{,}41^{++}$	(2)	$84{,}23^{++}$	(2)
cas	$\mathbf{86{,}97}^{++}$	$\mathbf{90{,}77}^{++}$	(1)	$\mathbf{80{,}33}^{++}$	(1)	$\mathbf{84{,}47}^{++}$	(1)

TABLE 9.7 – Résultats d'utilisation individuelle des traits morphosyntaxiques

Globalement, les résultats montrent que les traits morphosyntaxiques considérés sont plus utiles pour la labellisation des fonctions syntaxiques que pour l'identification du gouverneur : l'utilisation des traits améliore tous les scores LAS de manière statistiquement significative à l'exception de la personne dans le scénario *gold*. En revanche, les gains sont moins généralisés sur le score UAS : ce sont le cas, la morphologie complète, le nombre, et pour le scénario *gold* la forme verbale qui apportent des améliorations statistiquement significatives. Par ailleurs, le gain maximal en LAS est de 2,92 points dans les deux scénarios d'évaluation, alors qu'en UAS il est de 0,89 point dans le scénario *gold* et de 1,06 point dans le scénario automatique.

Quant aux traits individuels, le cas s'est montré le plus utile, indépendamment du scénario d'évaluation et de la mesure considérée. Il est également intéressant de noter que la morphologie complète et le nombre obtiennent respectivement les rangs 2 et 3 dans tous les scénarios observés. Le genre semble plus utile au niveau du LAS, et la forme verbale à celui de l'UAS, alors que la personne paraît le trait le moins utile.

À la différence de ce qui a été observé par Marton et al. (2013), dans nos expériences les

traits morphosyntaxiques gardent leur utilité lors du passage à l'étiquetage automatique. Cela est sans doute dû au fait que Talismane exploite les traits à partir du lexique et non pas d'une annotation automatique : l'étiquetage porte seulement sur les parties du discours, alors que les traits sont tirés du lexique externe dans les deux scénarios. Comme mentionné précédemment, cela signifie que les données morphosyntaxiques à la disposition du parser sont ambiguës, mais c'est le cas dans les deux situations d'évaluation. Cette particularité de Talismane assure donc une certaine robustesse à ce niveau lors du passage à un étiquetage automatique, malgré la chute des résultats globaux.

Il est également intéressant de noter que le cas est systématiquement plus utile que la morphologie complète. Il faut préciser que la morphologie complète ne sous-entend pas de fournir au parser tous les autres traits indépendamment les uns des autres : il s'agit d'un trait unique qui concatène les paires *trait=valeur* des autres traits disponibles. Elle contient donc toutes les informations morphosyntaxiques disponibles, mais les fournit au parser comme un bloc. Bien qu'on puisse s'attendre à ce que ce trait soit plus informatif que le cas, l'effet individuel du cas est plus marqué, avec une marge d'environ 1 point sur les scores LAS.

9.4.2 Évaluation de l'utilisation des combinaisons des traits

Ayant établi un premier bilan de l'apport individuel de différents traits morphosyntaxiques au parsing du serbe, nous souhaitons maintenant évaluer leur apport combiné. Comme l'indiquent Marton et al. (2013), certains traits morphosyntaxiques peuvent être mutuellement redondants du fait qu'ils apportent des informations sur le même phénomène syntaxique. Par conséquent, il est probable que leur apport combiné soit inférieur à une simple somme de leurs apports individuels.

Afin d'évaluer les interactions de ce type en serbe, nous établissons l'ordre de combinaison des traits suivant : comme la morphologie complète contient les valeurs de tous les autres traits, nous supposons qu'elle sera redondante par rapport aux autres. Elle est donc rajoutée en tant que dernier trait. Le premier à être exploité est le cas, étant donné son effet positif stable dans l'expérience précédente. Nous ajoutons ensuite le nombre, le deuxième trait le plus utile à l'exception de la morphologie complète. Vu le fait que le nombre, le genre et la personne participent tous les trois au marquage de l'accord, nous supposons qu'un degré de redondance peut exister entre eux également. Pour pouvoir l'évaluer, nous ajoutons ensuite le genre et la personne. Dans les deux dernières expériences, nous y joignons la forme verbale et la morphologie complète.

Les résultats obtenus sont montrés dans le tableau 9.8. Nous constatons que la combinaison des traits qui permet d'atteindre les résultats les plus élevés dans les deux scénarios est celle qui exploite tous les traits disponibles. Il est intéressant de remarquer que les

ajouts progressifs n'entraînent pas nécessairement des gains en performances : en effet, l'ajout de la personne aboutit à une légère baisse des scores dans les deux scénarios ; le genre a le même effet dans le cadre de l'évaluation sur les POS automatiques. Cela fait écho aux résultats des expériences avec les traits individuels, où la personne s'est globalement montrée comme la moins utile, et le genre apportait le gain le plus petit au niveau du score UAS dans l'évaluation sur les POS automatiques. Au-delà de notre hypothèse que les traits du nombre, du genre et de la personne étaient mutuellement redondants, ces résultats semblent indiquer que leur utilisation simultanée peut introduire de la confusion dans le parsing.

Combinaison de traits	POS *gold*		POS automatiques	
	LAS	UAS	LAS	UAS
Baseline	84,05	89,88	77,41	83,41
POS+c	86,97^{++}	90,77^{++}	80,33^{++}	84,47^{++}
POS+cn	87,21	90,77	80,67	84,50
POS+cng	87,29	90,82	*80,60*	*84,45*
POS+cngp	*87,14*	*90,64*	*80,47*	*84,13*
POS+cngpt	87,61	90,99	80,89	84,65
POS+cngptm	**88,03**	**91,31**	**81,17**	**84,70**

TABLE 9.8 – Résultats du parsing avec 6 combinaisons de traits d'apprentissage. c=cas, n=nombre, g=genre, p=personne, t=forme verbale, m=morphologie complète.

Un autre fait intéressant est que l'ajout de la morphologie complète (le dernier trait) apporte des gains comparables à ceux du cas et du nombre. Cela semble donc infirmer notre hypothèse que ce trait est redondant par rapport aux autres. Cette observation implique par conséquent que la concaténation des valeurs des autres traits propose des informations différentes au parser comparé aux traits individuels.

Néanmoins, les différences entre les scores sont minimes. Afin d'évaluer l'ampleur des effets observés, nous avons comparé chaque résultat à celui obtenu avec la combinaison précédente. Nous avons constaté que seul le cas apporte un changement statistiquement significatif.

Nous avons également examiné si l'élimination du trait de la personne facilitait le parsing en faisant un entraînement avec la combinaison POS+cngtm. Ce n'est pas le cas : les résultats obtenus sont légèrement inférieurs à ceux atteints avec la combinaison maximale des traits (LAS=87,73 et UAS=91,11 avec étiquettes *gold*, et LAS=81,17 et UAS=84,74 avec étiquetage automatique).

Compte tenu de ces résultats, nous considérons que la combinaison de traits la plus utile est celle qui regroupe tous les traits à notre disposition. C'est donc avec cette configuration que nous effectuons les optimisations des paramètres d'apprentissage automatique, décrites

dans la section suivante.

Avant de poursuivre, notons encore qu'à la différence de Marton et al. (2013), nous avons défini l'ordre indiqué ci-dessus en nous basant strictement sur les résultats des évaluations des traits individuels. Par ailleurs, nous l'avons fait avant d'entamer les expériences sur les combinaisons des traits, et ne l'avons pas modifié en fonction des résultats. Il serait donc intéressant de vérifier si une approche comparable à celle de Marton et al. (2013), basé sur une évaluation progressive de chaque combinaison, donnerait les mêmes résultats. Cette piste fait partie de nos perspectives pour la suite de ce travail.

9.5 Dernières optimisations : paramètres d'apprentissage automatique

Ayant identifié comme optimale la configuration qui exploite les étiquettes POS de base, la réduction la plus poussée du jeu d'étiquettes syntaxiques et l'ensemble des traits morphosyntaxiques disponibles, nous examinons maintenant les paramètres d'apprentissage et de parsing automatiques. Comme l'algorithme que nous utilisons est un Linear SVM (Fan et al., 2008), il exploite deux paramètres statistiques, nommés *cost* et *epsilon*. Par ailleurs, Talismane permet de définir les valeurs des paramètres de *cutoff* et de *beam width*. Le premier est utilisé en apprentissage : il représente le nombre minimal d'occurrences qu'une valeur particulière d'un trait d'apprentissage doit avoir dans le corpus d'entraînement afin d'être prise en compte dans le modèle. Typiquement, une valeur de *cutoff* plus élevée va aboutir à un modèle qui fait des généralisations plus fiables, étant donné que les traits d'apprentissage pris en compte sont moins spécifiques aux exemples trouvés dans le corpus d'apprentissage. La valeur optimale de ce paramètre dépend de la taille du corpus. Le *beam width* est un paramètre de parsing : il représente la taille de la fenêtre de décision que Talismane prend en compte. Augmenter sa valeur peut permettre à l'algorithme de corriger des erreurs commises dans l'analyse. Cependant, cela mène à une augmentation linéaire du temps de parsing : un parsing avec un *beam width*=k est k fois plus lent qu'un parsing avec un *beam width*=1 (cf. Urieli, 2013, p. 58).

Nous avons élaboré une *grid search* des différentes valeurs des paramètres de *cost* et d'*epsilon*. Cette démarche consiste à faire varier les valeurs des paramètres observés et à évaluer systématiquement les combinaisons de différentes valeurs afin de trouver la configuration optimale. Ce procédé a montré que les valeurs initiales respectives de 1,0 et 0,1 étaient optimales pour notre corpus. Nous avons donc focalisé notre attention sur les deux autres paramètres.

Nous avons effectué des tests avec les valeurs de *cutoff* allant de 1 à 5. Les résultats

sont donnés dans le tableau 9.9 [5].

	POS *gold*		POS automatiques	
cutoff	LAS	UAS	LAS	UAS
1	87,56	90,97	79,86	83,24
2	87,71	91,14	80,18	83,56
3	87,83	**91,19**	**80,28**	**83,66**
4	**87,85**	**91,19**	80,03	83,49
5	87,71	91,09	80,08	83,46

TABLE 9.9 – Résultats de parsing avec différentes valeurs de *cutoff*

Nous constatons qu'un *cutoff* de 3 donne les meilleurs résultats sur les étiquettes POS fournies par l'étiqueteur. En comparant chaque *cutoff* au suivant, la seule amélioration statistiquement significative est celle entre les valeurs 1 et 2 ($p < 0{,}025$ dans le test de McNemar), mais les valeurs plus élevées ont l'avantage de générer des modèles plus compacts. Comme *cutoff*=3 produit les meilleurs résultats sur un étiquetage automatique, nous retenons cette valeur comme optimale.

Enfin, nous avons utilisé le modèle basé sur le *cutoff*=3 pour évaluer différentes valeurs de *beam width*. Notre évaluation a porté sur les valeurs de 1, 2, 5 et 10. Les résultats sont donnés dans le tableau 9.10.

	POS *gold*		POS automatiques	
beam	LAS	UAS	LAS	UAS
1	87,83	91,19	80,28	83,66
2	88,03	91,31	81,02	84,45
5	**88,25**	**91,43**	81,58	**84,92**
10	88,18	91,36	**81,76**	84,45

TABLE 9.10 – Résultats de parsing avec différentes valeurs du faisceau

Le tableau montre que les scores augmentent jusqu'à une valeur de *beam* de 5 sur les étiquettes POS *gold*. Sur les étiquettes POS fournies par l'étiqueteur, les scores s'améliorent même avec le *beam* de 10. La prise en compte d'un contexte de décision plus large bénéficierait donc plus encore au traitement d'un texte étiqueté automatiquement. On peut faire l'hypothèse que cette configuration permet au parser de compenser certaines erreurs induites par une annotation morphosyntaxique erronée. Notons cependant que les

5. Les expériences de cette section ont été réalisées avec une version de Talismane antérieure à celle utilisée pour les évaluations des traits morphosyntaxiques, d'où la légère différence des scores de base par rapport à ceux donnés à la fin de la section précédente. Vu les contraintes de temps, les *grid search* pour les paramètres automatiques n'ont pas été refaites avec la nouvelle version. Néanmoins, cela a été fait pour l'expérience finale qui a permis d'établir le modèle diffusé. Nous avons utilisé les valeurs des paramètres identifiées comme optimales dans cette section.

seules améliorations significatives sont liées au changement de valeur de *beam* de 1 à 2 et de 2 à 5 ($p < 0{,}005$ dans le test de McNemar), alors que celui de 5 à 10 ne l'est pas, bien qu'il donne une légère augmentation du score LAS dans le scénario basé sur l'étiquetage automatique. Par conséquent, nous retenons *beam width*=5 comme la valeur optimale.

Au vu de ces résultats, nous retenons comme optimale la configuration suivante :
— jeu d'étiquettes POS de base,
— jeu d'étiquettes syntaxiques de la Réduction 3 (toutes les étiquettes sous-spécifiées réduites à une étiquette unique Dep),
— combinaison des traits morphosyntaxiques maximale (cngptm),
— *cutoff*=3 et *beam width*=5.

Cette configuration a été ré-entraînée et évaluée. Les résultats obtenus sont analysés dans la section suivante.

9.5.1 Évaluation finale de la configuration optimale

Nous avons évalué la configuration optimale sur les sections *dev* et *test* du corpus, avec des étiquettes POS *gold* et avec un étiquetage automatique. Les résultats sont donnés dans le tableau 9.11. Pour faciliter la comparaison, nous reprenons les scores atteints avec les valeurs par défaut des paramètres de *cutoff* et de *beam width* (cf. la configuration POS+cngptm de la section 9.4.2).

Section	POS *gold*		POS automatiques	
	LAS	UAS	LAS	UAS
dev baseline	88,03	91,31	81,17	84,70
dev	88,25	91,66	82,28	85,90
test	87,48	91,22	78,73	82,92

TABLE 9.11 – Résultats de la configuration optimale sur les sections *dev* et *test*

Comme attendu, l'ajustement des paramètres de *cutoff* et de *beam width* apporte une amélioration sur la section *dev* : nous constatons un gain de +0,22 en score LAS et de +0,35 en score UAS dans le scénario basé sur les étiquettes *gold*, alors que sur les étiquettes fournies de manière automatique le gain en LAS est de +1,11 et en UAS de +1,20. Dans le scénario exploitant les étiquettes *gold*, les résultats sur la section *test* sont légèrement plus bas que ceux de la section *dev*, mais restent proches. En revanche, dans le scénario basé sur l'étiquetage automatique, on constate une différence importante entre les deux portions du corpus. Cela pourrait indiquer que le modèle entraîné est sur-ajusté aux données de la section *dev* et qu'il se généralise mal en passant à la section *test*. Afin d'examiner cette question, dans la section suivante nous étudions de plus près le traitement de plusieurs relations syntaxiques particulières.

Avant d'aborder cette analyse, rappelons que l'état de l'art en parsing du serbe a été atteint par Agić & Ljubešić (2015) : leurs meilleurs résultats sur le serbe sont de 81,5 en LAS et de 86,0 en UAS. Les résultats en l'occurrence ont été atteints sur des étiquettes *gold*, avec utilisation des traits morphosyntaxiques atomiques. Nous constatons que nos résultats du même type (ceux sur la section *test* avec des étiquettes *gold*) dépassent ces scores de 5,98 en LAS et de 5,22 en UAS. Cependant, d'autres aspects de nos expériences diffèrent de manière importante : le corpus d'entraînement utilisé par Agić & Ljubešić (2015) est journalistique alors que le nôtre est littéraire ; nos schémas d'annotation ne sont pas identiques (ils utilisent ceux du projet Universal Dependencies) ; différents types de parsers ont été utilisés (Talismane est un parser par transitions, alors que Mate est basé sur les graphes). Par conséquent, les résultats ne sont pas directement comparables. Il est néanmoins intéressant de noter que les différences citées devaient favoriser les scores de Agić & Ljubešić (2015) : les textes journalistiques sont en général considérés comme plus faciles que les textes littéraires, les parsers par graphes sont censés être plus adaptés aux langues à la morphologie flexionnelle riche, et les schémas d'annotation UD visent l'optimisation du parsing indépendamment de la langue. Des évaluations plus directes sont nécessaires pour comprendre les effets de ces différents paramètres.

9.5.2 Analyse du traitement de quelques fonctions syntaxiques

Ici, nous proposons une comparaison du traitement d'un sous-ensemble des relations syntaxiques dans les sections *dev* et *test*. Nous basons cette comparaison sur la f-mesure des étiquettes syntaxiques individuelles, en la considérant comme un indicateur de la maîtrise générale du modèle par rapport à l'étiquette en question. Nous souhaitons confronter ces scores aussi bien entre les scénarios d'évaluation qu'entre les deux sections. Compte tenu des résultats globaux, nous nous attendons à observer des chutes sur les deux sections lors du passage à l'étiquetage automatique. En comparant les pertes sur les deux sections, nous voulons vérifier si les mêmes étiquettes étaient difficiles dans les deux cas. Deuxièmement, nous souhaitons confronter les deux échantillons et comparer directement les valeurs de la f-mesure obtenues sur chacun d'entre eux. Si les scores obtenus sur la section *test* se montraient systématiquement inférieurs à ceux de la section *dev*, cela constituerait une indication que le modèle de parsing est sur-ajusté aux données de la section *dev*.

Dans cette analyse, nous nous focalisons sur le rattachement labellisé, qui prend en compte à la fois l'identification du gouverneur d'un token et sa fonction syntaxique. Nous prenons en considération les sujets, les objets, les prédicatifs et la fonction générale Dep, ainsi que l'ensemble des fonctions relatives au traitement des subordonnées (étiquette Sub qui marque les subordonnants et les étiquettes en Pred, dédiées aux prédicats de différents types de subordonnées). Nous retenons les étiquettes qui ont au moins 10 occurrences dans

les deux échantillons évalués. Les résultats sont montrés dans le tableau 9.12.

Le premier segment du tableau montre les valeurs de la f-mesure pour les étiquettes retenues, sur les sections *dev* (D) et *test* (T), obtenues en utilisant les POS *gold*($_G$) ou automatiques ($_A$). Ensuite, pour chacune des étiquettes, nous observons les variations de la f-mesure selon différents critères.

Dans le deuxième segment du tableau, nous calculons la différence en f-mesure entre le scénario *gold* et le scénario automatique observée sur chacun des échantillons. D_G vs D_A correspond donc à la différence entre le *dev gold* et le *dev* automatique, et T_G vs T_A à celle entre le *test gold* et le *test* automatique.

Dans le troisième segment, nous confrontons directement les scores des deux sections selon le scénario d'évaluation : D_G vs T_G exprime la différence entre le *dev gold* et le *test gold*, alors que la colonne D_A vs T_A exprime celle entre *dev* automatique et *test* automatique.

Étiquette	D_G	D_A	T_G	T_A	D_G vs D_A	T_G vs T_A	D_G vs T_G	D_A vs T_A
Root	93,00	74,12	94,31	73,73	-18,88	-20,58	+1,31	-0,39
Suj	92,01	84,82	87,28	74,93	-7,19	**-12,35**	-4,73	**-9,89**
ObjDir	89,76	87,80	90,74	85,01	-1,96	**-5,73**	+0,98	-2,79
ObjIndirCas	91,43	83,67	85,88	76,54	-7,76	-9,34	**-5,55**	**-7,13**
Dep	93,64	90,99	92,96	89,28	-2,65	-3,68	-0,68	-1,71
PredicNom	74,58	64,00	78,02	53,15	-10,58	**-24,87**	+3,44	**-10,85**
PredicOpt	79,17	76,60	56,34	53,85	-2,57	-2,49	**-22,83**	**-22,75**
Sub	88,33	83,98	88,12	82,98	-4,35	-5,14	-0,21	-1,00
PredSub	73,91	54,32	85,71	70,59	**-19,59**	-15,12	**+11,80**	**+16,27**
PredRel	82,61	78,52	89,14	87,86	**-4,09**	-1,28	**+6,53**	**+9,34**
PredCompletive	84,56	81,94	91,21	89,84	**-2,62**	-1,37	**+6,65**	**+7,90**

TABLE 9.12 – F-mesure de différentes étiquettes selon l'échantillon et le scénario d'évaluation. D_G=*dev* avec POS *gold*, D_A=*dev* avec POS automatiques, T_G=*test* avec POS *gold*, T_A=*test* avec POS automatiques.

Effectivement, les valeurs de la f-mesure chutent systématiquement sur les deux sections de corpus en passant à l'étiquetage automatique. Nous observons également que les pertes sur la section *test* sont majoritairement plus importantes que celle observées sur la section *dev*. C'est plus prononcé sur le sujet (Suj), l'objet direct (ObjDir) et le prédicatif nominal (PredicNom). Cependant, il faut noter que c'est la section *dev* qui exhibe des chutes plus importantes sur les étiquettes des prédicats des subordonnées. Cela semble indiquer que le traitement des subordonnées est plus facile dans la section *test*. Ce fait est confirmé par les résultats des comparaisons entre les sections : indépendamment du scénario d'évaluation, les étiquettes des prédicats de subordonnées sont mieux traitées dans la section *test*. Quant aux autres relations analysées, l'image est moins claire. Dans le scénario avec les étiquettes *gold*, le sujet (Suj), l'objet indirect casuel (ObjIndirCas) et le

prédicatif optionnel (PredicOpt) sont moins bien traités dans la section *test*, mais la racine (Root) et le prédicatif nominal (PredicNom) marquent une différence positive par rapport à la section *dev*. En revanche, dans le scénario avec l'étiquetage automatique, les scores sont uniformes : c'est systématiquement sur la section *test* que les f-mesures sont moins élevées.

Au vu de ces résultats, il est possible qu'il y ait eu sur-ajustement du modèle de parsing, au moins sur les fonctions qui relèvent des dépendants du verbe : en passant à l'étiquetage automatique, la section *test* manifeste des pertes en f-mesure beaucoup plus importantes que la section *dev*. Par ailleurs, les scores de ces fonctions sont systématiquement plus bas sur la section *test* que sur la section *dev* quand on utilise l'étiquetage automatique. En revanche, les relations qui relèvent des subordonnées ne suivent pas les mêmes tendances : elles sont mieux traitées dans la section *test* que dans la section *dev*, et elles subissent des pertes moins importantes sur *test* que sur *dev* lors du passage à l'étiquetage automatique sur la même section. Il est difficile de dire si ces effets relèvent effectivement d'un sur-ajustement du modèle sur les données de la section *dev* : il se peut qu'ils soient dus aux spécificités du contenu des sections respectives. Par conséquent, une étude plus poussée de cet aspect de l'annotation doit être effectuée pour identifier les causes.

9.6 Conclusions et pistes

Nous récapitulons les résultats de différentes expériences présentées ci-dessus. Tout d'abord, nous avons identifié la configuration optimale pour l'entraînement de Talismane sur notre corpus. Il s'agit d'un entraînement basé sur les étiquettes POS de base, sur un jeu d'étiquettes syntaxiques réduit, sur les 6 traits morphosyntaxiques individuels puisés dans le lexique, et sur les valeurs des paramètres de *cutoff*=3 et de *beamwidth*=5. Sur la section *dev*, ce modèle final atteint un LAS de 88,25 et un UAS de 91,66 sur les étiquettes *gold*, et un LAS de 82,28 et un UAS de 84,70 sur l'étiquetage automatique. Si l'on considère la *baseline* (obtenue seulement avec les étiquettes POS, sans traits morphosyntaxiques et sans optimisation des paramètres d'apprentissage automatique), il s'agit d'un gain de +4,2 en LAS et +1,78 en UAS sur les étiquettes *gold*, alors que sur l'étiquetage automatique les gains sont de +4,87 en LAS et de +1,29 en UAS. Sur la section *test*, ses résultats finaux sont de 87,48 en LAS et de 91,22 en UAS sur les étiquettes *gold*, et de 78,73 en LAS et de 82,92 en UAS avec un étiquetage automatique. Le modèle final est librement disponible à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Models/Parsing/Talismane`.

Nos résultats sur la section *test* avec des étiquettes POS *gold* sont également plus élevés que l'état de l'art actuel en parsing du serbe : nous marquons des gains de +5,98 en LAS et de +5,22 en UAS par rapport aux meilleurs résultats signalés dans (Agić & Lju-

bešić, 2015). Néanmoins, de nombreuses différences entre nos configurations d'évaluation respectives existent ; les résultats ne peuvent donc pas être considérés comme directement comparables.

Quant aux autres enseignements de ce travail, l'inclusion du cas dans les étiquettes POS a eu l'effet attendu : elle facilite le parsing sur des étiquettes *gold*, mais elle rend l'étiquetage morphosyntaxique plus difficile, ce qui affecte négativement le parsing basé sur un étiquetage morphosyntaxique automatique. Au niveau du jeu d'étiquettes syntaxiques, le remplacement des étiquettes sous-spécifiées détaillées par une étiquette globale Dep aboutit à des améliorations légères, plus prononcées sur l'étiquetage automatique. Ceci est possiblement dû à la corrélation entre la nature morphosyntaxique des tokens et la définition de certaines de nos étiquettes syntaxiques.

En ce qui concerne l'exploitation d'informations morphosyntaxiques en tant que traits d'apprentissage, cette démarche a un effet positif systématique. Par ailleurs, elle paraît avoir un effet plus important sur l'identification des dépendances étiquetées que sur l'identification du gouverneur seul, étant donné que les gains observés étaient systématiquement plus importants au niveau du LAS qu'à celui de l'UAS. Quant aux traits individuels, le cas, la morphologie complète et le nombre se sont avérés les plus utiles indépendemment du scénario d'évaluation. Par ailleurs, de manière générale, les traits morphosyntaxiques gardent leur utilité lors du passage à un étiquetage automatique. Ceci est sans doute dû à la manière dont le parser Talismane exploite les informations morphosyntaxiques.

Notre dernière série des expériences a porté sur l'optimisation des paramètres d'apprentissage automatique de *cutoff* et de *beam width*. Les valeurs identifiées ont été utilisées pour l'entraînement et l'évaluation du modèle final. Cependant, lors de l'évaluation de ce modèle sur la section *test* du corpus, nous avons constaté des scores systématiquement plus bas que sur la section *dev*. Un premier examen du traitement d'un sous-ensemble des étiquettes syntaxiques suggère qu'il pourrait y avoir eu un sur-ajustement du modèle aux données de la section *dev*, sur laquelle il a été paramétré. Néanmoins, les mêmes effets pourraient être dus aux spécificités des deux sections du corpus.

Pour la suite de ce travail, notre premier objectif est de neutraliser autant que possible l'effet d'utilisation d'un étiquetage automatique. Nous allons donc travailler sur l'amélioration de l'étiquetage morphosyntaxique par Talismane. Pour le moment, l'exactitude moyenne est de 94 %. Une analyse d'erreur détaillée devrait nous donner des pistes d'amélioration : en fonction des résultats, nous pourrons envisager différentes mesures correctives, telles l'élaboration de ressources lexicales complémentaires, l'optimisation des traits d'apprentissage au niveau de l'étiquetage, ou encore l'élaboration des règles symboliques visant des problèmes précis. Les expériences d'A. Urieli avec les deux dernières méthodes ont donné des résultats prometteurs sur le français (cf. Urieli, 2013, p. 159-175, 185-187).

Nous allons également examiner la question du sur-ajustement du modèle. Comme noté

ci-dessus, toutes les évaluations présentées ont été effectuées comme des passes uniques sur les données de la section *dev*. Nous allons donc chercher à vérifier les effets observés dans le cadre d'une évaluation croisée à 10 itérations. Par ailleurs, nous disposons maintenant de la version complète du corpus ParCoTrain-Synt. Nous pourrons donc baser ce travail sur les 101 000 tokens désormais disponibles. Cette approche montrera si les performances discutées ci-dessus sont stables et pourrait nous aider à vérifier s'il y a eu un sur-ajustement du modèle actuel.

Enfin, nous allons également ré-évaluer la manière dont les traits syntaxiques sont combinés, cette fois en mettant en place une méthode basée sur une évaluation progressive de chaque ajout, à l'instar de celle décrite dans (Marton et al., 2013). Cette démarche permettra d'abord de vérifier la pertinence des combinaisons évaluées ici, et pourrait en suggérer de meilleures.

Plus globalement, ces expériences montrent que notre décision d'encoder les traits morphosyntaxiques fins dans le corpus est pertinente. Bien que dans ce cadre précis les traits aient été puisées du lexique et non pas du corpus, cela est dû à une particularité du parser utilisé. Le fait d'avoir annoté le corpus avec ces informations garantit leur disponibilité aux parsers qui se basent sur le corpus d'apprentissage dans cet aspect.

Enfin, les expériences présentées ici démontrent l'adéquation du corpus ParCoTrain-Synt pour des exploitations en parsing : comme mentionné ci-dessus, ce corpus a permis d'entraîner un modèle dont les résultats représentent le nouvel état de l'art en parsing de cette langue. Afin d'évaluer plus pleinement l'utilité de ParCoTrain-Synt, la partie suivante de cette thèse met à l'épreuve d'autres aspects de ce corpus, et notamment sa capacité à alimenter des recherches linguistiques.

Troisième partie

Exploitations du corpus annoté syntaxiquement ParCoTrain-Synt

Présentation de la partie III

Comme indiqué dans l'introduction de ce document, l'un des objectifs de cette thèse est de fournir un treebank utile aussi bien à la linguistique qu'au TAL. Dans le chapitre 9, nous avons déjà illustré un mode d'exploitation de ParCoTrain-Synt : en nous basant sur ce corpus, nous avons entraîné, paramétré et évalué un modèle de parsing du serbe dont les résultats représentent le nouvel état de l'art. Dans cette troisième et dernière partie de cette thèse, nous cherchons à pousser plus loin cette évaluation de l'utilité des ressources créées, que ce soit dans le domaine du TAL ou de la linguistique théorique. Dans cet objectif, nous proposons deux études indépendantes.

Dans le chapitre 10, nous traitons une question exclusivement linguistique, à savoir la position et la structure du groupe adjectival en serbe. Nous comparons nos résultats avec les descriptions existantes de ce phénomène dans les grammaires serbes et montrons comment le recours à un corpus annoté permet d'élargir le nombre de cas de figure couverts par l'analyse, et ainsi de quantifier les phénomènes identifiés et d'en explorer les propriétés.

Dans le chapitre 11, nous examinons un sujet qui intéresse le TAL et la linguistique : la non-projectivité. Ce phénomène a un impact important sur le TAL : les structures non projectives complexifient la tâche du parsing et exigent l'utilisation de parsers adaptés. Et comme différents degrés et différentes formes de non-projectivité se manifestent selon les langues, ce phénomène suscite un intérêt particulier dans le cadre de la syntaxe théorique : les constructions non projectives présentes dans différentes langues sont examinées afin d'identifier les principes de la non-projectivité en général. Une analyse automatique du corpus ParCoTrain-Synt nous permet d'extraire les structures non projectives, d'évaluer leur complexité formelle et de les classer en fonction des constructions linguistiques qu'elles instancient. Nous utilisons ensuite ces critères afin de positionner le serbe par rapport à d'autres langues. Par ailleurs, le versant TAL de cette étude a consisté à évaluer la capacité de deux méthodes de parsing à traiter les constituants discontinus. Cette étude apporte donc un nouvel éclairage en syntaxe théorique aussi bien qu'en TAL, dans une perspective monolingue ainsi que contrastive.

Chapitre 10

Position et structure du groupe adjectival en serbe : une approche empirique

Ce chapitre est consacré à l'étude d'une facette spécifique de la variation de l'ordre des mots en serbe : la position de l'adjectif au sein du GN [1]. En effet, l'adjectif serbe est canoniquement antéposé à sa tête nominale (cf. *puna kuća* lit. 'pleine maison', 'maison pleine'), mais il peut également être postposé (cf. *kuća puna dece* lit. 'maison pleine enfants.GEN', 'maison pleine d'enfants'). Dans le corpus ParCoTrain-Synt, environ 6 % des adjectifs gouvernés par un nom sont postposés, et parmi les adjectifs qualificatifs, ce taux est de 9 %. La postposition n'est donc pas un phénomène marginal en serbe. Elle n'est cependant pas traitée dans les grammaires de référence comme (Stanojčić & Popović, 2012) et (Mrazović, 2009) : ces ouvrages se contentent de proposer une description de la structure canonique du GN, sans mentionner la possibilité de la postposition de l'adjectif. Ce positionnement alternatif en serbe est présenté dans le travail sur les langues slaves de (Siewierska & Uhlířova, 1998), et quelques autres particularités du comportement de l'adjectif sont évoquées dans (Bošković, 2013), mais aucune des deux études ne propose des résultats quantitatifs.

En revanche, la position de l'adjectif est une question bien étudiée dans les langues romanes. À la différence du serbe et des langues slaves en général [2], dans ces langues l'adjectif se positionne canoniquement à droite du nom. Néanmoins, certains adjectifs peuvent

1. Les termes de groupe nominal (GN), groupe adjectival (GA) et groupe prépositionnel (GP) seront utilisés systématiquement dans ce chapitre. Ils ne signalent cependant pas un passage au cadre de la syntaxe en constituants : il s'agit simplement d'une manière économe de référer à un sous-arbre gouverné par un mot d'une catégorie grammaticale donnée.

2. Hormis le polonais, qui exhibe des comportements différents en fonction du type d'adjectif, (cf. Siewierska & Uhlířova, 1998, p. 134).

également apparaître en antéposition : dans le corpus French Treebank, 28,6 % des occurrences d'adjectifs sans dépendant post-adjectival se trouvent à gauche du nom, et environ 9,5 % des lemmes adjectivaux ont été détectés dans les deux positions (cf. Thuilier et al., 2012). Les travaux consacrés au positionnement de l'adjectif dans les langues romanes portent donc principalement sur l'identification des adjectifs capables d'être antéposés et des conditions qui autorisent ou facilitent ce positionnement alternatif. Pour ce faire, certains travaux s'appuient sur des critères lexico-sémantiques (cf. Bouchard, 1998 ; Cinque, 2010), alors que d'autres exploitent différents critères syntaxiques. Ces derniers ont été utilisés aussi bien dans des études théoriques (cf. Abeillé & Godard, 1999) que dans des travaux sur corpus, aboutissant pour certains à la définition de méthodes de prédiction automatique de la place de l'adjectif (cf. Thuilier et al., 2012 ; Thuilier, 2012 ; Gulordava & Merlo, 2015 ; Gulordava et al., 2015).

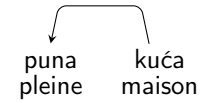
Compte tenu de l'utilité des critères syntaxiques dans les études citées ci-dessus, nous orientons notre étude dans cette direction. Ce choix présente également un avantage pratique : comme nos données sont dotées d'une annotation syntaxique, elles se prêtent facilement à une analyse automatique basée sur ces critères, ce qui n'est pas le cas des critères sémantiques. Nous procédons de la manière suivante : en nous basant sur le corpus ParCoTrain-Synt, nous effectuons une extraction automatique de données et une analyse initiale automatique, suivies d'un examen manuel détaillé.

Le chapitre est organisé comme suit : nous présentons d'abord rapidement les travaux existants sur la place de l'adjectif en serbe et dans d'autres langues, en nous focalisant sur ceux qui ont inspiré ce travail (section 10.1). La méthode utilisée pour le recueil et l'analyse des données est détaillée dans la section 10.2. Après avoir effectué une première analyse de la position de l'adjectif en fonction de sa sous-catégorie grammaticale (section 10.3), nous analysons le comportement des deux sous-catégories les plus mobiles : les possessifs (section 10.4) et les qualificatifs (section 10.5). Nous discutons ensuite les effets de différents critères syntaxiques que nous avons observés sur nos données (section 10.6). Enfin, nous proposons des conclusions et des pistes pour la suite de cette étude (section 10.7).

10.1 Étude de la position de l'adjectif en serbe et dans d'autres langues

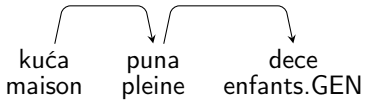
Comme mentionné ci-dessus, l'adjectif en serbe se trouve canoniquement en antéposition (cf. exemple 22a), mais un même adjectif peut également se trouver à la droite de sa tête (cf. exemple 22b). Il est important de remarquer qu'une antéposition du GA dans le deuxième cas de figure n'est pas possible (cf. exemple 22c). Par ailleurs, l'acceptabilité de la postposition dans l'exemple 22d est douteuse hors contexte. Cet examen rapide montre que ce positionnement n'est pas aléatoire, et indique l'existence de certaines contraintes.

(22) a.



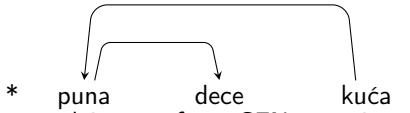
'maison pleine'

b.

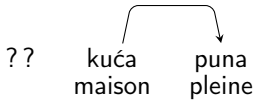


'maison pleine d'enfants'

c.



d.



Or, peu d'attention est accordée à cette variation dans les grammaires serbes actuelles. Stanojčić & Popović (2012) abordent la position de l'adjectif dans la section dédiée aux principes généraux d'organisation des syntagmes (p. 375-376). Pour tout type de tête de syntagme, ils indiquent que les constituants dits « ajouts spécifiques » se positionnent à gauche de leur tête (cf. colonne *Ajout spécifique* dans la figure 10.1), alors que les autres constituants se retrouvent « majoritairement à droite de la tête » (cf. colonne *Autres constituants* dans la même figure). La première catégorie correspond aux dépendants adjectivaux pour les GN, et aux dépendants adverbiaux pour les GA. La deuxième comprend les GP et les GN à cas oblique pour les deux types de têtes, ainsi que les relatives dans le cas des GN.

	Ajout spécifique	**Tête**	**Autres constituants**
Syntagme nominal	crna ona	torba	od kože koja leži na stolu
Syntagme adjectival	veoma	odan	drugovima

FIGURE 10.1 – Structure du GN et du GA d'après Stanojčić & Popović (2012, p. 376). *crna* = 'noire', *ona* = 'cette', *torba* = 'sac', *od kože* = 'en cuir', *koja leži na stolu* = 'qui est posée sur la table' ; *veoma* = 'très', *odan* = 'loyal', *drugovima* = 'ami.DAT.PL'

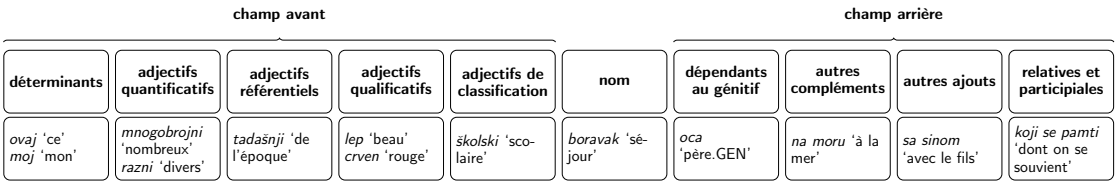


FIGURE 10.2 – Structure du GN en serbe d'après (Mrazović, 2009)

Quant à l'organisation interne de ces deux positions (à gauche et à droite de la tête), les auteurs l'expliquent par la proximité sémantique du dépendant par rapport au nom : les éléments « plus proches » de la tête sont censés se positionner dans son voisinage immédiat, alors que les éléments moins étroitement liés au nom se retrouvent à la périphérie du groupe. Les adjectifs qualificatifs sont ainsi considérés comme plus proches du nom que les possessifs, les indéfinis ou les démonstratifs, étant donné la position initiale de ces derniers (cf. ***ova/neka/svaka*** *nova knjiga* '**ce/un/tout** nouveau livre').

Notons que les auteurs présentent seulement la structure canonique de chaque type de syntagme, sans évoquer les permutations possibles, que ce soit au niveau du GN ou à celui du GA.

Une analyse plus détaillée du GN en serbe est proposée dans (Mrazović, 2009). L'auteure décrit la structure du groupe nominal en termes de *champ avant* et *champ arrière*. Tout comme Stanojčić & Popović (2012), elle indique que le champ avant est destiné à accueillir différents types d'adjectifs, alors que le champ arrière est réservé aux compléments et ajouts gouvernés par le nom. À la différence de Stanojčić & Popović, Mrazović définit en détail la structure interne des deux champs. Cette structure, qui reflète l'ordre de linéarisation des éléments du GN, est montrée dans la figure 10.2.

Avant de commenter cette proposition, rappelons d'abord que cet ouvrage est la seule grammaire du serbe qui reconnaît l'existence de la catégorie des déterminants dans cette langue. L'auteure y compte les formes des possessifs (p. ex. *moj* 'mon'), des démonstratifs (p. ex. *ovaj* 'ce'), des interrogatifs (p. ex. *koji* 'quel'), des indéfinis (p. ex. *neki* 'un (certain)'), des négatifs (p. ex. *ničiji* 'à personne'), des quantifieurs (p. ex. *mnogi* 'nombreux'), ainsi que les formes de *jedan* 'un' et *sam* 'seul'. Elle analyse ces formes comme déterminants même quand elles sont utilisées indépendamment d'un nom (cf. *Moj je stigao, a tvoj nije* lit. 'Mon est arrivé, et ton non', 'Le mien est arrive, et le tien non'), en supposant une ellipse du nom gouverneur. [3]

Si l'on observe maintenant la schématisation proposée de la structure du GN, on constate qu'elle repose en grande mesure sur des critères sémantiques. Dans le champ

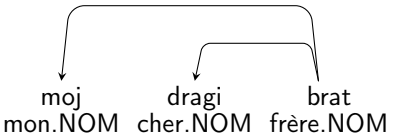
3. Ce traitement diffère donc de celui que nous accordons à ces formes dans notre corpus : nous considérons ces formes comme des adjectifs quand elles dépendent d'un nom, et les traitons comme des pronoms quand elles sont utilisées indépendamment d'un nom (cf. section 5.1.2).

avant, la première position est la seule qui est définie par des critères syntaxiques aussi bien que sémantiques, vu qu'elle est réservée aux déterminants. En revanche, la distinction entre les positions 2 à 5 est basée exclusivement sur les propriétés sémantiques des adjectifs. L'ordonnancement des différentes catégories est expliqué par le niveau de spécification qu'ils apportent au nom : ces dépendants sont censés fournir des informations de plus en plus spécifiques en allant de gauche à droite.

Le champ arrière est quant à lui défini par une combinaison de propriétés syntaxiques et sémantiques. Ainsi, l'auteure explique la position finale des relatives par leur longueur, et indique que la position adjacente au nom est réservée aux dépendants au génitif. En revanche, les positions 2 et 3 sont définies à travers l'opposition *complément* vs *ajout*. Les critères pour opérer cette distinction ne sont pas clairs : dans l'exemple *hotel* ***na moru*** 'hôtel **à la mer**', le GP est considéré comme un complément, alors que dans le cas de *rasprava* ***na hodniku*** 'discussion **dans le couloir**', le GP est analysé comme un ajout. Leurs réalisations syntaxiques sont néanmoins identiques (préposition *na* 'sur' suivie d'un nom au locatif), et leurs contenus sémantiques du même type (locatifs). On en déduit que la différenciation se fait à partir des propriétés sémantiques des noms-têtes.

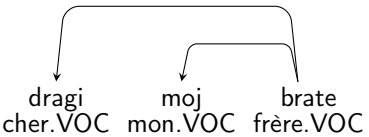
À la différence de (Stanojčić & Popović, 2012), Mrazović indique également quelques linéarisations alternatives possibles. Dans le champ avant, elle évoque l'effet du vocatif, qui peut mener à une inversion du déterminant et de l'adjectif qualificatif (cf. exemple 23).

(23) a.



'mon cher frère'

b.

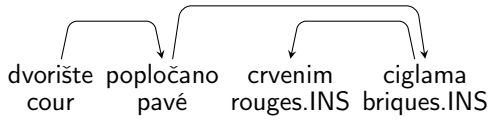


idem

Quant au champ arrière, l'auteure évoque une permutation intéressante : l'antéposition des constructions dites participiales. Un adjectif participe peut se retrouver à gauche du nom pour réaliser une topicalisation, mais alors les dépendants de l'adjectif se déplacent à gauche de leur tête également (cf. exemples 24a et 24b). L'auteure note au passage que les mêmes configurations sont autorisées pour un adjectif non-déverbal (cf. exemples 24c et 24d).

Il est remarquable qu'aucun des deux ouvrages ne commente explicitement la postpo-

(24) a.



'cour pavée avec des briques rouges'

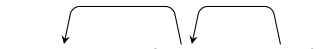
b.

idem

c.

'homme digne de respect'

d.



idem

sition de l'adjectif, d'autant plus que Mrazović utilise des exemples qui l'instancient (cf. exemple 24c). Les exemples 24b et 24d font également référence à une propriété exceptionnelle d'un GA gouverné par un nom, à savoir sa capacité à réorganiser ses dépendants en fonction du positionnement de l'adjectif par rapport au nom. Cependant, ce fait n'est pas analysé davantage par Mrazović, et cette question n'est pas évoquée par Stanojčić & Popović non plus. Globalement, les deux descriptions présentées ci-dessus font rarement référence aux caractéristiques syntaxiques des différents éléments du GN, leur focus portant plutôt sur les propriétés sémantiques des adjectifs. Or, des critères syntaxiques ont été utilisés avec succès pour expliquer certaines caractéristiques du comportement adjectival en langues romanes. Nous en examinons deux dans la suite.

10.1.1 Position de l'adjectif et notion de poids syntaxique

Dans leur travail sur le français, Abeillé & Godard (1999) introduisent la notion de poids syntaxique dans l'étude de la position de l'adjectif. Ce trait à deux valeurs (léger et non léger) est défini à la fois au niveau lexical (un lexème peut être léger ou non) et syntaxique (un syntagme peut être léger ou non). Les adjectifs systématiquement antéposés sont considérés comme lexicalement légers, ceux qui sont systématiquement postposés sont considérés comme lexicalement lourds, alors que les adjectifs alternants sont traités comme

sous-spécifiés quant à ce trait. Au niveau syntaxique, seule la modification d'un élément léger par un autre élément léger ainsi que la coordination de deux éléments légers sont considérées comme légères. Par conséquent, un adjectif doté d'un modifieur syntagmatique (typiquement d'un GP) est lourd. Comme les éléments légers précèdent les éléments non légers, les GA de ce type sont postposés.

Thuilier (2012) évalue le critère du poids en considérant la structure et la longueur du GA. Les données présentées dans ce travail montrent que les GA d'une longueur supérieure à 2 syllabes sont très majoritairement postposés (au-delà de 90 %)[4]. Par ailleurs, elle montre que la présence d'un modifieur adverbial et celle d'une coordination sont des paramètres qui favorisent la postposition. Cette tendance d'un GA complexe – et par conséquent long – à être postposé est interprétée par l'auteure comme une preuve de la pertinence du principe de poids pour le positionnement de l'adjectif français (p. 177).

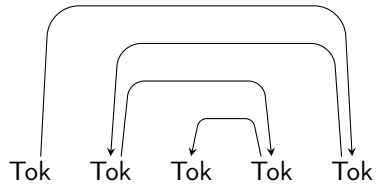
Gulordava & Merlo (2015) adoptent une définition du poids purement syntaxique et définissent comme lourd tout GA contenant 2 mots ou plus. Elles analysent la position de ce type de GA dans les treebanks de 17 langues différentes. Les résultats montrent que globalement les GA lourds sont plus fréquemment placés en postposition que les GA légers. Les auteures notent cependant que cet indicateur en tant que tel ne prend pas en compte les spécificités typologiques de différentes langues, et notamment la tendance des langues à ordre OV à placer les éléments longs avant les éléments courts. Afin de raffiner leur analyse, elles examinent le principe de minimisation de la longueur des dépendances.

10.1.2 Minimisation de la longueur des dépendances (DLM) à l'intérieur du GN

Le principe de la minimisation de la longueur des dépendances (angl. *dependency length minimization* ; dorénavant DLM) a été proposé dans les travaux de Temperley (2007) et Gildea & Temperley (2010). Il postule que les langues naturelles ont tendance à organiser les dépendants autour de leur tête de manière à favoriser les arcs de dépendance les plus courts possibles. La longueur d'une dépendance (DL) est exprimée comme le nombre de tokens que cette dépendance couvre ; si le gouverneur et le dépendant sont adjacents, la longueur est égale à 1. Suivant ce principe, les éléments de la phrase sont organisés de sorte à ce que la somme de toutes les DLs dans la phrase soit aussi petite que possible. Cette tendance a été mise en relation avec les propriétés du dispositif cognitif humain, et notamment avec la taille de la mémoire de travail (Liu, 2008). Dans les exemples 25a à 25e, nous montrons quelques illustrations de ce principe reprises de (Gildea & Temperley, 2010). Nous les expliquons dans la suite.

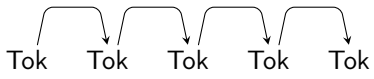
4. Les adjectifs dotés de modifieurs post-adjectivaux sont exclus de cette analyse, vu que cette structure est obligatoirement postposée en français. Thuilier (2012) considère donc les GA contenant des modifieurs adverbiaux et des coordinations.

(25) a.



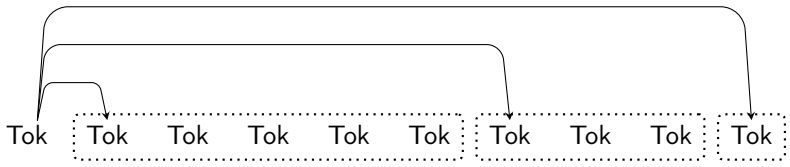
DL=4+3+2+1=10

b.



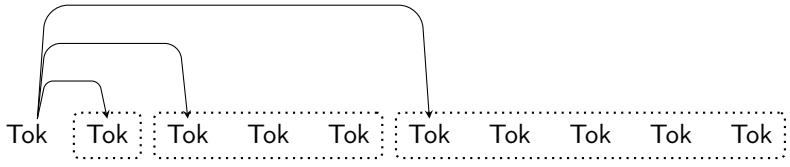
DL=1+1+1+1=4

c.



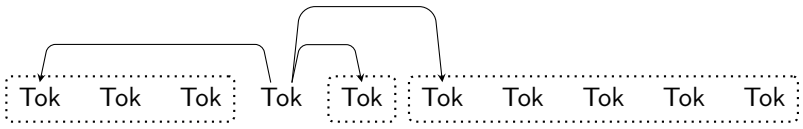
DL=1+6+9=16

d.



DLM=1+2+4=7

e.



DL=3+1+2=6

Comme l'indiquent Temperley (2008) et Gildea & Temperley (2010), le principe de la DLM est corroboré par plusieurs observations empiriques sur différentes langues. Les exemples 25a et 25b montrent qu'une organisation de dépendants en cascade est plus favorable qu'une organisation emboîtée selon la DLM. Cela coïncide avec le principe général de cohérence globale du sens des dépendances dans une langue (langues *head-initial* vs *head-final*). Quand on compare l'exemple 25c à l'exemple 25d, on remarque que le fait de positionner les dépendants en fonction de leur longueur permet également d'optimiser la longueur des dépendances, ce qui correspond au principe *short before long* (Hawkins, 1994, 2004). Enfin, l'exemple 25e illustre le fait que la disposition des dépendants des deux côtés

du gouverneur favorise encore la minimisation de la longueur des dépendances, ce qui est confirmé par la tendance des dépendants courts à avoir un sens de dépendance contraire au sens global (*inconsistent one-word constituent branching*, (Dryer, 1992)).

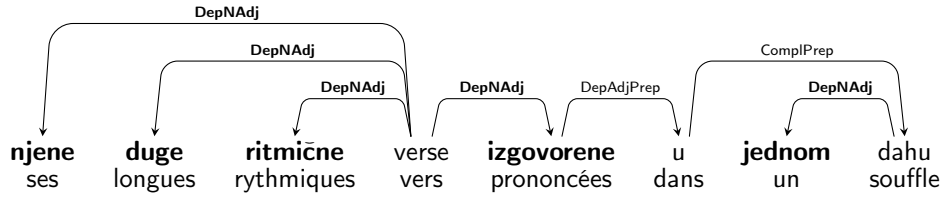
Les effets globaux de la DLM ont été observés sur les treebanks de 37 langues différentes (Futrell et al., 2015). (Gulordava & Merlo, 2015) et (Gulordava et al., 2015) se concentrent sur son effet à l'intérieur du GN et examinent le positionnement de l'adjectif dans 5 langues romanes (catalan, espagnol, français, italien et portugais). Leurs résultats montrent que le comportement observé des GA est majoritairement en accord avec les prédictions formulées suivant le principe de la DLM : la présence d'un dépendant à droite de l'adjectif favorise la postposition de l'adjectif au nom, ce qui correspond au principe d'optimisation de l'exemple 25b, alors que la présence d'un autre dépendant à droite du nom favorise l'antéposition de l'adjectif, ce qui correspond à la situation illustrée dans l'exemple 25e. Cependant, la présence d'un dépendant à gauche de l'adjectif présente un comportement plus complexe : si ce dépendant adjectival est de longueur 1 (il comporte un seul mot), il favorise l'antéposition de l'adjectif, ce qui est en accord avec le principe de la DLM. En revanche, si ce dépendant est constitué de plusieurs mots, la postposition est préférée ; cette situation ne suit donc pas le principe de la DLM. On remarque que ce dernier cas de figure pourrait plutôt refléter le principe du poids, avec un GA lourd qui se positionne après son gouverneur. Néanmoins, le travail de Gulordava et al. (2015) montre que la DLM est un meilleur prédicteur de la position de l'adjectif dans les langues romanes que le poids syntaxique.

Compte tenu de ce qui a été présenté ci-dessus, nous formulons les objectifs de cette étude. Le premier découle de l'état des descriptions existantes du comportement syntaxique des adjectifs en serbe : nous souhaitons compléter ces bilans par un examen basé sur une approche empirique à partir du corpus ParCoTrain-Synt. Autrement dit, nous cherchons à dresser un inventaire des cas de figure attestés en corpus et à dégager les conditions de leur réalisation, tout en nous appuyant sur une analyse quantitative. Dans un deuxième temps, nous considérons les principes du poids syntaxique et de la DLM : étant donné leur utilité dans l'analyse de ce phénomène en langues romanes, nous évaluons leur capacité à rendre compte de la position de l'adjectif en serbe. Ce travail est donc innovant selon plusieurs points de vue : il s'agit d'une première étude sur corpus de la position de l'adjectif en serbe ; c'est également la première tentative d'aborder cette question du point de vue syntaxique ; enfin, il s'agit aussi d'une première étude des effets du poids syntaxique et de la DLM sur le serbe.

10.2 Extraction des données du corpus ParCoTrain-Synt : focus sur le contexte syntaxique

Cette étude est faite à partir d'une extraction de données du corpus ParCoTrain-Synt. Dans ce travail, nous avons utilisé la première version complète du corpus de 101 000 tokens, décrite dans la section 8.8. Comme évoqué dans la section 5.2.7, selon notre schéma d'annotation, tout dépendant d'un nom sous forme d'un adjectif porte l'étiquette syntaxique DepNAdj. Ceci est fait indépendamment de la sous-catégorie de l'adjectif et de sa position par rapport au nom (cf. exemple 26).

(26)



'ses longs vers rythmiques prononcés en un souffle'

Avant de poursuivre, rappelons deux faits relatifs au traitement des adjectifs dans ParCoTrain-Synt. Premièrement, les formes déverbales dérivées par conversion d'un participe (cf. *izgovorene* 'prononcées' dans l'exemple 26) sont considérées comme des adjectifs si elles sont gouvernées par un nom. Deuxièmement, les formes des possessifs, des indéfinis, des interrogatifs, des relatifs et des démonstratifs gouvernés par un nom sont considérées comme des adjectifs (cf. *njene* 'ses' dans le même exemple). La motivation de ces décisions est décrite en détail dans la section 5.1.2.

Il faut également préciser que nous limitons notre analyse aux occurrences de DepNAdj gouverné par un nom et laissons de côté les têtes pronominales. Cette décision est due au fait que le positionnement canonique de l'adjectif avec les pronoms indéfinis est à droite de la tête (cf. *neko* ***lep*** lit. 'quelqu'un **beau**', 'quelqu'un de beau' ; *ništa* ***zanimljivo*** lit. 'rien **intéressant**', 'rien d'intéressant'), ce qui diffère du positionnement canonique de l'adjectif par rapport au nom. Nous considérons donc qu'il s'agit d'un phénomène syntaxique à part.

Cette étude est basée sur l'examen du contexte syntaxique de l'adjectif aussi bien au niveau du GA que du GN. Cela consiste à analyser différents types de dépendants de l'adjectif et du nom qui le gouverne, ainsi que la coordination. La liste complète des propriétés relevées est donnée dans la figure 10.3. Pour chacun des dépendants, nous notons non seulement sa présence mais aussi sa position par rapport à son gouverneur. La première caractéristique est indicative du poids du groupe dont le dépendant fait partie, alors que la deuxième permet d'examiner la structure du groupe par rapport au principe de la DLM. L'extraction des données du corpus ainsi que le recensement des propriétés

listées sont faites de manière automatique, à partir de l'annotation syntaxique du corpus. En revanche, les principes du poids syntaxique et de la DLM sont observés dans le cadre d'une analyse manuelle : nous examinons leur effet en les appliquant aux différents cas de figure identifiés.

- Structure du GA :
  - GA minimal (formé d'un seul adjectif) (*Adj_nu*)
  - Coordination (*Adj_Coord*)
  - Dépendant sous forme d'un adverbe (*DepAdjAdv*)
  - Dépendant sous forme d'un GN à cas oblique (*DepAdjCas*)
  - Dépendant sous forme d'un GP (*DepAdjPrep*)
- Structure du GN :
  - N avec le GA analysé comme seul dépendant (*N+GA_seul*)
  - N avec le GA analysé comme seul dépendant, et le GA est minimal (*N+Adj_nu_seul*)
  - Coordination (*N_Coord*)
  - Apposition (*Ap*)
  - Autre dépendant sous forme d'un GA (*DepNAdj*)
  - Dépendant sous forme d'un GN à cas oblique (*DepNCas*)
  - Dépendant sous forme d'un GP (*DepNPrep*)
  - Dépendant sous forme d'une subordonnée (*N+Sub*)

FIGURE 10.3 – Liste de propriétés lexicales et syntaxiques extraites du corpus

Nous précisons que dans le cas d'un GA contenant une coordination nous ne considérons que l'occurrence du premier adjectif. Autrement dit, nous n'analysons pas chaque adjectif coordonné à son tour. Cette démarche est biaisée : dans le cas d'un GA avec trois adjectifs coordonnés, nous ne marquons qu'une occurrence de la coordination, alors que l'approche inverse en compterait trois. Cependant, ce choix reflète notre décision de nous intéresser au GA en tant qu'unité globale dans un premier temps. Cet aspect particulier pourra être approfondi dans la suite de ce travail.

Afin d'obtenir une première estimation de l'intérêt des propriétés relevées, nous avons évalué l'association statistique de chacun des paramètres observés avec la position de l'adjectif à l'aide du test χ^2. Toutes les propriétés au niveau du groupe adjectival ont montré une association statistiquement très significative avec la position de l'adjectif ($p<0{,}001$; cf. section 10.3 pour les résultats individuels détaillés). En revanche, au niveau du groupe nominal, les seules propriétés statistiquement associées avec la position de l'adjectif étaient N+GA_seul et N+Adj_nu_seul. La suite de cette étude est orientée en fonction de ces résultats : nous analysons toutes les propriétés du GA, mais ne retenons que N+GA_seul et N+Adj_nu_seul au niveau du GN. L'examen détaillé des données sur la structure du GN, quoiqu'intéressant du point de vue qualitatif, n'a pas été réalisé dans le cadre de cette thèse.

10.3 Positionnement et propriétés combinatoires des adjectifs en fonction de leur sous-catégorie

Notre méthode d'extraction identifie au total 9666 occurrences de la relation DepNAdj qui correspondent aux critères posés. Un aperçu global de la distribution des occurrences en fonction de la position de l'adjectif et de sa sous-catégorie est donné dans le tableau 10.1. Comme mentionné ci-dessus, environ 6 % de tous les cas repérés se trouvent à droite du nom. Cependant, la distribution n'est pas identique pour toutes les sous-catégories morphosyntaxiques.

Deux types de comportements se dégagent de ces données : tandis que plus de 9 % d'occurrences des adjectifs qualificatifs se trouvent en postposition, les autres sous-catégories n'admettent quasiment pas la postposition. En effet, les démonstratifs, les indéfinis, les interrogatifs et les relatifs ne manifestent aucune mobilité, les possessifs étant les seuls à avoir été repérés à droite de leur tête nominale. Leur taux de postposition reste néanmoins très faible (1,02 %). Le test χ^2 montre qu'il existe une association statistiquement très significative entre la sous-catégorie de l'adjectif et sa position (χ^2=314,6, p<0,001) [5].

Sous-catégorie	Total	Antéposés		Postposés	
		N	%	N	%
Toutes sous-cat.	9666	9111	94,26	555	5,74
Démonstratifs	855	855	100,00	0	0,00
Indéfinis	677	677	100,00	0	0,00
Interrogatifs	18	18	100,00	0	0,00
Possessifs	2245	2222	98,98	**23**	**1,02**
Relatifs	56	56	100,00	0	0,00
Qualificatifs	5819	5287	90,86	**532**	**9,14**

TABLE 10.1 – Adjectifs en antéposition *vs* en postposition en fonction de sous-catégorie

Le comportement des différentes sous-catégories se distingue également au niveau des tendances combinatoires au sein du GA et du GN (cf. tableau 10.2). Les interrogatifs et les relatifs n'apparaissent jamais avec des dépendants dans notre corpus. La grande majorité des démonstratifs, des possessifs et des indéfinis exhibent le même comportement. Les quelques dépendants repérés sont de nature adverbiale dans le cas des possessifs et des indéfinis, (cf. ***apsolutno*** *nikakav* '**absolument** aucun', ***i*** *moja majka* lit. 'aussi ma mère', 'ma mère aussi'), tandis que dans le cas des démonstratifs, il s'agit des corrélatifs introduisant des consécutives (cf. *s* ***takvom*** *tačnošću* ***da*** lit 'avec **telle** exactitude **que**', 'avec une telle exactitude que'). Les qualificatifs ont une tendance moins prononcée à

5. Tous les tests χ^2 rapportés dans ce chapitre ont été effectués à l'aide du programme R. Pour les tests dans lesquels le nombre d'occurrences était bas, la simulation Monte Carlo a été utilisée. Dans ce cas, les degrés de liberté ne sont pas indiqués, vu qu'il ne sont pas pertinents dans ce cadre.

apparaître seuls, et leurs dépendants sont les plus diversifiés : il s'agit aussi bien de noms à cas oblique et de GP que d'adverbes et de subordonnées (cf. section 10.5).

Quand on considère la structure du GN (cf. tableau 10.2), on constate qu'une portion importante des relatifs et des interrogatifs sont le seul dépendant de leur tête (N+GA_seul >75 %). C'est moins souvent le cas avec les possessifs (N+GA_seul <60 %), alors que les indéfinis et les démonstratifs se rapprochent sur ce point des qualificatifs (N+GA_seul entre 32 % et 41 %). On observe des tendances comparables pour les cas où l'adjectif et le nom sont les seuls éléments du GN (cf. N+Adj_nu_seul). Pour les trois propriétés combinatoires, il existe une association statistiquement très significative avec la sous-catégorie de l'adjectif[6].

Sous-catégorie	Total	Adj_nu		N+GA_seul		N+Adj_nu_seul	
		N	%	N	%	N	%
Interrogatifs	18	18	100,00	15	83,33	15	83,33
Relatifs	56	56	100,00	43	76,79	43	76,79
Démonstratifs	855	847	99,06	353	41,29	348	40,70
Indéfinis	673	664	98,66	217	32,24	212	31,50
Possessifs	2245	2236	99,60	1343	59,82	1336	59,51
Qualificatifs	5819	4785	82,23	2167	37,24	1883	32,36

TABLE 10.2 – Combinatoire globale des adjectifs au niveau du GA et du GN. Adj_nu = adjectif sans dépendant, N+GA_seul = nom sans dépendant, N+Adj_nu_seul = Adj sans dépendant est le seul dépendant du N.

Il est intéressant de noter que les différentes sous-catégories non qualificatives montrent un degré de cohésion important. En répétant les mêmes tests statistiques après avoir fusionné tous les non-qualificatifs en une seule catégorie, nous avons trouvé une association statistiquement significative aussi bien avec la position de l'adjectif qu'avec les propriétés combinatoires ($p<0{,}001$). Ceci confirme que ces sous-catégories ont des propriétés syntaxiques communes, qui se démarquent par rapport à celles des qualificatifs. Toutefois, l'une d'entre elles peut nous renseigner sur la variation de la position de l'adjectif : celle des possessifs. Par conséquent, la suite de ce chapitre s'organisera autour de l'analyse du comportement des possessifs et des qualificatifs.

10.4 Adjectifs possessifs

Comme constaté à partir du tableau 10.1, les possessifs sont les seuls parmi les non-qualificatifs à avoir été repérés en postposition dans notre corpus. Bien que le nombre

6. Adj_nu : χ^2=693,63, p<0,001 ; N+GA_seul : χ^2=409,08, df=5, p<0,001 ; N+Adj_nu_seul : χ^2=564,77, df=5, p<0,001).

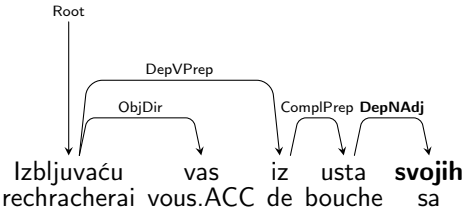
d'occurrences extraites soit bas (23 occurrences, 1,02 % de toutes les occurrences de postposition adjectivale), un examen des exemples permet d'identifier deux cas de figure spécifiques.

Le premier cas de figure représente une simple postposition du possessif à son nom gouverneur (cf. exemples 27a et 27b). Rien n'interdit théoriquement aux indéfinis et aux démonstratifs d'apparaître dans le même type de construction (cf. exemples 27c et 27d). L'apparente immobilité de ces sous-catégories pourrait donc être due à la taille et à la diversité relativement limitées de notre corpus. En revanche, une configuration comparable ne semble pas possible pour les interrogatifs et les relatifs : les locuteurs natifs consultés n'arrivent pas à en produire d'exemples. Il est donc possible qu'il s'agisse ici d'une contrainte absolue, qui peut s'expliquer par des contraintes syntaxiques plus globales, notamment par les effets du *wh-fronting*, qui exige que les formes de ce type occupent la position initiale dans la proposition.

Ce cas de figure ne couvre que 10 occurrences relevées. Dans les 13 occurrences restantes, le GN (et par conséquent le possessif postposé) se trouve au vocatif (cf. exemple 28). Si l'on rétablit l'ordre canonique dans ces exemples (cf. ***moj*** *mladiću*, ***moj*** *Bože*), les phrases résultantes sont moins naturelles, mais ne sont pas agrammaticales. Il est donc difficile de se prononcer sur le statut obligatoire ou non de cette permutation. Le maintien de l'ordre canonique au vocatif semble par ailleurs facilité par la présence d'une interjection d'apostrophe, cf. *E, moj Filipe* 'Eh mon.VOC Filip.VOC'. Notons néanmoins que dans le corpus, il existe seulement 2 occurrences d'un possessif au vocatif qui ne sont pas postposées. L'une d'entre elles représente le syntagme ***vaše*** *visočanstvo* '**votre** majesté' : ici, le possessif reste dans sa position initiale malgré le vocatif, mais il s'agit d'une expression figée, toujours utilisée dans cet ordre. Dans l'autre cas relevé, le possessif est accompagné d'un autre DepNAdj antéposé, avec lequel il a permuté : *draga* ***moja*** *Maruseta* lit. 'chère ma Maruseta' (cf. l'ordre canonique au nominatif ***moja*** *draga Maruseta*). Ces données indiquent donc que dans un GN au vocatif le possessif ne garde typiquement pas sa position initiale : s'il est le seul adjectif à accompagner le nom, il bascule en postposition ; s'il est accompagné d'un autre adjectif antéposé, il change de place avec celui-ci. Ces exemples confirment les observations de Mrazović (2009), évoquées dans la section 10.1.

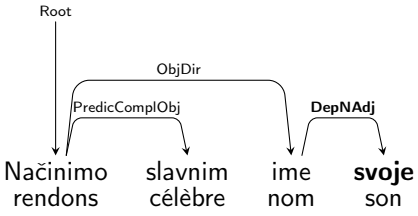
Le rôle du poids syntaxique et de la DLM dans la position des possessifs semblent incertains. Il est vrai que la position canonique des possessifs est en accord avec le principe du poids syntaxique : en tant que GA le plus souvent minimaux, ces éléments sont légers et se positionnent de manière naturelle devant leur tête. Leur postposition va donc à l'encontre de ce principe. Quant à la longueur des dépendances, au niveau du GA la postposition ne change rien, mais au niveau du GN deux effets sont possibles : dans les exemples 27a, 27b et 28a, la postposition minimise la longueur de la dépendance du nom, vu que le possessif ne s'interpose pas entre le nom et son gouverneur ; ceci n'est pas le cas

(27) a.



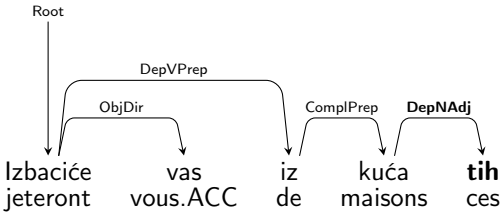
'Je vous recracherai de ma bouche.'

b.



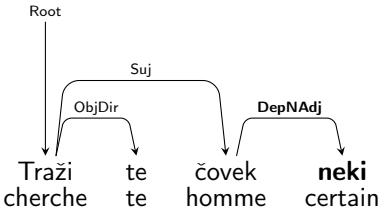
'Rendons célèbre notre nom.'

c.



'On vous jetera dehors de ces maisons.'

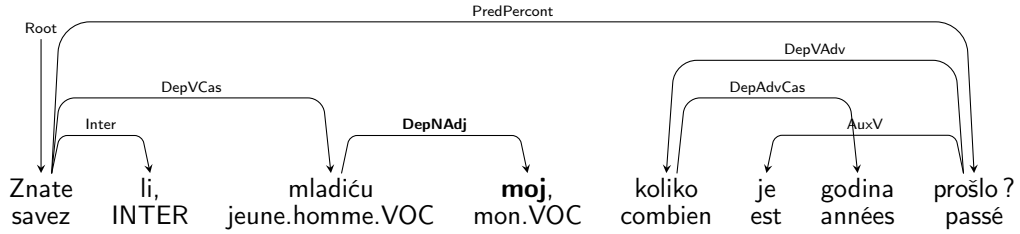
d.



'Un homme te cherche/Il y a un homme qui te cherche.'

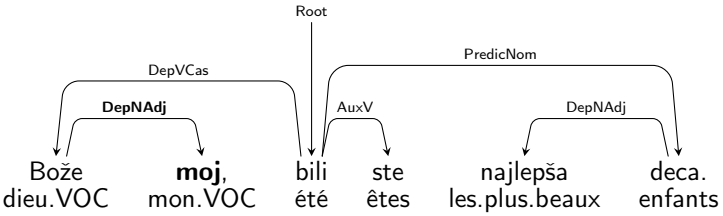
dans l'exemple 28b, où la postposition rallonge en effet la dépendance du nom, étant donné que le possessif s'insère entre le nom et son gouverneur. Donc, même si on observe souvent une minimisation de la longueur des dépendances, elle n'est pas systématique et dépend d'un contexte plus large. Il faut également considérer le fait que dans les exemples 27a et 27b la postposition est ressentie comme marquée, l'ordre non marqué étant l'antéposition, cf. *Izbljuvaću vas iz* ***svojih*** *usta*, *Učinimo slavnim* ***svoje*** *ime*. De fait, les exemples comme 27a et 27b peuvent être qualifiés comme appartenant à un langage poétique et semblent relever d'un choix stylistique de l'auteur. Si la DLM avait un effet prépondérant sur le positionnement des possessifs, on s'attendrait à ce que l'antéposition soit défavorisée, étant donné qu'elle rallonge la dépendance reliant le nom à son gouverneur. Ils représentent

(28) a.



'Savez-vous, mon jeune homme, combien d'années est passé ?'

b.



'Mon Dieu, vous étiez les enfants les plus beaux.'

cependant le cas de figure majoritaire. Un autre indicateur syntaxique potentiellement intéressant a néanmoins été repéré : dans 7 cas sur 10 de ce type de postposition, l'adjectif est le seul dépendant du nom. L'examen d'un nombre d'occurrences plus important est cependant nécessaire pour évaluer l'importance de ce facteur.

10.5 Adjectifs qualificatifs

Comme nous l'avons vu dans la section 10.3, les qualificatifs sont la sous-catégorie d'adjectifs la plus propice à se retrouver en postposition et à apparaître dans des structures complexes, aussi bien au niveau du GA que du GN. L'analyse proposée ici s'intéressera particulièrement aux interactions entre ces deux caractéristiques : nous croiserons les données sur le positionnement du qualificatif avec celles relatives à son entourage syntaxique. Nous examinons d'abord le lien entre la structure globale du GA et du GN et le positionnement de l'adjectif (cf. section 10.5.1) ; nous examinons ensuite en détail le comportement de l'adjectif doté d'un dépendant adverbial (cf. 10.5.3) et ensuite celui de l'adjectif accompagné d'un dépendant casuel ou prépositionnel (cf. section 10.5.2).

10.5.1 Observations globales sur la combinatoire des adjectifs qualificatifs

Comme mentionné ci-dessus, les adjectifs qualificatifs accueillent proportionnellement le plus de dépendants et ces dépendants sont les plus diversifiés par rapport aux autres sous-catégories adjectivales. Cette tendance ne se réalise cependant pas de la même manière

en antéposition et en postposition. Les données dans le tableau 10.3 montrent la fréquence de différentes configurations du GA et du GN en fonction de la position de l'adjectif. Tandis que la configuration N+GA_seul s'équilibre à peu près entre les GA antéposés et les GA postposés, ceci n'est pas le cas pour les deux autres paramètres observés. La proportion de GA minimaux (Adj_nu) est quasiment inversée entre l'antéposition et la postposition : parmi les adjectifs antéposés, 89 % sont sans dépendant, alors que parmi les postposés, ce n'est que 8 % qui sont sans dépendant. Par ailleurs, pour seulement 4 % des postposés, le nom et l'adjectif sont les seuls éléments du GN (N+Adj_nu_seul), alors que parmi les antéposés ce taux est de 35 %. Les trois propriétés combinatoires montrent un degré d'association statistiquement très significatif avec la position de l'adjectif qualificatif [7].

Position	Occur.	Adj_nu		N+GA_seul		N+Adj_nu_seul	
		No.	%	No.	%	No.	%
Antéposés	5287	4741	89,67	2024	38,28	1861	35,20
Postposés	532	44	8,27	143	26,88	22	4,14

TABLE 10.3 – Propriétés combinatoires globales des adjectifs qualificatifs. Adj_nu = adjectif sans dépendant, N+GA_seul = nom sans autre dépendant, N+Adj_nu_seul = Adj sans dépendant est le seul dépendant du N

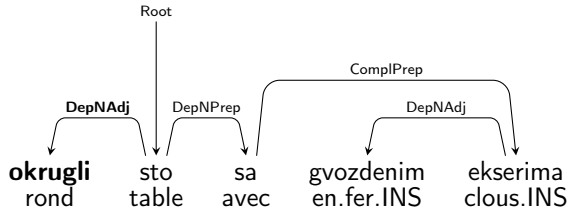
Les cas de figure majoritaires représentés ici respectent le principe général du poids syntaxique : l'antéposition favorise les GA minimaux, qui sont légers, alors que la postposition préfère les GA plus lourds, dans lesquels l'adjectif n'est pas le seul élément. Ces deux cas de figure sont illustrés respectivement dans les exemples 29a et 29b.

En revanche, comme le montrent les résultats du tableau 10.3, des occurrences à l'encontre de ce principe ont également été relevées en corpus : dans l'exemple 29c, on voit un adjectif seul positionné à droite de sa tête, alors que l'exemple 29d illustre la situation dans laquelle un adjectif doté d'un dépendant casuel se trouve en antéposition.

Si l'on considère l'exemple 29c, on constate que ce patron de linéarisation (**Gouv$_{(N)}$ N Adj**) optimise la longueur des dépendances par rapport au patron attendu **Gouv$_{(N)}$ Adj N**. On pourrait donc être tenté de considérer que la postposition d'un GA léger est motivée par le principe de la DLM. Or, rappelons que cette configuration est largement minoritaire : moins de 10 % des GA minimaux dans notre corpus aparaissent en postposition, alors que le patron **Gouv$_{(N)}$ Adj N** est omniprésent (cf. l'adjectif *gvozdenim* dans l'exemple 29a et l'adjectif *naivnih* dans l'exemple 29b). Par ailleurs, l'effet observé ci-dessus dépend entièrement de la position du nom par rapport à son gouverneur : si le nom est à sa gauche, la postposition de l'adjectif n'est pas favorable à la minimisation des dépendances (cf. le patron **N Adj Gouv$_{(N)}$**). La postposition d'un qualificatif léger nous semble donc

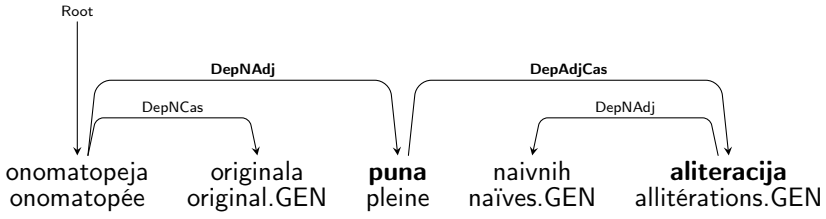
7. Adj_nu : χ^2=2186,4, df=1, p<0,001 ; N+GA_seul : χ^2=26,405, df=1, p<0,001 ; N+Adj_nu_seul : χ^2=211,68, df=1, p<0,001).

(29) a.



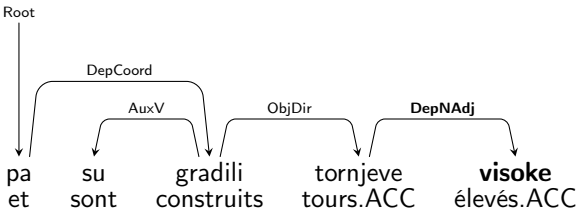
'table ronde avec des clous en fer'

b.



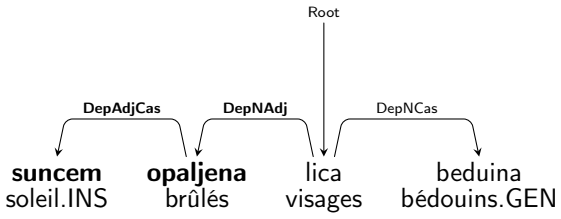
'onomatopée de l'original pleine d'allitérations naïves'

c.



'et ils construisaient des tours élevées'

d.



'visages des bédouins brûlés par le soleil'

proche de la postposition des possessifs (hors cas du vocatif) : elle relèverait plutôt d'un effet de style que d'un principe purement syntaxique.

Remarquons cependant que l'antéposition du GA lourd présente un comportement particulier. Le dépendant casuel de l'adjectif se place typiquement à droite de son gouverneur, dans le patron **Adj DepAdjCas**. Or, dans l'exemple 29d, le DepAdjCas se trouve à gauche de l'adjectif, formant le patron suivant : **DepAdjCas Adj N**. Qui plus est, le patron attendu d'après le positionnement canonique du DepAdjCas, à savoir **Adj DepAdjCas N** n'est pas grammatical : **opaljena* ***suncem*** *lica beduina* 'brûlées soleil.INS visages bédouins.GEN'. Il est important de noter que le patron **DepAdjCas Adj N** permet de minimiser la longueur de la dépendance du DepNAdj par rapport au positionnement at-

tendu **Adj DepAdjCas N** grâce au fait que le DepAdjCas se positionne du côté opposé de l'adjectif par rapport à son gouverneur. Le principe de la DLM semble donc avoir un rôle dans la réalisation de cette configuration.

Plus globalement, les trois types de dépendants syntagmatiques de l'adjectif observés dans notre corpus disposent d'un certain degré de mobilité : 8 % d'occurrences du DepAdjCas sont antéposés, et ce taux est de 8,9 % pour le DepAdjPrep. Le dépendant adverbial de l'adjectif (DepAdjAdv), qui se trouve canoniquement à gauche de son gouverneur, apparaît en postposition dans 5 % des occurrences repérées. L'exemple 29d indique que ce positionnement alternatif des dépendants adjectivaux pourrait être lié au positionnement de l'adjectif lui-même. Afin d'examiner de plus près cette hypothèse, nous poursuivons cette analyse en examinant l'interaction entre le positionnement de l'adjectif et celui de ces dépendants.

10.5.2 Interactions avec les dépendants casuels et prépositionnels de l'adjectif

Les tableaux 10.4 et 10.5 montrent que le dépendant casuel et le dépendant prépositionnel de l'adjectif suivent les mêmes patrons de comportement globaux : ils apparaissent en large majorité avec des qualificatifs postposés, et quand ils sont gouvernés par un adjectif antéposé, ils ont tendance à être antéposés à leur tour. Cette distribution corrobore notre hypothèse que le placement de ces dépendants était lié à la position de l'adjectif lui-même. Ceci est par ailleurs confirmé au niveau statistique : pour les deux types de dépendants, la position de l'adjectif et leur propre position sont associées de manière statistiquement très significative [8].

	DepAdjCas					
	DepAdjCas postposé*		DepAdjCas antéposé		Total	
	No.	%	No.	%	No.	%
Adj antéposé	2	1,16	11	**6,40**	13	7,56
Adj postposé	157	**91,28**	2	1,16	159	92,44
Total	159	92,44	13	7,56	172	100,00

TABLE 10.4 – Distribution des occurrences de DepAdjCas en fonction de sa position. * = position canonique du dépendant.

Les deux organisations majoritaires sont montrées dans les exemples 30a à 30d : la configuration **N Adj DepAdj** est illustrée dans les exemples 30a et 30b, alors que les exemples 30c et 30d correspondent à la structure **DepAdj Adj N**. Il est important de noter que l'ordre **N Adj DepAdj** est l'ordre canonique, alors que l'ordre **DepAdj Adj**

8. DepAdjCas : χ^2=119,51, p<0,001 ; DepAdjPrep : χ^2=78,888, p<0,001.

	DepAdjPrep					
	DepAdjPrep postposé*		DepAdjPrep antéposé		Total	
	No.	%	No.	%	No.	%
Adj antéposé	4	2,16	13	**7,03**	17	9,19
Adj postposé	160	**86,49**	8	4,32	168	90,81
Total	164	88,65	21	11,35	185	100,00

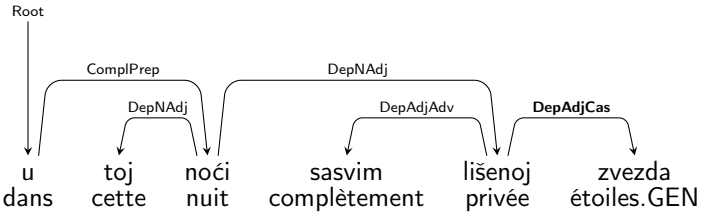
TABLE 10.5 – Distribution des occurrences de DepAdjPrep en fonction de sa position. * = position canonique du dépendant.

N est marqué et semble relever de la topicalisation. Néanmoins, pour les 4 exemples cités, l'ordre alternatif est acceptable (cf., pour l'exemple 30a, *u toj zvezda sasvim lišenoj noći* lit. 'dans cette étoiles.GEN complètement privée nuit' ; pour l'exemple 30c, *cifre ispisane sopstvenom rukom* lit. 'chiffres écrits propre.INS main.INS'). Il semble s'agir donc d'un mécanisme de linéarisation régulier. Notons encore que la position du dépendant adjectival est optimale du point de vue de la DLM dans les deux configurations considérées : en se positionnant du côté opposé de l'adjectif par rapport au nom, le dépendant minimise la longueur de la dépendance de l'adjectif.

L'image globale qui se dessine ici est donc la suivante : la position canonique de l'adjectif accompagné d'un dépendant casuel ou prépositionnel est à droite du nom ; cette configuration correspond au principe du poids syntaxique. Cependant, si pour des raisons de topicalisation un GA de ce type se retrouve en antéposition, le principe de la DLM semble imposer une réorganisation du GA qui permet d'optimiser la longueur de la dépendance entre l'adjectif et le nom en remplaçant la configuration attendue **Adj DepAdj N** par la configuration **DepAdj Adj N** relevée en corpus. Les configurations où le dépendant adjectival se trouve entre l'adjectif et le nom, à savoir **Adj DepAdj N** et **N DepAdj Adj**, sont défavorisées, cf. les exemples difficilement acceptables de ?*noć zvezda lišena* lit. 'nuit étoiles.GEN privée' et ?*lišena zvezda noć* lit. 'privée étoiles.GEN nuit'.

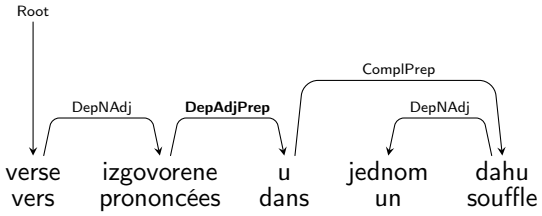
Néanmoins, des exemples de ces structures non optimales ont été relevés en corpus (cf. exemple 31). Comme le nombre total de ces cas de figure reste relativement bas (4 pour DepAdjCas et 12 pour DepAdjPrep), il est difficile d'identifier les principes derrière ces configurations. Ils sont tout de même suffisamment nombreux pour ne pas pouvoir être considérés comme marginaux. Nous en retenons le fait que les principes de linéarisation formulés ci-dessus relèvent plutôt de tendances, quoique prononcées, que de contraintes absolues.

(30) a.



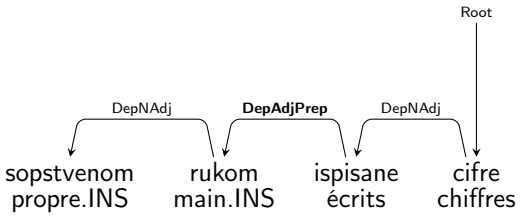
'dans cette nuit complètement privée d'étoiles'

b.



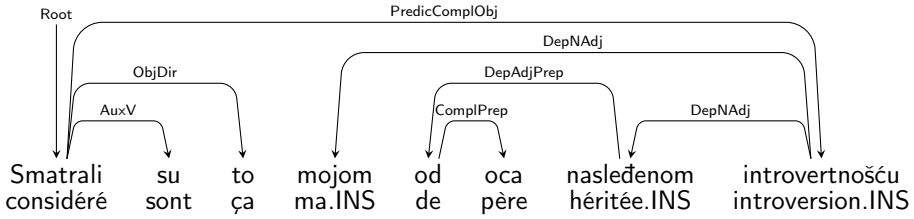
'vers prononcés en un souffle'

c.



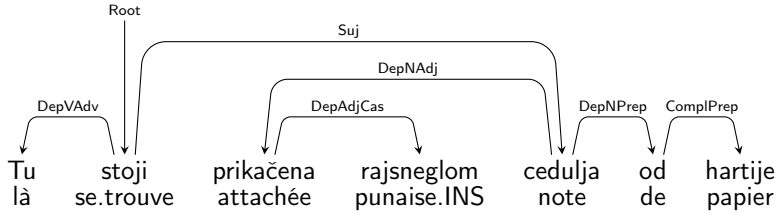
'chiffres écrits de sa propre main'

d.



'Ils prenaient ça pour mon introversion héritée de mon père'

(31)



'Là se trouve une note en papier attachée avec une punaise'

10.5.3 Interactions avec le dépendant adverbial de l'adjectif

La grande majorité des occurrences du dépendant adverbial de l'adjectif apparaissent en antéposition par rapport à leur tête (environ 95 %, cf. tableau 10.6). Ce type de dépendant se montre donc relativement peu susceptible de changer de place, quelle que soit la position du GA par rapport au nom (cf. exemples 32a et 32b).

	DepAdjAdv					
	DepAdjAdv antéposé*		DepAdjAdv postposé		Total	
	No.	%	No.	%	No.	%
Adj antéposé	132	**55,23**	0	0,00	132	55,23
Adj postposé	95	**39,75**	12	5,02	107	44,77
Total	227	94,98	12	5,02	239	100,00

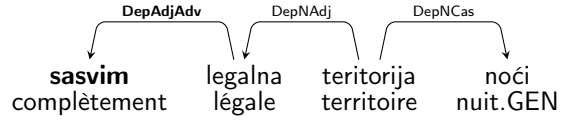
TABLE 10.6 – Distribution des occurrences de DepAdjAdv en fonction de sa position

Le premier constat qui s'impose ici est que le poids syntaxique n'a pas le même effet sur ce type de GA que sur ceux dotés d'un dépendant casuel ou prépositionnel : si leur comportement était identique, on s'attendrait à ce que les GA dotés d'un DepAdjAdv préfèrent également la postposition. Or, le tableau 10.6 montre que l'antéposition l'emporte. À la différence des cas de figure considérés jusqu'ici, le principe du poids syntaxique ne permet donc même pas d'expliquer le comportement majoritaire.

Si l'on considère les deux configurations majoritaires d'après le tableau 10.6, on remarque qu'elles correspondent aux patrons respectifs de **Adv Adj N** et de **N Adv Adj**. Le premier d'entre eux est bien en accord avec le principe de la DLM, et semble par ailleurs être la seule possibilité dans le cas de l'antéposition de l'adjectif d'après nos données (0 occurrence de la configuration **Adj Adv N**). Le patron **N Adv Adj**, en revanche, ne reflète pas le principe de la DLM, vu que l'adverbe est positionné entre l'adjectif et le nom, la configuration optimale de ce point de vue étant **N Adj Adv**. Cette construction très fréquente ne s'explique donc pas par le principe de la DLM.

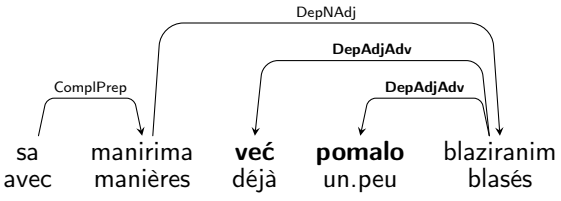
Le patron **N Adj Adv** de façon très minoritaire dans le corpus (12 occurrences). Bien que cette configuration respecte le principe de DLM, d'autres contraintes semblent jouer. 5 occurrences sur 12 correspondent au cas de figure représenté par l'exemple 32c, où la postposition de l'adverbe semble la seule option : ? ?*majka, ruku* ***uvis*** *dignutih* 'mère, mains vers.le.haut levées'. Ce cas de figure est donc bien différent du cas de figure global, où l'antéposition représente l'ordre canonique quelle que soit la position de l'adjectif. Par ailleurs, l'omission de l'adverbe rend ces exemples agrammaticaux, cf. **majka, ruku dignutih* 'mère, mains levées'. Il faut noter également qu'il s'agit des adjectifs dérivés des participes par conversion ; ils régissent donc les adverbes en question par héritage du schéma lexico-syntaxique du verbe duquel ils sont dérivés. La postposition de ces adverbes

(32) a.



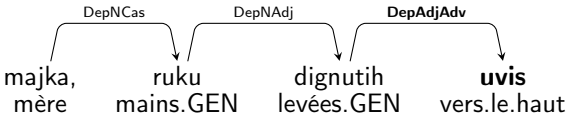
'territoire de la nuit complètement légal'

b.



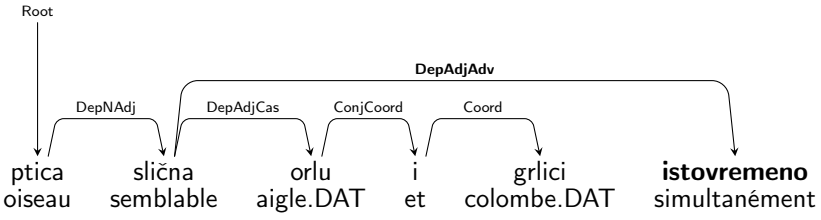
'avec des manières déjà un peu blasées'

c.



'mère, les mains levées vers le haut'

d.



'oiseau semblable à un aigle et à une colombe à la fois'

semble donc dépendre de facteurs qui dépassent la structure du GA et sa position.

Dans les 7 occurrences restantes, dont l'exemple 32d, l'antéposition de l'adverbe à l'adjectif est possible (cf. *ptica istovremeno slična orlu i grlici* 'oiseau simultanément semblable aigle.DAT et colombe.DAT'), et l'omission de l'adverbe ne compromet pas la grammaticalité de la phrase (cf. *ptica slična orlu i grlici*). Ces exemples sont donc plus proches du cas de figure global. La postposition de ces adverbes semble tout de même être en interaction avec un ou plusieurs autres facteurs : l'adjectif est accompagné d'un ou plusieurs autres dépendants, l'adverbe a des dépendants à son tour, l'adjectif ou ses dépendants se trouvent en coordination, un deuxième adjectif est présent à gauche du nom, etc. Il est donc difficile d'affirmer que le positionnement de ces adverbes est conditionné seulement par celui de l'adjectif.

10.6 Discussion des effets observés du poids syntaxique et de la DLM

Comme le montre l'analyse présentée ci-dessus, les effets du poids syntaxique et de la DLM en serbe semblent loin d'être systématiques et stables. Tout d'abord, nous avons constaté que les comportements majoritaires relevés en corpus sont en accord avec le principe du poids : les GA légers, formés d'un seul adjectif, préfèrent l'antéposition, alors que les GA lourds, dotés de dépendants ou d'une coordination, favorisent nettement la postposition. Ces résultats sont en accord avec les principes formulés par Abeillé & Godard (1999), ainsi qu'avec les constats de Thuilier et al. (2012) sur un corpus français et ceux de Gulordava & Merlo (2015) sur des corpus des langues romanes. Or, des exceptions à cette tendance de base ont été identifiées dans nos données. La première relève des GA constitués d'un adjectif et d'un dépendant adverbial : malgré leur poids, ces GA préfèrent nettement l'antéposition. D'autres cas de figure moins systématiques ont également été repérés : on trouve des GA minimaux en postposition, qu'il s'agisse des adjectifs qualificatifs ou des possessifs, et on rencontre également des GA constitués d'un adjectif doté d'un dépendant casuel ou prépositionnel en antéposition. Ces variations vont donc à l'encontre du principe du poids.

Des effets de la DLM ont également été observés sur nos données, mais ils sont encore moins systématiques que ceux du poids. Le seul cas de figure relativement stable concerne les cas d'antéposition d'un adjectif doté d'un dépendant casuel ou prépositionnel. Quoique cette configuration soit motivée par la topicalisation plutôt que par la DLM elle-même, le principe de la DLM semble entrer en jeu ici pour acter la réorganisation du GA : le dépendant de l'adjectif ne se trouve plus dans sa position canonique à droite de l'adjectif, mais en antéposition, ce qui permet de minimiser la longueur de la dépendance entre l'adjectif et son gouverneur à lui. Il ne s'agit tout de même pas d'une contrainte absolue, étant donné que des exemples où ce réordonanncement n'a pas lieu existent également dans notre corpus.

La minimisation de la longueur des dépendances entre l'adjectif et son gouverneur a été repérée dans d'autres cas de figure aussi, notamment lors de la postposition d'un GA minimal sous forme d'un possessif ou d'un qualificatif. Néanmoins, l'effet observé dépend de la position du nom par rapport à son gouverneur, et non pas seulement de la position de l'adjectif par rapport au nom. D'ailleurs, la postposition de ce type de GA étant nettement minoritaire en corpus, l'effet de la DLM serait ici d'une très faible ampleur. Ces observations s'opposent donc aux résultats de Gulordava & Merlo (2015) et Gulordava et al. (2015), qui constatent un effet systématique et prégnant de la DLM sur le positionnement de l'adjectif dans les langues romanes.

Nous constatons donc que ces deux principes sont moins aptes à expliquer la varia-

tion de la position de l'adjectif en serbe qu'en langues romanes. Les variations observées dans notre corpus semblent au moins en partie s'expliquer par des procédés à portée pragmatique, comme la topicalisation. Il est néanmoins fort probable qu'une évaluation systématique de ces deux principes à l'aide d'une analyse automatique, à l'instar de celles effectuées dans Gulordava & Merlo (2015) et Gulordava et al. (2015), nous renseignerait davantage sur la présence et l'intérêt de ces phénomènes en serbe.

10.7 Bilan, conclusions et perspectives

Dans ce chapitre, nous avons présenté les résultats d'une première exploitation du corpus ParCoTrain-Synt dans le but de décrire le positionnement du groupe adjectival en serbe. Nous nous sommes servie d'une extraction automatique de données, couplée avec des analyses quantitatives et un examen qualitatif des exemples repérés. Ceci nous a permis de dresser un bilan initial tout en dégageant des propriétés de ces structures qui ne sont pas évoquées dans leurs descriptions existantes.

En ce qui concerne les principes globaux du poids syntaxique et de la DLM, nous avons constaté que leurs effets respectifs étaient moins généralisés que dans les langues romanes. Si le principe du poids permet d'expliquer les tendances majoritaires que nous avons rencontrées, il ne réussit pas à en expliquer les variations. Les effets de la DLM sont sporadiques, sauf dans un cas de variation de l'ordre de l'adjectif, celle qui implique l'antéposition d'un adjectif doté d'un dépendant casuel ou prépositionnel, dans laquelle la DLM semble exiger une réorganisation du GA. Toutefois, il nous semble plus juste de considérer que la DLM accompagne ce type d'antéposition du GA plutôt qu'elle ne le motive.

Nous avons donc pu observer un certain degré de pertinence de ces principes au niveau local ; il reste à évaluer leur étendue globale en serbe. Par conséquent, nous envisageons de poursuivre ce travail dans le cadre d'une étude des principes de la DLM et du poids syntaxique d'un point de vue global et quantitatif, en partant d'une analyse automatique de la totalité de notre corpus.

Dans un avenir plus immédiat, nous allons exploiter les données collectées sur la structure du GN, jusqu'à présent laissées de côté. Malgré le fait qu'il ne semble pas y avoir une association statistiquement significative entre les éléments du GN et la position du GA, un examen plus approfondi pourrait apporter des éclairages intéressants. Nous allons également modifier l'extraction et l'analyse des structures coordonnées afin de mieux rendre compte de l'ampleur de cette structure dans les GA. Enfin, nous nous intéresserons aussi aux paramètres lexicaux. Comme le montrent Thuilier et al. (2012) et Thuilier (2012), une combinaison d'informations lexicales et de données de fréquence sont utiles à la prédiction

de la position de l'adjectif en français. Un prolongement immédiat de cette étude visera l'évaluation de ces facteurs en serbe.

Plus globalement, ce travail nous a permis d'établir une description du comportement de l'adjectif au sein du GN en serbe qui prend en compte la capacité de l'adjectif à être postposé et qui examine l'interaction entre la position du GA et sa structure. Nous avons pu déterminer la fréquence relative de l'antéposition et de la postposition du GA, identifier les comportements majoritaires et les exceptions à ces comportements, et décrire les cas de figure identifiés. Nous avons ainsi obtenu une image du comportement du GA plus complète que celles disponibles dans les grammaires existantes du serbe. Ce travail montre donc comment l'exploitation d'un corpus annoté au niveau syntaxique peut changer la perception d'un phénomène linguistique et, donnant l'accès aux données, faciliter sa caractérisation.

Chapitre 11

Non-projectivité en serbe : analyse de propriétés formelles et linguistiques

Dans ce dernier chapitre, nous montrons comment la disponibilité du corpus ParCoTrain-Synt rend possible une étude de la non-projectivité en serbe. La non-projectivité désigne les structures syntaxiques dans lesquelles un dépendant est séparé de son gouverneur par un élément d'un autre sous-arbre, créant ce qu'on appelle aussi des discontinuités ou des constituants discontinus. Les structures non projectives identifiées dans différentes langues se distinguent par leurs caractéristiques formelles aussi bien que linguistiques. Du point de vue formel, la complexité des structures non projectives est quantifiée à travers, par exemple, le nombre de dépendances non projectives au sein du même sous-arbre syntaxique, ou bien le nombre de sous-arbres disjoints qui participent à la création d'une dépendance non projective. Du point de vue linguistique, l'analyse des constructions sous-jacentes à la non-projectivité dans différentes langues mène à des observations utiles à la typologie des langues et à la linguistique contrastive. Par ailleurs, la non-projectivité est un phénomène qui intéresse également le TAL : ces structures augmentent la complexité du parsing et ne peuvent être traitées que par les parsers par graphes ou par les parsers par transitions dotés des extensions dédiées à la maîtrise des dépendances non projectives.

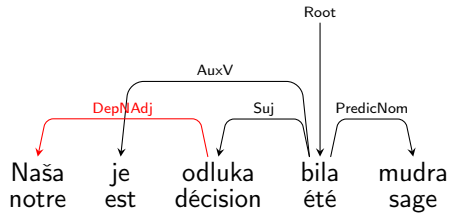
Pour toutes ces raisons, la non-projectivité est étudiée sur un ensemble de langues croissant. L'objectif de ce travail est donc de positionner le serbe par rapport à ces langues à travers une analyse contrastive avec les travaux existants. Nous apportons de nouvelles observations au niveau de la syntaxe théorique du serbe et cherchons également à dégager des informations utiles au traitement automatique de cette langue. Cet examen de la non-projectivité est abordé selon trois angles différents : nous dressons un profil des propriétés formelles des structures non projectives trouvées dans le corpus ParCoTrain-Synt, ensuite,

nous analysons les structures linguistiques sous-jacentes et enfin, nous comparons la maîtrise de ces structures par deux parsers différents, l'un basé sur les graphes, et l'autre sur les transitions.

11.1 Intérêt des constructions non projectives pour la syntaxe théorique et pour le parsing

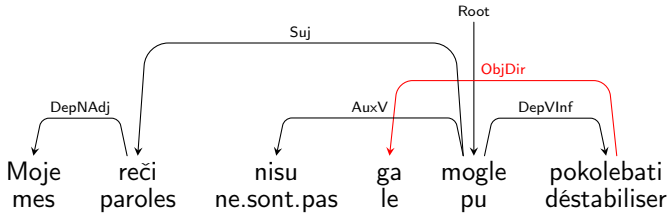
Ce chapitre est dédié à une analyse des propriétés formelles et linguistiques des structures non projectives en serbe. Le terme de non-projectivité désigne les structures syntaxiques dans lesquelles un dépendant est séparé de son gouverneur par un élément d'un autre sous-arbre, ce qui se traduit typiquement par un croisement des arcs dans un arbre en dépendances. Si l'on observe les exemples 33a et 33b, chacun d'entre eux contient une dépendance non projective, dessinée en rouge. Dans 33a, le dépendant *naša* 'notre' est séparé de son gouverneur *odluka* 'décision' par l'auxiliaire *je* 'est', qui n'appartient pas au sous-arbre du gouverneur *odluka*. D'une manière analogue, dans 33b, le dépendant *ga* 'le' est séparé de son gouverneur *pokolebati* 'faire hésiter' par la forme *mogle* 'pu', qui n'est pas dans le sous-arbre de *pokolebati*.

(33) a.



'Notre décision était sage.'

b.



'Mes paroles n'ont pas pu le faire hésiter.'

Typiquement, les langues à morphologie flexionnelle riche et à ordre des constituants flexible ont tendance à avoir plus de structures non projectives. À titre d'illustration, Havelka (2007) observe que l'espagnol exhibe 0,07 % de dépendances non projectives réparties sur 1,72 % de phrases, alors que pour le tchèque, ces valeurs sont respectivement de 2,13 % et de 23,15 %. Une autre langue typologiquement proche du serbe s'est montrée relativement riche en structures non projectives dans le cadre de ce travail : le slovène

exhibe plus de 2 % de dépendances non projectives réparties sur plus de 20 % des phrases. Il est donc justifié de s'attendre à ce que le serbe fournisse des données intéressantes sur ce phénomène.

Comme nous l'avons dit plus haut, la non-projectivité intéresse aussi bien la linguistique théorique que le TAL. Du point de vue de la syntaxe théorique, la non-projectivité est étudiée dans le cadre de la syntaxe en constituants comme dans celui de la syntaxe en dépendances. La syntaxe en constituants identifie ce phénomène sous le nom de structures discontinues et l'explique par les notions de mouvement et de traces dans les approches transformationnelles (cf. Chomsky, 1995, 1993), ou par les mécanismes de passage des traits dans les théories non transformationnelles (cf. Gazdar et al., 1985 ; Pollard & Sag, 1994 ; Bresnan, 2001). Dans le cadre de la syntaxe en dépendances, elle est analysée comme un phénomène d'élévation (angl. *rising* (Groß & Osborne, 2009)), d'émancipation (Gerdes & Kahane, 2001) ou de montée (angl. *climbing* (Duchier & Debusmann, 2001)). En ce qui concerne le TAL, le traitement de la non-projectivité augmente la complexité temporelle de la tâche de parsing et ne peut par ailleurs pas être effectué par les parsers par transitions de base, ce qui a mené à la mise en place de différentes stratégies pour pallier cette lacune (cf. section 3.4.2 pour une description détaillée).

Pour toutes ces raisons, la non-projectivité a déjà été examinée dans plusieurs langues, et notamment en tchèque par Hajičová et al. (2004), en tchèque et en danois par Kuhlmann & Nivre (2006), en arabe, bulgare, tchèque, danois, néerlandais, allemand, japonais, portugais, slovène, espagnol, suédois et turc par Havelka (2007), en hindi, bengali, ourdou et télougou par Bhat & Sharma (2012b), en grec ancien par Mambrini & Passarotti (2013). Dans ces travaux, différents points de vue sont adoptés. Par exemple, Kuhlmann & Nivre (2006) et Havelka (2007) se focalisent sur les propriétés formelles des dépendances et des arbres non projectifs, comme la propriété de *well-nestedness* (emboîtement), le *maximum edge degree* (degré maximal de l'arête) et le *maximum gap degree* (degré maximal de hiatus) (cf. section 11.4.1 pour une définition de ces notions). Des travaux ont également cherché à identifier les structures linguistiques qui donnent lieu à la non-projectivité, notamment Hajičová et al. (2004) pour le tchèque, Bhat & Sharma (2012b) pour le hindi, l'ourdou et le bengali, et Mambrini & Passarotti (2013) pour le grec ancien. L'analyse de ces deux types de propriétés – formelles et linguistiques – facilite différentes comparaisons entre les langues. À titre d'exemple, les expériences de Havelka (2007) confirment les conclusions de Kuhlmann & Nivre (2006) sur le fait que les structures non projectives les plus fréquentes sont en général simples et qu'elles respectent très majoritairement la contrainte de *well-nestedness*. Du point de vue linguistique, Mambrini & Passarotti (2013) soulignent le rôle des clitiques dans les structures non projectives en grec ancien : à ces formes sont attribués plus de 40 % des cas de la non-projectivité. Les enclitiques [1] en

1. Clitiques qui forment une unité prosodique avec la forme pleine qui les précède. À différencier

serbe partagent le comportement des clitiques en grec ancien : ils suivent la loi de Wackernagel (Wackernagel, 1892 ; Ruijgh, 1990), selon laquelle les enclitiques ont tendance à occuper la 2^{e} position dans la proposition, cf. l'exemple 33a, où la forme de l'auxiliaire est un enclitique. On peut donc s'attendre à observer des effets comparables sur le serbe. Un autre exemple concerne le fait qu'en tchèque aussi bien qu'en hindi la non-projectivité peut être causée par les dépendants d'un infinitif d'une construction à contrôle qui se trouvent en dehors de leur proposition. La même structure est possible en serbe (cf. exemple 33b). Nous abordons donc la présente analyse avec l'objectif de faire des parallèles entre le serbe et d'autres langues, ce qui présente un intérêt aussi bien pour le traitement automatique (pour identifier les outils et ressources les mieux adaptés aux langues en question) que pour la typologie linguistique (pour recenser les types de structures syntaxiques non projectives représentées dans ces langues).

Nous nous proposons donc d'établir un double profil de la non-projectivité en serbe, en examinant d'abord ses propriétés formelles puis linguistiques. Nous nous intéressons également à la capacité de deux parsers différents à maîtriser ces structures : nous comparons les performances d'un parser basé sur les transitions doté d'un outil de parsing pseudo-projectif (Talismane de Urieli (2013)) et celles d'un parser basé sur les graphes, capable de traiter la non-projectivité grâce à la nature de son algorithme (MST de McDonald et al. (2006)). Le reste de ce chapitre est organisé comme suit : tout d'abord, nous présentons brièvement notre corpus de travail (section 11.2) et rappelons les particularités du schéma d'annotation pertinentes pour ce travail (section 11.3). La section 11.4 est dédiée à l'analyse des propriétés formelles de la non-projectivité dans le corpus, alors que la section 11.5 propose une analyse linguistique des structures qui mènent à la non-projectivité. Ensuite, la section 11.6 examine les performances des parsers sélectionnés sur les structures non projectives présentes dans le corpus, avec une attention spéciale accordée à l'examen de l'annotation produite par les outils et à l'identification des constructions les plus problématiques. Enfin, nous proposons nos conclusions et perspectives pour la suite de ce travail dans la section 11.7.

11.2 Corpus de travail : 81 000 tokens de ParCoTrain-Synt

Comme annoncé ci-dessus, cette étude est basée sur une analyse de données annotées manuellement. Tout comme dans le chapitre 9, nous nous appuyons ici sur les quatre

des proclitiques, qui forment une unité prosodique avec la forme pleine qui les suit dans la phrase. En serbe, certaines formes des auxiliaires et les formes fléchies des pronoms personnels, ainsi que la particule d'interrogation *li* sont des enclitiques, alors que les prépositions, les conjonctions et la particule de négation *ne* sont des proclitiques (Mrazović, 2009, p. 55). Par exemple, dans la phrase *Filip* ***mi ga je*** *doneo* 'Filip me l'a apporté', le *cluster* des enclitiques forme une unité prosodique avec le sujet *Filip*, alors que dans la phrase *Filip radi* ***u*** *Beogradu* 'Filip travaille à Belgrade' la préposition *u* 'dans' forme une unité prosodique avec son complément *Beogradu*.

premiers échantillons du corpus ParCoTrain-Synt. Notre corpus de travail contient donc environ 81 000 tokens dotés d'annotations manuelles aux niveaux morphosyntaxique et syntaxique. Nous rappelons ici les caractéristiques principales du corpus par rapport à la présente étude (cf. tableau 11.1).

Tokens	Phrases	Formes fléchies	Lemmes	Étiq. POS	Étiq. détail.	Étiq. synt.
81 204	2949	19 681	10 223	15	647	67 (50+17)
$\text{Long}_{(\text{ph})}$	$\text{Prof}_{(\text{a})}$	Dép. longues	Arbres non proj.		Arcs non proj.	
27,53	7,23	5,78 %	503 (17,06 %)		607 (0,75 %)	

Table 11.1 – Aperçu du corpus de travail

Comme nous l'avons déjà observé dans le chapitre 9, ce corpus contient 15 étiquettes POS différentes, 647 étiquettes morphosyntaxiques détaillées et 67 étiquettes syntaxiques (dont 17 dédiées à l'ellipse). La longueur de phrase moyenne ($\text{Long}_{(\text{ph})}$) est relativement élevée (27,53 tokens), et c'est également le cas de la profondeur d'arbre maximale moyenne ($\text{Prof}_{(\text{a})}$). Notons encore que plus de 5 % de dépendances dans le corpus sont longues (distance gouverneur-dépendant > 6 tokens), et que le corpus contient 0,75 % de dépendances non projectives réparties dans 17 % de phrases.

11.3 Effets du schéma d'annotation sur la représentation de la non-projectivité en corpus

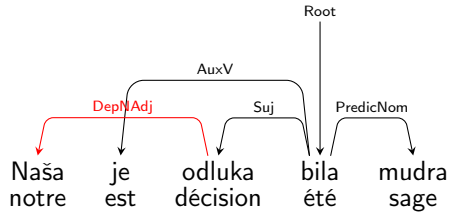
Avant d'entrer dans les détails de ce travail, il est important de préciser que les résultats d'une analyse comme celle-ci sont nécessairement dépendants du schéma d'annotation utilisé dans le corpus exploité. À titre d'illustration, Agić & Ljubešić (2015) constatent que le corpus croate SETimes contient 10,1 % de phrases non projectives s'il est traité selon un schéma d'annotation favorisant les têtes fonctionnelles ; ce taux chute à 7,6 % si le même corpus est annoté suivant les principes du projet UD, qui favorise les têtes lexicales. Si ParCoTrain-Synt favorisait les têtes lexicales, le taux de phrases non projectives serait donc sans doute inférieur au taux actuel de 17 %.

Au-delà de ces principes d'annotation généraux qui impactent le corpus dans sa totalité, nous avons identifié deux traitements particuliers qui affectent la représentation de la non-projectivité : le traitement des auxiliaires et celui des relatifs. Nous illustrons leurs effets respectifs dans la suite.

Comme déjà expliqué dans la section 5.2.3, dans le corpus ParCoTrain-Synt, c'est le verbe lexical qui est considéré comme la racine de la phrase et le gouverneur du sujet, plutôt que le verbe auxiliaire. Dans la majorité des occurrences non projectives causées

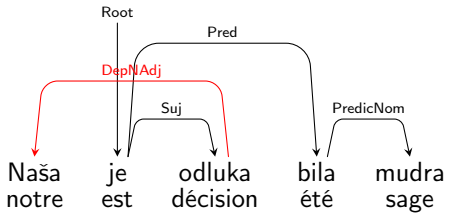
par les auxiliaires qui ont été détectées dans notre corpus, les deux traitements génèrent de la non-projectivité (cf. exemples 34a et 34b).

(34) a.



'Notre décision était sage.'

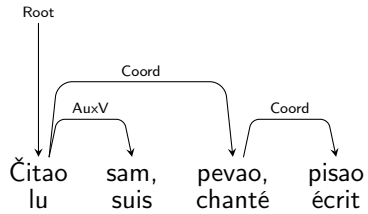
b.



idem

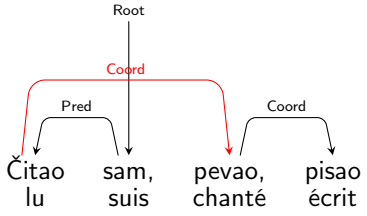
En revanche, dans le cas où l'auxiliaire introduit une suite de participes coordonnés, ces traitements donnent des résultats différents. Dans cette situation, en serbe, un auxiliaire à la forme enclitique se positionne après le premier participe. Le traitement adopté dans ParCoTrain-Syntne mène pas à la non-projectivité, vu que c'est le premier participe qui est annoté comme la racine de la phrase (cf. exemple 35a). En revanche, si l'on considère l'auxiliaire comme racine, la même phrase est non projective (cf. exemple 35b).

(35) a.



'J'ai lu, chanté, écrit.'

b.



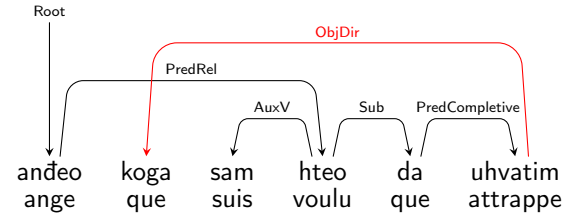
idem

Notons encore que le traitement de la coordination joue également un rôle ici. Si

l'on décidait que le premier conjoint gouverne directement tous les autres, l'exemple 35a resterait projectif, mais dans l'exemple 35b, la dépendance reliant le premier participe au dernier serait également non projective.

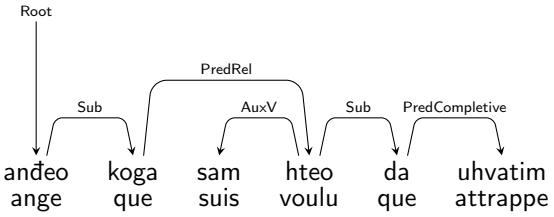
Quant aux relatives, notre schéma d'annotation traite le relatif en fonction de son rôle à l'intérieur de la relative, ce qui peut générer de la non-projectivité, ainsi dans l'exemple 36a. En revanche, une approche alternative discutée dans la section 5.2.8 annote les relatifs comme la tête de la proposition relative, ce qui induit une structure projective (cf. exemple 36b).

(36) a.



'ange que je voulais attraper'

b.



idem

Compte tenu des observations faites ci-dessus, si les résultats présentés dans la suite de ce chapitre permettent d'estimer la fréquence de différentes structures non projectives, ils ne sont pas directement comparables avec ceux des autres études dans la mesure où des schémas d'annotation différents ont été utilisés.

11.4 Analyse formelle de la non-projectivité dans le corpus

Nous avons évoqué le fait que trois propriétés formelles des arbres syntaxiques sont utilisées dans l'analyse de la non-projectivité dans différentes langues (*well-nestedness*, *maximum edge degree* et *maximum gap degree*). Ces propriétés, définies ci-dessous, permettent de décrire plus précisément les arbres non projectifs rencontrés en corpus et d'identifier, par conséquent, les caractéristiques dont doivent disposer les algorithmes de parsing afin de pouvoir prendre en compte les données provenant de différentes langues naturelles. En effet, l'examen de ces propriétés a montré que seul un sous-ensemble de tous les types d'arbres non projectifs sont réellement représentés dans les corpus existants (cf. Kuhlmann

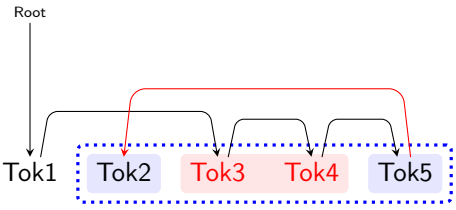
& Nivre, 2006 ; Havelka, 2007). Afin de pouvoir situer le serbe par rapport aux autres langues quant à cet ensemble de critères, nous examinons d'abord les propriétés formelles des structures non projectives présentes dans notre corpus.

11.4.1 Définition des propriétés formelles des structures non projectives

Dans l'analyse du corpus, nous suivons les définitions formelles présentées dans (Kuhlmann & Nivre, 2006) et (Havelka, 2007). Afin de faciliter la lecture de ce document, nous proposons ici une description moins formelle des principaux concepts exploités.

Une phrase est constituée d'une séquence de tokens. Un arbre en dépendances représentant la structure de la phrase a la forme d'un graphe acyclique connecté et orienté, dont la racine est représentée par un token fictif, externe à la phrase. Les tokens représentent les **nœuds** du graphe, alors que chaque arc de dépendance reliant un gouverneur à son dépendant représente une **arête**. On considère qu'un nœud en **domine** un autre si le premier nœud est l'ancêtre du deuxième : dans l'exemple 37, $Tok5$ domine $Tok2$, alors que $Tok3$ domine les tokens 4, 5 et 2. La **projection** d'un nœud correspond au sous-arbre syntaxique dominé par le nœud ; elle contient donc le nœud en question et tous ses descendants. La projection du nœud $Tok5$ dans le même exemple comprend donc les tokens 5 et 2, alors que celle de $Tok4$ contient les tokens 4, 5 et 2. Un nœud est considéré comme **projectif** si sa projection ne contient pas de *gap* (hiatus), un ***gap*** étant défini par le fait que deux nœuds adjacents d'un sous-arbre sont séparés par un ou plusieurs tokens d'un sous-arbre différent (Holan et al., 1998). Dans ce cas, on considère que les tokens intercalés se trouvent dans le *gap*. D'après ces définitions, le nœud $Tok5$ dans l'exemple 37 est non projectif : le sous-arbre qu'il domine (dessiné en pointillé) comporte un *gap* (indiqué par la bulle rouge) qui contient les tokens 3 et 4. Ceux-ci proviennent du sous-arbre dominé par $Tok1$, et non pas par $Tok5$. En revanche, le nœud $Tok3$ dans le même exemple est projectif : il n'y a pas de *gap* dans sa projection, qui contient les tokens 3, 4, 5 et 2.

(37)



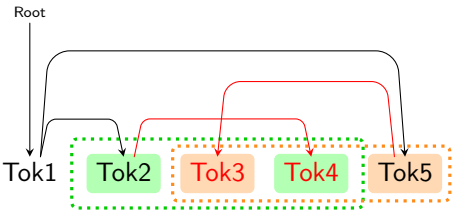
Arête non projective, nœud non projectif et *gap*

En plus de la notion de nœuds non projectifs, nous exploitons celle des arêtes (dépendances) non projectives, suivant (Havelka, 2007). Une **arête non projective** est une arrête allant du token i au token j et telle que au moins un token entre i et j n'est pas dominé par i. C'est le cas de la dépendance allant de $Tok5$ à $Tok2$ (dessinée en rouge)

dans l'exemple 37, où $Tok3$ et $Tok4$ ne sont pas dominés par $Tok5$. Cette même arête illustre un autre point important : une arête non projective peut impliquer plusieurs nœuds non projectifs. Ici, il existe une seule arête non projective, $Tok5 \rightarrow Tok2$. Elle correspond néanmoins à deux nœuds non projectifs : $Tok5$ (avec $Tok3$ et $Tok4$ dans le *gap*) et $Tok4$, dont la projection ($Tok4$, $Tok5$, $Tok2$) comporte un gap qui contient le $Tok3$, dominé par $Tok1$.

Un arbre est considéré comme projectif si tous ses nœuds et toutes ses dépendances sont projectifs. Les indicateurs de base en ce qui concerne le degré de non-projectivité admis par une langue consistent donc à mesurer le pourcentage de dépendances, de nœuds et d'arbres non projectifs. En plus de ces indicateurs directs, Kuhlmann & Nivre (2006) analysent plusieurs autres caractéristiques formelles de la non-projectivité, notamment le *maximum gap degree* (degré maximal du hiatus) (Holan et al., 1998 ; Bodirsky et al., 2005), le *maximum edge degree* (degré maximal de l'arête) (Nivre, 2006) et la propriété de *well-nestedness* (emboîtement) (Bodirsky et al., 2005). Le *gap degree* d'un nœud correspond au nombre de *gaps* distincts dans son sous-arbre, indépendamment de leur taille. Le *edge degree* d'un nœud représente le nombre d'arêtes différentes arrivant dans un *gap*, autrement dit, le nombre de sous-arbres disjoints dans le *gap*. Le *maximum gap degree* et le *maximum edge degree* représentent la valeur maximale de ces paramètres trouvée dans l'arbre. Pour l'arbre montré dans l'exemple 37, le nœud $Tok5$ contient un seul *gap* (celui contenant les tokens 3 et 4), et ce *gap* contient un seul sous-arbre (celui dominé par $Tok1$). Son *gap degree* et son *edge degree* ont donc tous les deux la valeur de 1. Il en est de même pour le nœud non projectif $Tok4$. Par conséquent, le *maximum gap degree* et le *maximum edge degree* de l'arbre sont également de 1.

(38)



Arbre qui ne respecte pas la contrainte de *well-nestedness* (mal emboîté, *ill-nested*)

La dernière propriété que nous examinons dans cette étude, à savoir celle de *well-nestedness*, relève du positionnement des sous-arbres non projectifs dans un arbre. Elle postule que pour tous nœuds A et B, si le nœud A ne domine pas le nœud B, alors A ne domine aucun *gap* dans le sous-arbre de B. Si l'on considère l'arbre de l'exemple 38, on constate qu'il existe deux arêtes non projectives : $Tok5 \rightarrow Tok3$, avec le nœud $Tok4$ dans le *gap*, et $Tok2 \rightarrow Tok4$, avec le nœud $Tok3$ dans le *gap*. Dans les deux cas, le nœud dans le *gap* est gouverné par un nœud d'un sous-arbre disjoint. En effet, les projections respectives

des $Tok2$ (en vert) et $Tok5$ (en orange) s'entrelacent. Par conséquent, l'arbre de l'exemple 38 est un arbre non projectif qui ne respecte pas la contrainte de *well-nestedness* : il est mal emboîté ou *ill-nested*. Ce n'est pas le cas de l'arbre dans l'exemple 37 : le *gap* entre $Tok5$ et $Tok2$ est dominé par $Tok1$, mais $Tok1$ domine également $Tok5$. Par conséquent, cet arbre est bien emboîté.

Les résultats de l'analyse de notre corpus relatifs à ces indicateurs sont donnés dans la section suivante.

11.4.2 *Maximum edge degree*, *maximum gap degree* et *well-nestedness* en serbe

Nous avons extrait de notre corpus de travail les informations relatives à la fréquence des dépendances et des arbres non projectifs et à leurs propriétés. Ceci a été effectué avec le module d'analyse statistique de corpus du parser Talismane. Les résultats concernant la fréquence des dépendances non projectives, des arbres non projectifs et des arbres qui ne respectent pas la contrainte de *well-nestedness* sont donnés dans le tableau 11.2, alors que le tableau 11.3 présente les différentes valeurs de *maximum gap degree* et *maximum edge degree* trouvées en corpus.

Pour comparaison, nous indiquons les mêmes résultats pour d'autres langues à partir des travaux existants : les données sur le tchèque, le slovène et le néerlandais dans le tableau 11.2 proviennent de (Havelka, 2007), alors que celles sur le tchèque et le danois dans le tableau 11.3 ont été reprises de (Kuhlmann & Nivre, 2006). Les données sur le grec ancien et le hindi dans les deux tableaux viennent respectivement de (Mambrini & Passarotti, 2013) et de (Bhat & Sharma, 2012b). Nous choisissons ces langues pour les raisons suivantes : le tchèque et le slovène appartiennent à la même famille linguistique que le serbe et il est donc raisonnable de supposer des résultats comparables sur les trois langues. Le danois et le néerlandais sont parmi les langues européennes connues pour leurs structures non projectives, et les données sur le hindi permettent de faire des comparaisons avec une langue relativement éloignée. Enfin, le grec ancien est la langue pour laquelle les travaux existants montrent les taux de non-projectivité les plus élevés [2].

Les résultats présentés dans le tableau 11.2 montrent que le serbe exhibe le pourcentage le moins élevé de dépendances non projectives parmi les langues analysées ($<$1 %), et que le pourcentage d'arbres non projectifs est également parmi les plus bas observés ici. Il n'est néanmoins pas négligeable : 17 % de phrases dans notre corpus contiennent au moins une dépendance non projective.

Ainsi, le serbe se démarque du profil respectif du tchèque et du slovène : dans ces deux langues, on observe 2,13 % de dépendances non projectives reparties sur 22 % à 23 % des

2. Il faut toutefois noter que le corpus sur lequel le travail de Mambrini & Passarotti (2013) a été effectué contient majoritairement de la poésie.

Langue	Arêtes		Arbres		
	Tot. arêtes	Non proj.(%)	Tot. arbres	Non proj.(%)	Mal emboîtés (%)
Serbe	81 204	0,75	2949	17,06	0,17
Tchèque	1 105 437	2,13	72 703	23,15	0,11
Slovène	25 777	2,13	1 534	22,16	0,20
Néerlandais	179 063	5,90	13 349	36,44	0,11
Hindi	NA	1,65	20 497	14,85	0,19

TABLE 11.2 – Arêtes non projectives, arbres non projectifs et mal emboîtés en serbe et dans d'autres langues

Langue	Arbres	Gap degree (%)				Edge degree (%)					
		Gd0	Gd1	Gd2	Gd3	Ed0	Ed1	Ed2	Ed3	Ed4	Ed5
Serbe	2949	82,94	16,58	0,44	0,03	82,94	15,36	1,66	0,03	-	-
Tchèque	73 088	76,85	22,72	0,42	0,01	76,85	22,69	0,35	0,09	0,01	<0,01
Danois	4393	84,95	14,89	0,16	-	84,95	13,29	1,32	0,39	0,05	-
Hindi	20 497	85,14	14,56	0,28	0,02	85,14	14,24	0,45	0,11	0,03	-
Grec ancien	24 825	25,20	68,33	6,17	0,28	25,20	43,73	14,15	7,07	3,88	-

TABLE 11.3 – *Gap degree* et *edge degree* en serbe et dans d'autres langues

arbres. Leurs résultats sont donc très cohérents, malgré la différence de taille importante entre les deux corpus. Il faut néanmoins noter que l'analyse de ces deux langues par Havelka (2007) a été effectuée sur le treebank tchèque PDT (Hajič, 1998) et le treebank slovene SDT (Džeroski et al., 2006). Le schéma d'annotation utilisé dans l'élaboration du treebank slovène a été largement repris du corpus tchèque PDT. Ces deux ressources partagent donc leurs principes d'annotation, alors que notre corpus en diffère (cf. section 5.2). Il est alors possible que certains phénomènes non projectifs soient traités de manières différentes dans ces deux corpus par rapport au nôtre, ce qui pourrait expliquer les différences observées.

Une autre observation importante à faire concerne la contrainte de *well-nestedness* : dans toutes les langues analysées, y compris le grec ancien, le taux d'arbres qui la respectent dépasse 99 %. En effet, le taux d'arbres mal emboités est inférieur ou égal à 0,2 %. C'est également le cas de la majorité des autres langues étudiées par Havelka (2007) : dans les corpus du bulgare, du japonais et de l'espagnol, on ne trouve pas d'arbres mal emboîtés, et pour le néerlandais, le portugais et l'arabe, le pourcentage observé reste en dessous de 0,2 %. Les seules exceptions sont l'allemand (1,06 %), le suédois (0,64 %) et le turc (0,28 %). Ces résultats, ainsi que les résultats de Kuhlmann & Nivre (2006), indiquent que le principe de *well-nestedness* est un relâchement utile de la contrainte de projectivité : en général, il suffit qu'un parser soit capable de traiter les structures non projectives qui respectent cette contrainte pour atteindre une couverture de 99 % de données des corpus analysés ; il n'est donc pas indispensable de viser la maîtrise de tous les types de structures

projectives. D'après nos résultats, cette observation globale s'applique également au serbe.

Quant aux paramètres de *maximum gap degree* et de *maximum edge degree* donnés dans le tableau 11.3, nous constatons que les langues modernes présentent un profil cohérent entre elles : un *gap degree* de 0 ou 1 et les mêmes valeurs de *edge degree* permettent de couvrir plus de 98 % de données. Le grec ancien montre un comportement différent : c'est la seule langue pour laquelle plus de 6 % des phrases correspondent à un *gap degree* supérieur à 2, et où plus de 14 % des phrases exhibent un *edge degree* supérieur à 2. Cependant, il a déjà été mentionné que le corpus de grec ancien sur lequel ces résultats ont été obtenus contient majoritairement de la poésie ; ces déviations par rapport aux langues modernes peuvent donc relever des spécificités du genre textuel.

Si l'on considère de plus près les langues modernes, le serbe exhibe un comportement très proche de celui des autres langues quant au *gap degree* : pour 99,52 % des phrases, la valeur de ce paramètre est inférieure ou égale à 1. Un *gap degree* de 1 permet donc déjà une excellente couverture de nos données. La situation est légèrement différente en ce qui concerne le *edge degree* : à côté du danois, le serbe est la seule langue moderne pour laquelle un *edge degree* inférieur ou égal à 1 couvre moins de 99 % de phrases, le taux exact étant de 98,3 %. Dans le cas du serbe, il serait donc nécessaire d'autoriser des phrases non projectives avec un *edge degree* allant jusqu'à 2 afin d'atteindre une couverture des données supérieure à 99 %.

Bien que cette analyse quantitative nous ait permis d'identifier des relâchements utiles à la contrainte de projectivité dans le traitement du serbe, elle ne nous informe pas sur les types de constructions linguistiques qui mènent à ces différentes formes de non-projectivité. Cet aspect de notre étude est présenté dans la section suivante.

11.5 Structures linguistiques non projectives en serbe

Comme mentionné ci-dessus, une analyse linguistique des structures non projectives en corpus a déjà été effectuée pour plusieurs langues (cf. section 11.1).

Par exemple, Hajičová et al. (2004) analysent le tchèque en se servant du corpus PDT. Ils identifient 12 constructions non projectives différentes au niveau de la syntaxe de surface et les classent en fonction des structures correspondantes au niveau de la syntaxe profonde. Mannem et al. (2009) ont travaillé sur le hindi en utilisant un treebank initial de 35 000 tokens. Ils décrivent 9 structures non projectives tout en accordant une attention particulière à la distinction entre la non-projectivité obligatoire et la non-projectivité optionnelle en identifiant les constructions qui admettent un réordonnancement de mots qui donne une structure projective. Bhat & Sharma (2012b) ont utilisé une version enrichie du même treebank et ont étendu leur analyse à trois autres langues indiennes (l'ourdou,

Type de non-proj.	Occur.	%	Exemples
Splitting	254	41,85	(39,40,41)
Permutation trans-propositionnelle	89	14,66	(45)
Extrapostion	81	13,34	(46)
Mouvement *wh-*	63	10,38	(42,43,44)
Topicalisation	19	3,13	-
Scission du pronom négatif	11	1,81	(47)
Autre	66	10,87	-
Non analysé	24	1,2	-
Total	607		

TABLE 11.4 – Distribution de la non-projectivité par type de structure

le bengali et le télougou). Ils identifient 8 constructions spécifiques en fonction du type de non-projectivité observé : la topicalisation, l'extraposition, l'extraction du GN, le quantificateur flottant, le *scrambling* (permutation de l'ordre des dépendants verbaux) et la non-projectivité inhérente. Quant au travail sur le grec ancien, Mambrini & Passarotti (2013) classent les dépendances non projectives selon le type de leur gouverneur (nom ou verbe) et analysent plus en détail le rôle des clitiques.

Afin d'établir des comparaisons avec les résultats de ces travaux, nous procédons à une analyse des constructions non projectives dans notre corpus, présentée dans la suite.

11.5.1 Nature et fréquence des constructions non projectives en serbe

Au total, 607 dépendances non projectives ont été détectées dans notre corpus de travail. La grande majorité d'entre elles appartiennent à des types de discontinuité déjà connus, comme le mouvement *wh-* (*wh-fronting*), l'extraposition, la topicalisation, le *scrambling* à longue distance (permutation trans-propositionnelle des dépendants du verbe) et le *splitting* (constructions scindées) [3]. Nous avons également observé une structure qui semble spécifique au serbe : il s'agit du détachement du préfixe des pronoms dits négatifs à l'intérieur d'un GP. La distribution des occurrences en fonction du type de non-projectivité est donnée dans le tableau 11.4.

La dernière ligne du tableau (occurrences non analysées) comprend les dépendances non projectives dues aux irrégularités intrinsèques aux textes (p. ex., des subordonnées sans verbe principal) ou aux erreurs d'annotation manuelle. Toutes les autres occurrences ont été classées manuellement selon le type de construction représenté. La catégorie *Autre* correspond aux cas de figure non systématiques, avec trop peu d'occurrences pour aboutir à une analyse informative. Il s'agit notamment des éléments extraprédicatifs et du discours

3. Pour une définition de ces constructions dans le cadre de la syntaxe des dépendances, nous renvoyons à (Groß & Osborne, 2009).

rapporté. Nous constatons que plus de 40 % de la non-projectivité en corpus est causée par les constructions scindées, suivies par le *long-distance scrambling*, l'extraposition et le *wh-fronting*, couvrant respectivement 14 %, 13 % et 10 % d'occurrences. La topicalisation et la scission du pronom négatif sont beaucoup moins fréquentes (respectivement 3 % et 1 % d'occurrences). Par conséquent, dans la suite de cette section nous discuterons plus en détail les 4 types de non-projectivité les plus fréquents : la scission dans la section 11.5.2, le mouvement *wh-* dans la section 11.5.3, la permutation transpropositionnelle des dépendants du verbe dans la section 11.5.4, et l'extraposition dans la section 11.5.5. Enfin, nous présenterons brièvement la scission du pronom négatif (section 11.5.6), étant donné sa nature particulière à l'interface de la morphologie et de la syntaxe.

Avant de poursuivre, il nous semble judicieux d'apporter quelques remarques relatives au fonctionnement syntaxique du serbe. Tout d'abord, rappelons que cette langue dispose d'un ordre des constituants flexible, permettant les 6 permutations possibles des constituants de base (SVO, SOV, VSO, VOS, OVS, OSV) (cf. section 1.1.2).

Quant aux enclitiques, Corbett (1987) identifie un *cluster* d'enclitiques constitué de 6 créneaux, dédiés à différentes formes enclitiques des auxiliaires et des pronoms, ainsi qu'à la particule interrogative *li*. Une analyse détaillée de la structure morphologique et syntaxique du *cluster* est proposée dans (Groß, 2011). Dans le cadre de cette étude, sa propriété la plus pertinente est que la contrainte de Wackernagel fait que le *cluster* peut se positionner immédiatement après le premier élément du groupe en tête de phrase, séparant ainsi cet élément du reste du groupe. Les enclitiques représentent par conséquent un facteur important dans les constructions non projectives en serbe.

Enfin, le serbe est doté d'une propriété qui n'est pas typique de la majorité des langues slaves, mais qui est partagée par d'autres langues des Balkans. Il s'agit du fait que la construction à contrôle (avec deux verbes qui partagent le même sujet) connaît deux expressions différentes. Elle peut se réaliser sous forme d'une construction infinitive, mais aussi sous forme d'une complétive pleine, introduite par le subordonnant *da* 'que' avec un verbe au présent. Par conséquent, les phrases comme *Filip želi kupiti knjigu* lit. 'Filip veut acheter livre' et *Filip želi da kupi knjigu* lit. 'Filip veut que [il] achète livre' signifient la même chose : 'Filip veut acheter un/le livre'. Les deux constructions sont impliquées dans plusieurs types de structures non projectives dans notre corpus, et notamment dans le *long-distance scrambling*.

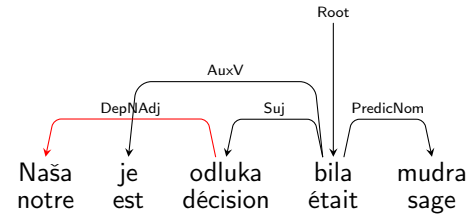
11.5.2 *Splitting* (constructions scindées)

Les constructions scindées ou le *splitting* relèvent des cas de figure dans lesquels un élément s'insère dans le groupe en tête de la proposition, séparant ainsi un dépendant de son gouverneur (cf. exemples 39a, 39b et 39c). Ce type de non-projectivité est le plus

fréquent dans notre corpus : il représente environ 33 % de toutes les dépendances non projectives. Le *splitting* du GN est également une source de non-projectivité importante en tchèque : Hajičová et al. (2004) indiquent que cette construction représente 11 % des dépendances non projectives dans PDT.

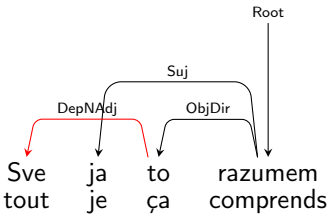
Dans notre corpus, les constructions scindées comportent typiquement un enclitique ou un *cluster* d'enclitiques positionné à la 2e place dans la phrase, directement après le premier élément du groupe en tête de la phrase, détachant ainsi cet élément du reste du groupe. Comme les enclitiques sont typiquement gouvernés par le verbe principal, cela mène souvent à des dépendances non projectives dans l'arbre (cf. exemple 39a).

(39) a.



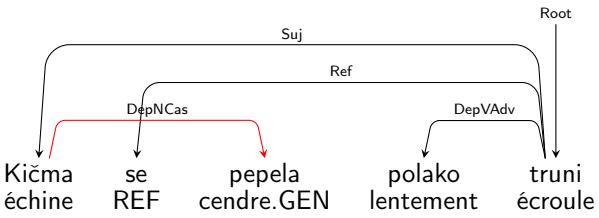
'Notre décision était sage.'

b.



'Je comprends tout ça.'

c.



'L'échine des cendres s'écroule lentement.'

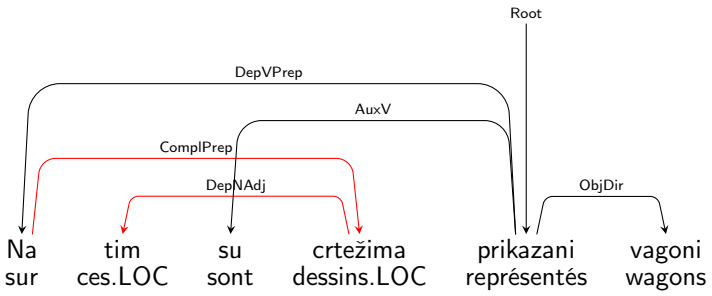
Néanmoins, le *splitting* peut également être créé par une forme non clitique, comme dans l'exemple 39b : ici, la forme *ja* 'je', qui se trouve dans le *gap*, représente la forme pleine du pronom, et non pas sa forme clitique.

Dans les exemples 39a et 39b, le *splitting* se réalise entre le gouverneur et sa branche gauche. Cependant, elle peut également apparaître entre le gouverneur et sa branche droite, comme dans l'exemple 39c, où le nom au génitif *pepela* (de *pepeo* 'cendre') est le dépendant droit du nom sujet *kičma* 'échine'.

Enfin, les gouverneurs nominaux ne sont pas les seuls concernés par ce type de non-

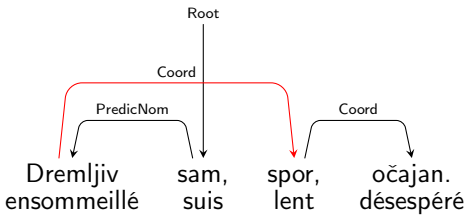
projectivité : quoique beaucoup moins fréquent, le *splitting* peut se réaliser aussi à l'intérieur d'un GA ou d'un GAdv, selon les mêmes principes. Ce cas de figure représente 16,4 % de toutes les occurrences de la scission identifiées en corpus.

Un type de scission particulier concerne les GN qui se trouvent à l'intérieur d'un GP en tête de proposition. Comme les prépositions sont des proclitiques en serbe, elles forment une unité prosodique avec le contenu immédiatement à leur droite. Par conséquent, un enclitique (ou un *cluster* d'enclitiques) ne peut pas s'insérer directement après la préposition ; il occupe plutôt la position après le premier élément du GN qui dépend de la préposition. Dans ce cas de figure, deux dépendances sont non projectives, puisque le sous-arbre dominé par la préposition ainsi que celui dominé par le complément de la préposition contiennent le même *gap* (cf. l'exemple 40).



(40) 'Sur ces dessins sont représentés des wagons.'

Dans tous les exemples cités ci-dessus, la non-projectivité est optionnelle : l'enclitique ou le *cluster* peuvent aussi bien occuper une position à côté du verbe sans provoquer un changement de sens important. Ainsi, la phrase dans l'exemple 40 peut être reformulée comme *Na tim crtežima* ***su*** *prikazani vagoni* ou comme *Na tim crtežima prikazani* ***su*** *vagoni*. Au contraire, la non-projectivité paraît obligatoire si une forme clitique du verbe *biti* 'être' introduit un prédicatif nominal coordonné (cf. exemple 41).



(41) 'Je suis ensommeillé, lent, désespéré.'

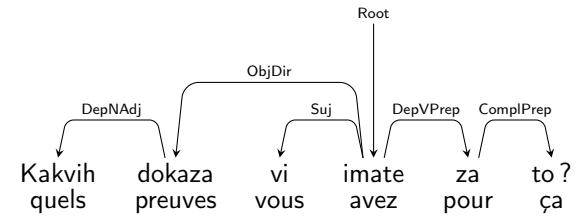
Si l'on cherche à établir une version projective de cette phrase par un réordonnancement de ses éléments, la seule solution est d'avoir le verbe soit en position initiale soit en position finale dans la phrase. La première de ces deux options est impossible parce que le verbe

sam ‘suis’ est un enclitique et doit être précédé par une forme prosodiquement pleine. La deuxième est jugée comme agrammaticale par nos informateurs. Cette incapacité de l’enclitique à occuper la position finale dans la phrase semble due à la nature de l’élément qui le précède, étant donné que ce positionnement est possible avec un prédicatif nominal simple : *Dremljiv sam.* ‘Je suis ensommeillé’. Ceci est également le cas dans la configuration analogue où la forme enclitique du verbe *biti* ‘être’ est un auxiliaire qui introduit une suite de participes coordonnés : *Čitao sam, pisao, pevao* ‘J’ai lu, écrit, chanté’. Bien que cet exemple ne soit pas non projectif selon notre schéma d’annotation (cf. section 11.3), il présente le même comportement quant au positionnement de l’enclitique : les phrases ***Sam*** *čitao, pisao, pevao* et *Čitao, pisao, pevao* ***sam*** sont toutes les deux agrammaticales, alors que *Čitao sam* est grammatical.

Comme mentionné dans la section 11.1, Mambrini & Passarotti (2013) soulignent le fait que les 5 mots qui apparaissent le plus souvent dans le *gap* en grec ancien sont des postpositifs (majoritairement des clitiques), et qu’ils sont responsables de presque 40 % des *gaps*. Des observations liées aux clitiques ont également été faites sur le tchèque : Hajičová et al. (2004) indiquent que les cas de figure où la particule interrogative *li* occupe la deuxième position dans la phrase et cause la non-projectivité correspondent à 5,1 % des dépendances non projectives dans un échantillon de 615 phrases tirées du PDT. Nos constats présentés ci-dessus confirment pour le serbe que le comportement des clitiques sujets à la loi de Wackernagel est une source importante de non-projectivité.

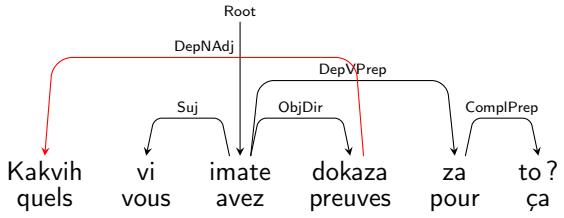
11.5.3 Mouvement *wh-*

(42) a.



‘Quelle preuve en avez-vous ?’

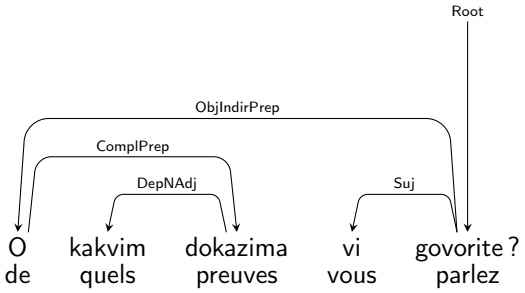
b.



(*idem*)

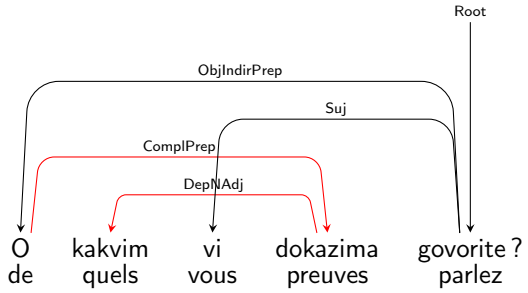
Tout comme dans de nombreuses autres langues, les mots interrogatifs et relatifs (les

(43) a.



'De quelles preuves parlez-vous ?'

b.



(*idem*)

mots en *qu-*, ou, en anglais, les mots *wh-*) en serbe ont tendance à occuper la position en tête de la proposition, que ce soit dans les questions directes ou indirectes ou dans les subordonnées relatives. Notons que la *Left Branch Condition* (Ross, 1967) ne tient pas en serbe : à la différence de l'anglais, en serbe un adjectif interrogatif peut être détaché de son gouverneur et positionné seul en tête de la proposition. Cela signifie que les exemples 42a et 42b sont tous les deux possibles, la seule différence étant que dans le premier tout le GN est topicalisé, alors que dans le deuxième c'est seulement l'interrogatif. Le deuxième génère une structure non projective.

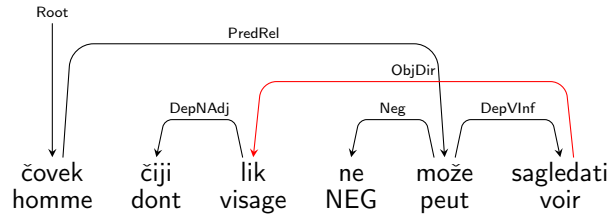
C'est une deuxième propriété que le serbe partage avec le tchèque : d'après Hajičová et al. (2004), les interrogatifs et les relatifs en tchèque peuvent également se retrouver en position initiale sans leur gouverneur, et cette construction correspond à 1,6 % des dépendances non projectives dans leur corpus.

À la différence de l'anglais, en serbe le dégagement de la préposition (*preposition stranding*) n'est pas possible. Par conséquent, si le mot en *qu-* se trouve à l'intérieur d'un GP, tout le GP se positionne en tête de la proposition (cf. exemple 43a). En revanche, le GN qui dépend de la préposition peut subir une scission, comme dans l'exemple 43b. Cela produit deux dépendances non projectives suivant les mêmes principes que dans l'exemple 40.

Avec la construction à contrôle, qu'elle soit réalisée sous la forme d'une proposition infinitive ou d'une complétive *da*+V_{pres}, le mot en *qu-* se positionne naturellement devant

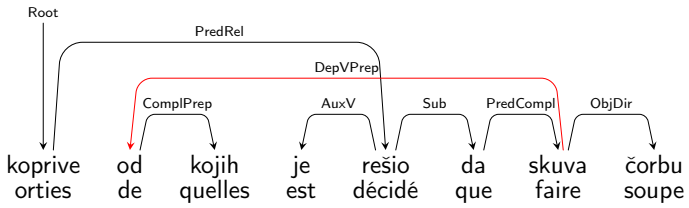
le verbe qui introduit la construction (cf. exemples 44a et 44b). Cela cause de la non-projectivité même dans les structures qui ne seraient pas discontinues dans une proposition simple (p. ex., avec les pronoms relatifs qui dépendent directement du verbe).

(44) a.



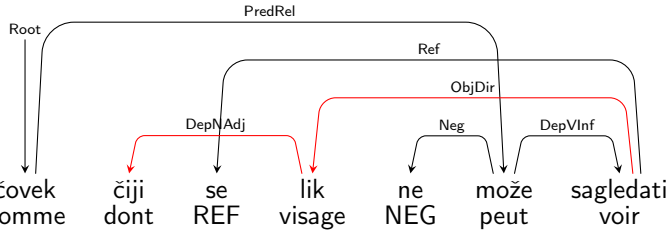
'l'homme dont il/elle ne peut voir le visage'

b.



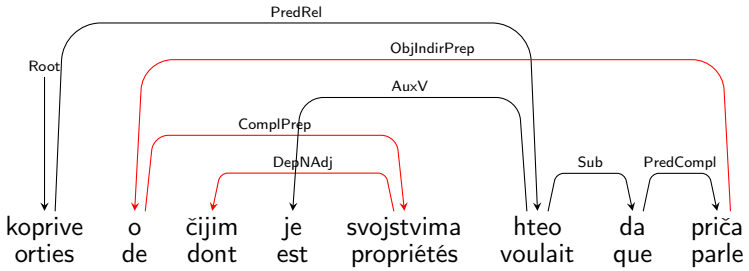
'les orties dont il a décidé de faire une soupe'

c.



'l'homme dont le visage ne peut être vu'

d.



'les orties dont il voulait parler des propriétés'

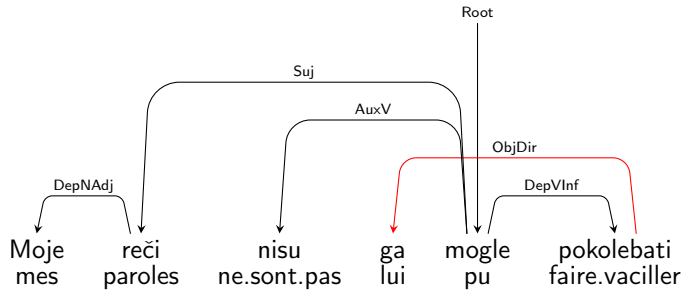
Par ailleurs, ces contextes ne bloquent pas la réalisation du *splitting*. Comme on peut voir dans les exemples 44c et 44d, une telle configuration peut mener à l'apparition de deux ou même trois dépendances non projectives dans la même phrase. Cette configuration n'est pas rare dans nos données : elle représente 31 % de toutes les occurrences de non-projectivité liée au comportement des mots en *qu-*. Les propositions relatives étant difficiles pour le parsing en elles-mêmes, on peut supposer que les constructions que nous venons

d'illustrer ne font que complexifier leur traitement.

11.5.4 Permutation trans-propositionnelle des dépendants du verbe

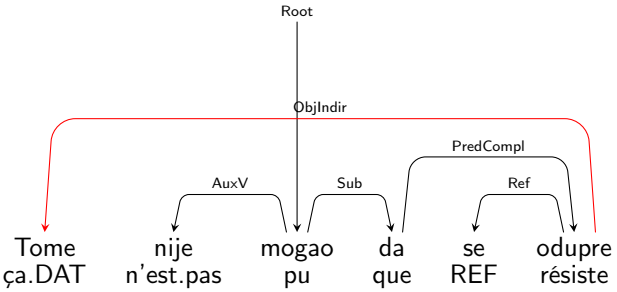
Un dépendant d'une construction à contrôle peut également apparaître en dehors de sa proposition même s'il ne s'agit pas d'une forme interrogative ou relative. Cela est possible aussi bien avec une proposition infinitive (cf. exemple 45a) qu'avec une complétive *da*+V_{pres} (cf. exemple 45b). Autrement dit, le serbe autorise la permutation trans-propositionnelle des dépendants du verbe (*long-distance scrambling*).

(45) a.



'Mes paroles n'ont pas pu le faire vaciller.'

b.



'À ça il ne pouvait pas résister.'

C'est également le cas du tchèque : selon les observations de Hajičová et al. (2004), la permutation trans-propositionnelle des dépendants d'un infinitif représente 9 % des dépendances non projectives dans leur corpus. Cette propriété est également présente en hindi (Bhat & Sharma, 2012b) ; en revanche, sa fréquence n'est pas établie. Rappelons que, dans notre corpus, cette structure représente 14 % de toutes les structures non projectives.

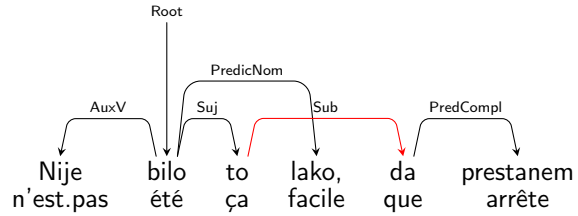
Tandis que la non-projectivité était incontournable dans le cas des interrogatifs et des relatifs, il ne l'est pas ici, au moins avec les complétives : l'objet indirect *tome* 'à ça' dans l'exemple 45b peut aisément occuper sa place canonique à l'intérieur de la complétive : *Nije mogao da se odupre* ***tome***, ou bien *Nije mogao da se* ***tome*** *odupre*. La permutation présente dans l'exemple original contribue à produire un effet de topicalisation sur l'élément qui apparaît en dehors de sa position canonique. Néanmoins, la situation est moins nette avec les infinitives : si dans l'exemple 45a on essaye de repositionner l'objet direct *ga* 'le' à

l'intérieur de la proposition, les deux réalisations possibles sont jugées douteuses de la part de nos informateurs : ??*Moje reči nisu mogle* ***ga*** *pokolebati*, ??*Moje reči nisu mogle pokolebati* ***ga***. Cela est possiblement dû au fait que la forme *ga* est un enclitique : si la forme pleine *njega* est utilisée, les deux phrases deviennent grammaticales, mais le pronom reçoit une interprétation topicalisée, cf. les phrases *Moje reči nisu mogle* ***njega*** *pokolebati* and *Moje reči nisu mogle pokolebati* ***njega***, qui se traduisent toutes les deux comme 'Lui, mes paroles n'ont pas pu le faire vaciller.'

11.5.5 Extraposition

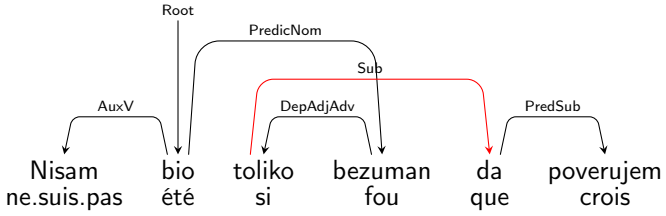
L'extraposition typique se réalise sous la forme d'un élément positionné plus loin à droite par rapport à sa position canonique à cause de son poids (cf. exemple 46a). Cependant, dans notre corpus elle se présente sous plusieurs cas de figure ; nous en examinons deux dans la suite.

(46) a.



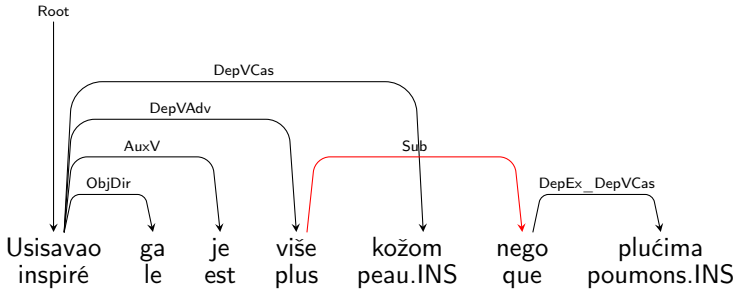
'Ce n'était pas facile, que je m'arrête.'

b.



'Je n'étais pas si fou que d'y croire.'

c.



'Il l'inspirait plus avec sa peau qu'avec ses poumons.'

La première d'entre elles, illustrée dans l'exemple 46b, correspond à une structure corrélative avec un intensifieur dans la proposition principale qui introduit une consécutive.

Dans l'exemple cité, l'adverbe occupe la position canonique d'un dépendant adverbial d'un adjectif – à gauche de son gouverneur. Or, la consécutive qu'il introduit est placée plus loin à droite, ce qui mène à la non-projectivité du nœud de l'adverbe. Une version projective de cette construction est possible si l'adverbe se positionne à droite de son gouverneur : *Nisam bio bezuman* ***toliko*** *da poverujem.* Notons que dans cette phrase, l'adverbe est topicalisé : 'Je n'étais pas fou au point d'y croire'.

La deuxième construction concerne une autre structure corrélative, dans laquelle figurent les formes de comparatif d'un adjectif ou d'un adverbe et leur dépendant introduit par *nego* 'que' (cf. exemple 46c). Ici aussi, un réordonnancement projectif de la phrase est possible si l'adverbe *više* est placé à droite du premier nom à l'instrumental : *Udisao ga je kožom* ***više*** *nego plućima.* Toutefois, cela produit un effet de topicalisation du premier nom : 'C'était plus avec la peau qu'avec les poumons qu'il l'inspirait'. Cette construction a également été observée dans PDT ; elle était la source de 2,7 % de la non-projectivité identifiée dans ce corpus (Hajičová et al., 2004).

11.5.6 Pronoms négatifs dans un GP

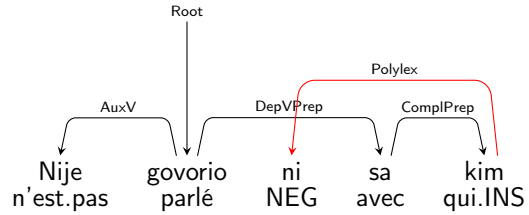
Ce type de non-projectivité n'est pas très fréquent dans notre corpus (1,8 %), mais nous le présentons comme une forme de discontinuité à la frontière entre la morphosyntaxe et la syntaxe. Ce phénomène est d'autant plus intéressant que nous n'avons pas rencontré de descriptions d'une structure similaire dans d'autres langues.

La scission d'un pronom négatif se réalise quand un pronom dit négatif apparaît dans un GP. Les pronoms dits négatifs comme *niko* 'personne' et *ništa* 'rien' sont dérivés respectivement des pronoms interrogatifs *ko* 'qui' et *šta* 'quoi', à l'aide du préfixe négatif *ni-*. Si un tel pronom apparaît dans un GP, le préfixe se détache du pronom et se positionne devant la préposition, ne laissant que la partie fléchie du pronom à la droite de la préposition (cf. exemple 47). Pour le moment, selon notre schéma d'annotation, ce préfixe est annoté comme un élément d'une unité polylexicale et rattaché à la partie fléchie du pronom, qui est à son tour gouvernée par la préposition. Par conséquent, cette structure mène à des dépendances non projectives.

Cette scission est parfois ignorée à l'oral : *Ne verujem u ništa* lit. 'Je crois.NEG en rien'. Cependant, elle est considérée comme la forme correcte du point de vue normatif, et elle semble systématique dans notre corpus.

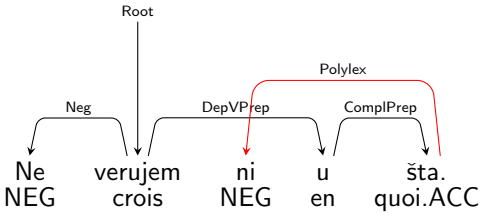
Il est tout de même important de remarquer que d'autres principes d'annotation syntaxique (ou morphosyntaxique) peuvent éliminer la représentation non projective de cette structure. À titre d'illustration, un schéma d'annotation syntaxique qui favorise les têtes lexicales, à l'instar de celui du projet UD, annoterait à la fois la préposition et le préfixe comme dépendants de la partie fléchie du pronom, ne donnant que des dépendances

(47) a.



'Il n'a parlé avec personne.'

b.



'Je ne crois en rien.'

projectives (cf. section 2.3.5).

L'analyse linguistique présentée ci-dessus montre qu'il existe une diversité importante de structures non projectives en serbe, et qu'elles sont bien attestées. Afin d'évaluer d'une manière plus précise la difficulté qu'elles posent au parsing, la section suivante est dédiée à une évaluation des performances de Talismane et du parser MST sur ces structures.

11.6 Parsing par transitions pseudo-projectif *vs* parsing par graphes : maîtrise des structures non projectives en serbe

Comme il a déjà été évoqué, l'une des différences principales entre les parsers par graphes et les parsers par transitions relève du traitement des dépendances non projectives. En effet, les parsers par transitions ne sont pas capables d'apprendre ou de produire des dépendances non projectives, contrairement aux parsers par graphes. Ces derniers ont donc une meilleure couverture des structures trouvées dans les langues naturelles et c'est grâce à cette capacité qu'ils sont souvent considérés comme mieux adaptés au traitement des langues à morphologie flexionnelle riche et à ordre des constituants flexible. Cette capacité a cependant un prix : le traitement non projectif augmente la complexité temporelle de la tâche du parsing, ce qui rend les parsers par graphes en général plus lents que les parsers par transitions. Par ailleurs, différentes stratégies visant le traitement des structures non projectives ont été mises en place pour les parsers par transitions, tel le parsing pseudo-projectif (Nivre & Nilsson, 2005), les transitions sur arcs non-adjacents (Attardi, 2006) et le réordonnancement en ligne (Nivre, 2009a) sans compromettre la vitesse du parsing (cf.

section 3.4.2 pour plus de détails). Les deux types d'outils disposent donc dans les faits de la capacité à traiter ces phénomènes linguistiques ; on peut néanmoins se demander si les extensions des parsers par transitions citées ci-dessus sont aussi efficaces que le parsing non projectif naturel des outils basés sur les graphes.

Afin d'évaluer la robustesse de ces deux types d'approches face aux données serbes, nous testons un représentant de chacune d'entre elles : la méthode du parsing par transitions pseudo-projectif de Talismane (Urieli, 2013) et le parsing par graphes du parser MST (McDonald et al., 2006). Talismane a été retenu pour ses résultats élevés dans nos évaluations précédentes (cf. chapitre 9), alors que le parser MST a été choisi principalement en tant que parser par graphes de référence, mais aussi parce qu'il a déjà été évalué sur le croate et le serbe (Agić et al., 2013b).

Au-delà d'une comparaison des résultats globaux de ces deux méthodes, nous nous intéressons ici à une analyse plus détaillée de leurs performances sur les structures non projectives. Plus précisément, nous cherchons à identifier leurs points forts et leurs points faibles vis-à-vis de nos données. Comme nous nous concentrons ici sur la maîtrise des structures non projectives de ces deux outils, nous isolons leurs performances en parsing : les tests sont faits exclusivement sur l'étiquetage morphosyntaxique *gold*. Nos observations principales sont présentées dans la suite.

11.6.1 Parsing pseudo-projectif de Talismane

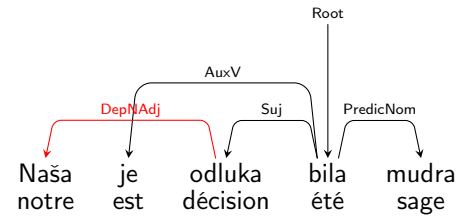
Dans les expériences décrites ici, nous utilisons le modèle de parsing identifié comme optimal dans le chapitre 9 (cf. section 9.5), suivi d'un post-traitement par le module du parsing pseudo-projectif de Talismane. Ce module a été développé par A. Urieli dans le cadre de notre travail commun sur la non-projectivité présenté dans (Miletic & Urieli, 2017).

Talismane met en place un système de traitement des dépendances non projectives proche de celui du parsing pseudo-projectif de (Nivre & Nilsson, 2005), lui-même basé sur la notion de pseudo-projectivité intorduite par (Kahane et al., 1998) : en amont de l'entraînement, les données sont rendues projectives tout en marquant les étiquettes des dépendances modifiées par un suffixe. L'entraînement est ensuite effectué sur ces données « projectivisées ». Si les étiquettes modifiées sont suffisamment fréquentes dans le corpus d'entraînement pour être apprises par le parser, elles sont reproduites lors du parsing. Une méthode heuristique est ensuite utilisée pour retrouver le véritable gouverneur de ces dépendances et corriger leur étiquette.

Dans le cas de Talismane, l'algorithme de projectivisation procède de manière suivante : pour chaque paire d'arcs croisés, on cherche à résoudre la non-projectivité en modifiant le rattachement du dépendant de l'un de ces arcs. Afin d'identifier un parent projectif

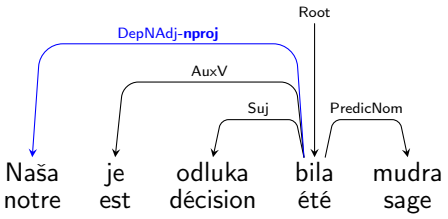
potentiel, on monte dans l'arbre en suivant les ancêtres de ces dépendants jusqu'à ce que l'on retrouve ceux qui permettent un rattachement projectif. On compare ensuite la position de ces ancêtres projectifs dans l'arbre et on retient celui qui se trouve à une position plus profonde dans l'arbre. Il est désigné donc le nouveau gouverneur de son descendant non projectif et le suffixe *-nproj* est ajouté à l'étiquette de la dépendance en question. D'après cette démarche, l'exemple original 48a correspond à la structure projectivisée montrée en 48b. L'apprentissage de Talismane est effectué sur les données ainsi modifiées, et c'est ce type de structure qui est attendu comme sortie du module de parsing. C'est ensuite le module de rattachement non-projectif qui identifie le véritable gouverneur de ces relations. Notons cependant que le succès du module de rattachement dépend directement de la capacité du module de parsing de reproduire les étiquettes modifiées : dans leur absence, le module de rattachement non-projectif ne se déclenche pas.

(48) a.



'Notre décision était sage.'

b.



idem

Par ailleurs, à la différence de la méthode de Nivre & Nilsson (2005), le module de rattachement non projectif de Talismane ne prévoit pas de rétablissement des têtes non projectives pour toutes les dépendances *-nproj* produites par le parser. Son algorithme se focalise sur le *splitting* à l'intérieur d'un GN (cf. section 11.5.2). Comme mentionné ci-dessus, ce module a été développé dans le cadre d'un travail spécifique sur le serbe (Miletic & Urieli, 2017). Par conséquent, A. Urieli avait décidé de viser le type de non-projectivité le plus fréquent dans notre corpus. Concrètement, l'algorithme cherche un nom ou un pronom gouverné par le verbe principal, ainsi qu'un adjectif séparé du nom (ou du pronom) par un dépendant du verbe principal et compatible avec lui d'après ses traits morphosyntaxiques. Si cette démarche aboutit à une identification non ambiguë, on considère que le nom est le véritable gouverneur de l'adjectif et celui-ci lui est rattaché.

Cette méthode a certes une couverture limitée par rapport aux différents types de non-projectivité présentés dans la section 11.5. Cependant, elle s'est montrée assez efficace pour compenser au moins en partie l'avantage des parsers capables de traiter la non-projectivité par défaut. Les résultats détaillés sont donnés dans la section 11.6.3.

11.6.2 Parsing par graphes du parser MST

Nous avons entraîné le parser MST sur le même échantillon de corpus que Talismane, à savoir sur la section *test* du corpus de 81 000 tokens décrit dans la section 9.2.1. Comme mentionné ci-dessus, le parser MST a déjà été utilisé par (Agić et al., 2013b) sur le croate et le serbe. Nous avons donc repris les paramètres d'apprentissage et de parsing identifiés comme optimaux dans ce travail : un entraînement non projectif avec exploitation des traits d'apprentissage du deuxième ordre, fait en 10 itérations avec sélection de 5 meilleurs arbres générés pour la création des contraintes [4]. Suite à cet apprentissage, il suffit de lancer le parsing avec les mêmes paramètres concernant la non-projectivité et l'ordre des traits utilisés pour que l'outil produise des dépendances non projectives.

Quant aux données d'apprentissage, Agić et al. (2013b) exploitent les étiquettes des parties du discours de base et les étiquettes morphosyntaxiques détaillées (colonnes CPOS et POS du format CoNLL-X), sans faire appel aux traits morphosyntaxiques atomiques. Étant donné que notre corpus dispose également de ce dernier type d'information, nous avons décidé de l'utiliser [5]. Nous fournissons donc à MST parser les mêmes traits atomiques qui ont été mis à la disposition de Talismane : le genre, le nombre, le cas, la personne et la forme verbale. En revanche, à la différence de Talismane, qui puise ces informations dans le lexique morphosyntaxique, le parser MST les prend dans le corpus d'apprentissage. Notons également que Talismane n'exploite pas les étiquettes morphosyntaxiques détaillées, contrairement au parser MST. Cela signifie que les deux parsers n'ont pas été entraînés dans des conditions identiques, mais ceci n'est pas problématique dans le cadre de la présente étude, puisque notre intérêt porte sur les capacités maximales de ces deux parsers. Aussi, plutôt que de leur imposer des conditions d'apprentissage identiques, nous cherchons à proposer à chacun des outils les conditions qui garantissent son fonctionnement optimal. Cela nous permet de considérer que les performances observées dans ces évaluations représentent leur limite supérieure sur nos données.

4. Cela se traduit par les paramètres de la ligne de commande suivants : *decode-type:non-proj order:2 iters:10 k-best:5*.

5. Une évaluation rapide a été faite par rapport à la configuration exacte d'Agić et al. (2013b). Nous avons constaté que l'ajout des traits atomiques apportait une amélioration des scores légère mais consistante sur les sections *dev* et *test* (jusqu'à +0,32 en LAS et jusqu'à +0,17 en UAS).

11.6.3 Analyse globale des résultats quantitatifs

Si l'on observe les résultats globaux des deux parsers présentés dans le tableau 11.5, on constate que Talismane obtient systématiquement des scores plus élevés, indépendamment de la section du corpus. La différence est tout de même plus prononcée sur la section *dev* : quasiment 3 points, aussi bien en LAS qu'en UAS, alors qu'elle est inférieure à 1 pour les deux scores sur la section *test*.

Section	Tokens	Talismane		MST parser	
		LAS	UAS	LAS	UAS
dev	4051	88,42	91,43	85,61	88,57
test	7803	87,54	90,94	86,62	90,43

TABLE 11.5 – Talismane *vs* MST parser : résultats globaux

Section	Dép. non proj.	Talismane		MST parser	
		LAS (occur.)	UAS (occur.)	LAS (occur.)	UAS (occur.)
dev	58	51,72 (30)	51,72 (30)	60,34 (35)	60,34 (35)
test	69	34,78 (24)	36,23 (25)	39,13 (27)	40,58 (28)

TABLE 11.6 – Talismane *vs* MST parser : résultats sur les dépendances non projectives

En revanche, quand on limite l'évaluation aux dépendances non projectives, c'est le parser MST qui réalise systématiquement des scores plus élevés (cf. tableau 11.6). Il dépasse Talismane de quasiment 10 points en LAS et en UAS sur la section *dev*, alors que sur la section *test*, son avantage est de 5 points en LAS et de 4 points en UAS. Il est également intéressant de noter que pour chacun des deux parsers la différence entre LAS et UAS globaux est petite : en effet, ces scores sont identiques sur la section *dev* pour Talismane aussi bien que pour le parser MST, alors que sur la section *test*, les deux outils réussissent à rattacher un token en plus (UAS) par rapport à ceux qui ont été bien rattachés et bien labellisés (LAS). Ceci n'est pas étonnant pour Talismane, vu le fonctionnement de son module pseudo-projectif. En revanche, ce fait est plus intéressant dans le cas du parser MST car il montre que le rattachement labellisé et le rattachement non labellisé des dépendances non projectives sont d'une difficulté comparable pour cet outil.

Ces résultats indiquent que le parser MST est moins performant que Talismane sur les relations projectives. Ceci est confirmé par un examen des résultats des deux parsers sur un sous-ensemble des étiquettes syntaxiques, présenté dans le tableau 11.7 [6].

6. Nous analysons les étiquettes ayant au moins 10 occurrences dans chacune des deux sections du corpus, pour lesquelles la différence entre MST parser et Talismane était supérieure à 1 point et orientée de la même manière (positive ou négative) sur les deux sections.

Étiquette	*dev*			*test*		
	Talismane	MST	diff	Talismane	MST	diff
Ap	64,71	66,67	1,96	27,59	44,44	16,85
ExtraPred	78,95	82,05	3,1	66,67	69,33	2,66
ObjDir	90,91	92,86	1,95	89,61	91,14	1,53
ObjIndirCas	90,57	92,31	1,74	84,44	91,53	7,09
ConjCoord	91,44	88,04	-3,4	92,74	91,37	-1,37
Coord	88,32	81,72	-6,6	89,04	82,55	-6,49
DepEx_Dep	77,78	57,14	-20,64	73,17	62,22	-10,95
Juxt	37,04	20,69	-16,35	45,07	42,11	-2,96
PredRel	88,37	73,87	-14,5	88,14	71,86	-16,28

TABLE 11.7 – Talismane *vs* MST parser : F-mesure pour un sous-ensemble des étiquettes. *diff* = MST – Talismane.

Si l'on observe les différences en termes de f-mesure pour ces étiquettes, les résultats sont disparates, comme l'indiquent les valeurs tantôt positives (en faveur de MST), tantôt négatives (en faveur de Talismane) de la différence entre les valeurs de f-mesure. Même si Talismane est globalement meilleur sur les dépendances projectives, chaque analyseur présente des atouts spécifiques. Le parser MST est plus performant sur certaines relations infra-propositionnelles comme l'apposition (Ap), les extraprédicatifs (ExtraPred), l'objet direct (ObjDir) et l'objet indirect casuel (ObjIndirCas). En revanche, il gère moins bien les structures complexes comme la coordination (ConjCoord et Coord), les relatives (PredRel), la juxtaposition (Juxt) et l'ellipse (DepEx_Dep). Cette observation est surprenante, étant donné que certaines de ces structures (notamment les relatives, la juxtaposition et l'ellipse) impliquent souvent des relations à longue distance. Or, ce sont typiquement les parsers à graphes qui sont plus performants sur ce type de dépendances (cf. McDonald & Nivre, 2011). Il est possible que ces résultats aient été en partie conditionnés par les traitements choisis pour ces phénomènes : comme indiqué dans la section 5.2.9, nous avons adopté le traitement de la coordination qui s'est montré le plus adapté pour Talismane. Il est donc possible qu'une autre manière d'annoter ce phénomène aurait favorisé les performances du parser MST. Reste le fait que, dans le cadre de notre évaluation, les différences négatives entre MST et Talismane sont généralement plus marquées que les différences positives, ce qui correspond à l'image présentée par les scores globaux.

11.6.4 Analyse d'erreurs : constructions non projectives maîtrisées par Talismane et MST parser

Quand on compare les performances de Talismane et du parser MST sur différents types de structures non projectives présentes dans les échantillons d'évaluation, on constate que

la majorité des occurrences bien traitées pour les deux parsers relèvent du *splitting*. Ceci est parfaitement logique pour Talismane, vu que son module de rattachement non projectif vise spécifiquement cette construction. Dans le cas du parser MST, cela s'explique sans doute par la fréquence du *splitting* : c'est la structure non projective la plus fréquente aussi bien dans la section *train* que dans les sections *test* et *dev* (cf. section 11.5.1).

dev	Occur.	Talismane		MST	
		No.	%	No.	%
splitting	32	27	84,38	29	90,63
dans le GN	28	27	96,43	27	96,43
dans le GP	4	0	0,00	2	50,00
permutation trans-prop.	9	0	0,00	1	11,11
mouvement *wh-*	1	0	0,00	0	0,00
extraposition	5	0	0,00	2	40,00
autre	8	3	37,50	3	37,50
erreur annot.	3	0	0,00	0	0,00
Total	58	30	51,72	35	60,34

TABLE 11.8 – Performances par type de structure non projective - section *dev*

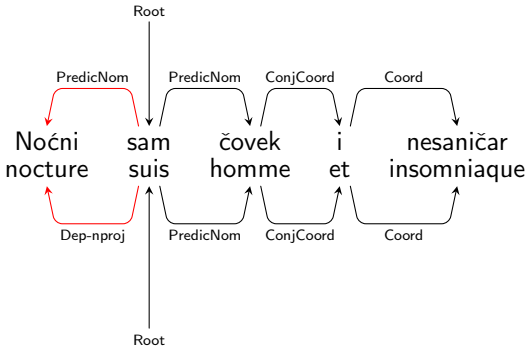
test	Occur.	Talismane		MST	
		No.	%	No.	%
splitting	35	23	65,71	24	68,57
dans le GN	29	23	79,31	22	75,86
dans le GP	3	0	0,00	2	66,67
dans le GA	3	0	0,00	0	0,00
permutation trans-prop.	12	0	0,00	0	0,00
extraposition	4	0	0,00	1	25,00
topicalisation	2	0	0,00	0	0,00
pronom négatif	2	0	0,00	0	0,00
autre	13	1	7,69	2	15,38
erreur annot.	1	0	0,00	0	0,00
Total	69	24	34,78	27	39,13

TABLE 11.9 – Performances par type de structure non projective - section *test*

Si l'on examine de plus près les résultats de Talismane, deux observations principales s'imposent. Premièrement, le parser réussit à bien traiter quelques occurrences de non-projectivité qui ne relèvent pas du *splitting* (cf. ligne *autre* dans les tableaux 11.8 et 11.9). Un examen manuel de ces cas de figure montre qu'il s'agit des configurations qui coïncident par hasard avec les conditions de rattachement définies par le module pseudo-projectif. Deuxièmement, on constate que Talismane n'a pas une couverture parfaite de

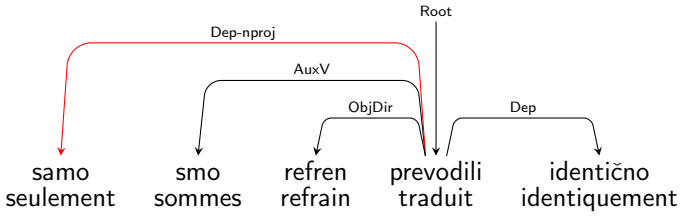
la construction qu'il vise : s'il récupère 96,43 % des cas de *splitting* à l'intérieur d'un GN sur la section *dev*, son taux de réussite est de 79,31 % sur la section *test*. Parmi les 7 occurrences de cette construction que Talismane a mal annotées, on en retrouve 5 qui ont reçu une annotation initiale projective erronée : comme l'étiquette suffixée Dep-nproj n'a pas été utilisée par le parser, le module de post-traitement n'a pas détecté ces occurrences. Il s'agit notamment des cas de figure où le clitique séparant l'adjectif du nom est une forme du verbe *biti* 'être' utilisé comme verbe principal et où l'adjectif est identifié comme un prédicatif de ce verbe (cf. exemple 49a, où l'annotation erronée produite par le parser est montrée au dessus de la phrase, alors que l'annotation projective que le parser aurait dû produire est donnée en dessous). Dans ces exemples, le nom et l'adjectif sont systématiquement annotés tous les deux comme PredicNom. Comme un verbe ne peut avoir qu'un dépendant de ce type, on peut envisager de faire appel à la capacité de Talismane d'intégrer des règles symboliques et de définir des contraintes par rapport à l'utilisation de cette étiquette. Si cette manipulation permettait d'annoter correctement ce type de phrases, la tâche du module de rattachement non projectif pourrait en être facilitée.

(49) a.



'Je suis un homme nocturne et un insomniaque.'

b.



'C'est seulement le refrain que nous traduisions à l'identique.'

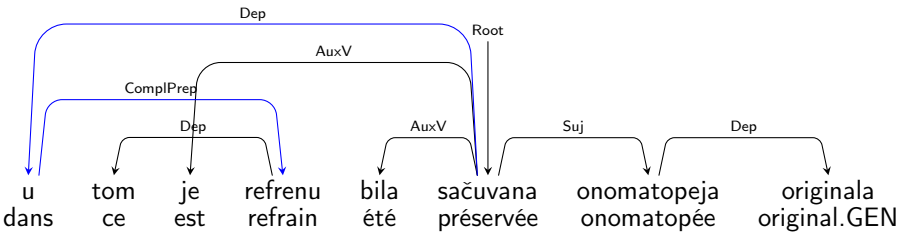
Nous avons également identifié deux exemples avec l'annotation projective correcte qui n'ont pas été traités par le module de rattachement. Dans les deux cas, le dépendant séparé du gouverneur nominal est une particule d'emphase plutôt qu'un adjectif (cf. exemple 49b). Ce cas de figure n'a pas été inclus dans le module de post-traitement afin d'éviter une sur-correction : les particules en tête de phrase peuvent réellement dépendre du verbe principal, et comme elles sont invariables, on ne peut pas se servir des traits morphosyntaxiques pour

vérifier si la forme en question dépend du nom. Ce type de structure reste donc en dehors de la couverture prévue par Talismane.

Il est intéressant de noter que les occurrences du *splitting* à l'intérieur d'un GN qui ont été mal annotées par le parser MST relèvent des mêmes cas de figure : sur 8 occurrences au total, une correspond à la structure présentée dans l'exemple 49b, alors que 6 autres sont du même type que l'exemple 49a. L'occurrence restante représente une structure de coordination mal annotée.

Le parser MST s'est également montré capable d'annoter correctement la dépendance non projective entre la préposition et son complément dans les cas du *splitting* à l'intérieur d'un GP (cf. exemple 50, où les dépendances non projectives correctement reproduites par le parser MST sont indiquées en bleu). Sur 7 occurrences de cette construction au total, le parser MST en traite correctement 4, alors que dans les 3 cas restants, il récupère au moins la non-projectivité à l'intérieur du GN.

(50)



'Dans ce refrain était preservée l'onomatopée de l'original.'

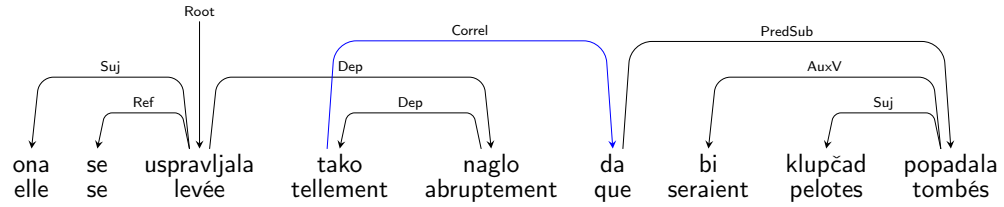
En revanche, l'outil n'a pas maîtrisé le *splitting* à l'intérieur du GA. Cela est possiblement dû à la fréquence moins élevée de ce type de construction : comme indiqué dans la section 11.5.2, ce type de *splitting* représente seulement 16,4 % de toutes les constructions de *splitting* dans notre corpus.

Quant à l'extraposition, le parser MST a annoté correctement 3 occurrences sur 9 présentes dans les échantillons d'évaluation. Il est intéressant de noter que ces occurrences correspondent aux deux types spécifiques d'extraposition évoqués dans la section 11.5.5 : deux d'entre elles représentent des consécutives introduites par un intensifieur qui fonctionne comme un corrélatif (cf. exemple 51a), alors que la troisième correspond à une construction comparative introduite par *nego* 'que' (cf. exemple 51b) [7]. Comme la majorité des exemples mal annotés relèvent d'autres types d'extraposition, le parser semble avoir acquis une maîtrise de ces deux structures particulières. Néanmoins, parmi les exemples mal annotés on trouve également une occurrence d'une consécutive avec corrélatif. Afin

7. Dans cet exemple, la construction comparative a été mal rattachée par le parser MST : elle est gouvernée par le deuxième conjoint, alors qu'elle est relative à toute la coordination et qu'elle devrait être gouvernée par le premier conjoint. En revanche, la dépendance non projective entre le comparatif et le corrélatif est bien traitée.

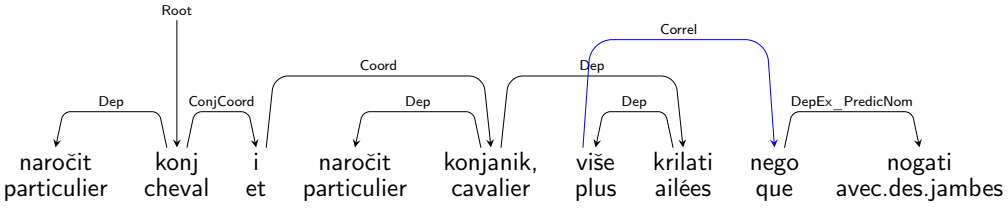
d'évaluer plus précisément la capacité du parser MST à traiter ces structures, il serait nécessaire de mettre en place des échantillons d'évaluation avec plus d'exemples d'extraposition.

(51) a.



'Elle se levait si abruptement que les pelotes tombaient.'

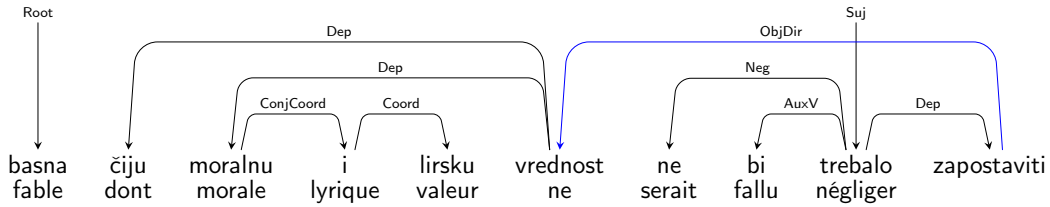
b.



'un cheval et un cavalier particuliers, avec plus d'ailes que de jambes'

Enfin, commentons encore le fait que le parser MST traite bien une seule occurrence du *long-distance scrambling* sur les 21 présentes dans les deux échantillons d'évaluation (cf. exemple 52) [8]. Or, cette construction est relativement fréquente dans le corpus d'apprentissage (33 occurrences), notamment en comparaison avec les constructions d'extraposition analysées ci-dessus (6 comparatives, 10 consécutives). Ce taux de réussite très bas sur cette structure indiquerait donc qu'il s'agit d'une construction intrinsèquement difficile pour le parser MST, mais des évaluations plus ciblées sont nécessaires pour le confirmer.

(52)



'une fable dont il ne faudrait pas négliger la valeur morale et lyrique'

8. On remarque que dans cet exemple la relative et la construction infinitive sont mal annotées par le parser MST. Cependant, la dépendance trans-prépositionnelle entre l'infinitif et son objet direct est bien traitée.

11.6.5 Discussion

D'après les éléments présentés ci-dessus, le parser MST s'est clairement montré plus efficace que Talismane dans le traitement des structures non projectives dans notre corpus. Au-delà de ses scores globaux plus élevés sur ce type de dépendances, il couvre plusieurs types de non-projectivité : comme Talismane, il maîtrise le *splitting* à l'intérieur du GN, mais à la différence de celui-ci, il réussit également à bien annoter le *splitting* à l'intérieur du GP, ainsi que deux types d'extraposition, et une occurrence du *long-distance scrambling*. Malheureusement, la fréquence peu élevée de certaines de ces constructions (notamment de l'extraposition) dans les échantillons d'évaluation fait qu'il est difficile d'avoir une estimation fiable de ses performances. Une évaluation sur des échantillons plus riches en structures de ce type serait utile dans ce sens.

Malgré cette incertitude quant à ses performances quantitatives, ces résultats confirment que le parser MST a la capacité de traiter les structures non projectives citées ci-dessus. Talismane, comme déjà indiqué, ne peut en traiter qu'une (le *splitting* à l'intérieur du GN), et par ailleurs, sa couverture de cette structure n'est pas parfaite. En plus, la performance de Talismane dépend fortement de la structure des échantillons annotés : sur des textes où le *splitting* serait moins fréquent par rapport aux autres types de non-projectivité, Talismane serait moins performant.

Cependant, Talismane reste plus efficace que MST sur la totalité de nos échantillons d'évaluation (dépendances projectives et non projectives confondues). Ceci est notamment dû au fait que le parser MST perd par rapport à Talismane sur des structures complexes comme la coordination et les relatives. Le parsing par graphes et le parsing par transitions se sont donc montrés complémentaires, ce qui va dans le sens des observations globales de McDonald & Nivre (2011). L'éventuel choix entre ces deux outils dépendrait donc de l'exploitation envisagée des annotations : le parser MST serait plus adapté à des utilisations visant en particulier les structures non projectives, alors que Talismane serait un meilleur choix si l'on cible la qualité des dépendances projectives. Si l'on souhaitait exploiter les points forts des deux méthodes, on pourrait envisager de les combiner : il pourrait s'agir d'une véritable intégration des deux algorithmes à l'instar des expériences présentées dans (McDonald & Nivre, 2011), d'un système de vote entre les résultats des deux parsers, voire d'un simple post-traitement qui combinerait les sorties des deux outils.

11.7 Bilan et conclusions

Dans ce chapitre, nous avons proposé un premier profil formel et linguistique de la non-projectivité en serbe basé sur le corpus ParCoTrain-Synt. Notre analyse a montré que le serbe exhibe une proportion d'arbres non projectifs comparable à celle des autres

langues slaves, bien que le nombre de dépendances non projectives soit moindre. Une caractéristique par laquelle le serbe se démarque des autres langues observées est le fait qu'il admet plus facilement les discontinuités créées par des sous-arbres disjoints (cf. la proportion d'arbres avec *maximum edge degree* au-delà de 1). L'analyse des structures linguistiques sous-jacentes a montré que la non-projectivité en serbe relève de formes de discontinuité déjà connues, comme les constructions scindées (*splitting*), le comportement des mots en *qu-* (*wh-fronting*), l'extraposition et la permutation trans-propositionnelle des dépendants du verbe (*long-distance scrambling*). Nous avons également vu que certains types de non-projectivité trouvés dans notre corpus existent aussi dans d'autres langues : les constructions de scission ont également été observées en tchèque, et aussi bien le tchèque que le hindi admettent la permutation trans-propositionnelle des dépendants d'une construction à contrôle. D'une manière plus générale, les remarques de Mambrini & Passarotti (2013) par rapport à l'importance du comportement des clitiques pour les structures non projectives en grec ancien se sont avérées pertinentes pour le serbe aussi : dans notre corpus, les clitiques (et plus spécifiquement, les enclitiques) ont un rôle important dans différents types de non-projectivité, notamment dans les constructions de scission et dans le comportement des mots en *qu-*. Nos expériences en parsing ont confirmé les observations de McDonald & Nivre (2011) sur la complémentarité des parsers par graphes et des parsers par transitions : le parser MST s'est montré plus efficace sur les dépendances non projectives, mais Talismane a été plus performant sur les dépendances projectives, ce qui lui a permis d'atteindre les scores globaux plus élevés. Les deux modèles utilisés sont disponibles à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources/tree/master/Models`.

Quant à la suite de ce travail, plusieurs pistes immédiates s'ouvrent. Tout d'abord, il sera nécessaire d'examiner de plus près le comportement des enclitiques, en cherchant notamment à quantifier d'une manière plus précise leur participation aux structures non projectives. Ensuite, dans le domaine du parsing, nous effectuerons des entraînements et des évaluations orientées vers la non-projectivité dans l'objectif d'améliorer les performances des parsers. Cela implique la constitution d'échantillons qui ciblent les structures non projectives. Enfin, nous vérifierons si nos observations initiales se confirment sur d'autres types de corpus.

Plus globalement, ce travail nous a permis de positionner le serbe par rapport aux autres langues quant à la non-projectivité, que ce soit en fonction de ses propriétés formelles ou linguistiques. Il illustre encore une fois que l'étude de ce phénomène peut servir de point d'appui pour des analyses typologiques et contrastives.

En plus de cette perspective de comparaison avec d'autres langues, nous avons également apporté des éclairages par rapport à la variété et à la fréquence des structures

non projectives en serbe. À notre connaissance, ce phénomène n'avait pas été étudié de ce point de vue dans le cadre de la linguistique serbe.

Il faut souligner également que cette étude était possible grâce à l'existence de ParCoTrain-Synt, un corpus annoté syntaxiquement. Cette ressource ouvre un accès aux exemples par extraction automatique ou semi-automatique et permet par ailleurs de quantifier les résultats. L'intérêt du corpus est d'autant plus évident dans les expériences de parsing, qui auraient été impossibles sans lui. Le travail présenté ici illustre donc la double utilité d'un tel corpus, aussi bien pour le TAL que pour la linguistique.

Conclusion

Dans le cadre de cette thèse, nous avons constitué un ensemble de ressources pour le traitement automatique du serbe. Il s'agit en premier lieu du treebank ParCoTrain-Synt, qui contient 101 000 tokens annotés en morphosyntaxe, en lemmes et en syntaxe de dépendances. Les trois couches d'annotation sont documentées en détail dans les guides d'annotation correspondants. Nous avons également confectionné le lexique ParCoLex, doté de 7 millions d'entrées provenant de 157 000 lemmes différents. Enfin, en exploitant ces deux ressources, nous avons développé des modèles pour le parsing, mais aussi pour l'étiquetage et pour la lemmatisation. Les deux modèles de parsing développés dépassent l'état de l'art actuel en parsing du serbe : Talismane, un parser à transitions, réalise 87,5 points en LAS et 91,2 points en UAS, alors que le parser MST, basé sur les graphes, obtient 86,6 points en LAS et 86,0 points en UAS. Les meilleurs résultats précédents sont de 81,5 points en LAS et de 86,0 points en UAS (Agić & Ljubešić, 2015). Toutes les ressources citées sont librement diffusées à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources`.

Ce travail a pu être réalisé dans le cadre d'une thèse en premier lieu grâce à la méthode que nous avons adoptée. Cette méthode intègre deux éléments de base : l'annotation agile et l'exploitation de la préannotation automatique. Plus particulièrement, elle exploite un *bootstrapping* itératif multicouches. Le travail est effectué sur des échantillons de corpus successifs. Le prétraitement automatique est effectué en cascade, et la sortie de chaque outil est manuellement validée avant d'être transmise à l'outil suivant. Chaque cycle d'annotation finit par une étape d'évaluation, dédiée notamment à un retour d'expérience des annotateurs. Ainsi, les remarques des annotateurs peuvent être incluses dans les guides d'annotation si nécessaire.

Nous avons également mis à l'épreuve l'utilité des ressources constituées dans différents cadres applicatifs. Notamment, le corpus ParCoTrain-Synt et le lexique ParCoLex ont été exploités pour le développement du modèle de parsing de Talismane mentionné ci-dessus. Ce travail a également permis d'étudier l'impact des informations morphosyntaxiques sur le parsing du serbe. Ensuite, le corpus ParCoTrain-Synt a servi de base pour la première étude empirique de la position du groupe adjectival en serbe, ainsi que pour la première

analyse des structures non-projectives dans cette langue. Dans le cadre de cette dernière étude, le corpus a également été utilisé pour l'évaluation contrastive des parsers Talismane et MST sur le serbe, avec un accent particulier sur les structures non projectives.

En ce qui concerne les ressources constituées, la conclusion la plus importante concerne les conditions d'apprentissage assurées par notre corpus. Comme nous l'avons vu, les résultats de Talismane obtenus avec ParCoTrain-Synt dépassent de manière importante ceux du parser Mate obtenus avec le corpus SETimes (Agić & Ljubešić, 2015). Or, les résultats de Talismane ont été obtenus dans des conditions *a priori* plus difficiles : notre corpus est littéraire, alors que SETimes est journalistique, et notre jeu d'étiquettes syntaxiques est beaucoup plus étendu (67 étiquettes vs 15). Une comparaison plus directe est possible avec le parser MST : cet outil a été évalué sur le corpus SETimes (cf. Agić et al., 2013b), mais il obtient des résultats plus élevés sur ParCoTrain-Synt. Nous en tirons donc la conclusion que ParCoTrain-Synt assure de meilleurs conditions d'apprentissage que SETimes. Cela indique que nos décisions relatives aux principes d'annotation sont valides. Tout d'abord, malgré sa taille importante, notre jeu d'étiquettes syntaxiques s'est avéré adapté au parsing. Nos résultats rejoignent ainsi ceux de Mille et al. (2012). Nos choix relatifs à l'annotation morphosyntaxique se sont également montrés pertinents : l'exploitation de notre jeu d'étiquettes raisonné réparti sur plusieurs couches d'annotation a permis d'améliorer les résultats du parsing.

Plusieurs perspectives s'ouvrent pour la suite du travail sur ces ressources. Premièrement, comme nous disposons maintenant de tous les éléments nécessaires, l'annotation du volet serbe du corpus ParCoLab peut être mise en place. Par ailleurs, étant donné les pistes de recherche qu'ouvre le projet UD, nous considérons une conversion de notre treebank vers ce formalisme. Cependant, l'annotation selon le schéma UD ne supprimera pas l'annotation existante : elle serait ajoutée comme une couche d'annotation supplémentaire.

Quant aux améliorations possibles du corpus existant, nous envisageons de reprendre certains points de l'annotation syntaxique, notamment le traitement des datifs, des réflexifs et des relations sous-spécifiées en Dep. Le travail sur ce dernier point a déjà été entamé dans le cadre de l'étude sur le groupe adjectival (cf. chapitre 10). Il sera également nécessaire d'examiner le traitement des unités polylexicales : une tokénisation qui les prend en compte pourrait avoir des effets bénéfiques sur le parsing.

Notons encore qu'un travail d'extension et de diversification de ParCoTrain-Synt est déjà en cours à l'Université de Belgrade : un travail de Master 2 que nous co-encadrons avec D. Stosic et V. Stanojevic vise la création d'un échantillon journalistique de 30 000 tokens qui sera ajouté au corpus.

Quant aux apports méthodologiques de cette thèse, nous constatons que la démarche que nous avons proposée s'est montrée très efficace. Nos évaluations de la vitesse des anno-

tateurs montrent clairement que la préannotation automatique a apporté une accélération importante de l'annotation. En même temps, les analyses régulières du travail des annotateurs nous ont facilité le maintien de la qualité de l'annotation. Les résultats de nos annotateurs indiquent également que l'effort investi dans leur préparation a été crucial. Au-delà du fait de les former à leur tâche, nous les avons incités à s'approprier le projet et avons insisté sur les effets concrets de leur travail. Cela a donné comme résultat un niveau de motivation et d'implication élevé tout au long de leur participation au projet. Notre méthode a également l'avantage de ne pas exiger une équipe étendue, notamment en ce qui concerne le nombre de personnes ayant des responsabilités de gestion et d'expertise scientifique : si une personne combinant les compétences en linguistique et en TAL est disponible, elle peut assurer plusieurs rôles. Nous-mêmes avons assuré ceux de gestionnaire des campagnes, d'annotateur expérimenté, et de taliste.

Le travail sur ParCoTrain-Synt nous a cependant amenée à identifier deux points dans cette méthode qui méritent d'être améliorés. Premièrement, il faut assurer des mécanismes plus efficaces pour la création des guides d'annotation. D'après notre expérience, il est préférable d'établir une première version des guides assez rapidement, pour pouvoir ensuite la tester sur les données et la compléter ainsi. Il faudrait donc inclure dans la méthode un mécanisme qui permette d'optimiser le temps accordé à cette étape de travail.

Nous souhaitons également inclure une phase d'adaptation des annotateurs de manière explicite. Leur travail est facilité par le fait de leur accorder un temps non seulement pour s'approprier les guides, mais aussi pour maîtriser la tâche d'annotation. Cet élément devrait donc être considéré comme prioritaire dans la méthode.

L'intérêt de ces deux évolutions pourra être testé dans un avenir immédiat : en effet, notre méthode sera transposée sur l'occitan dans le but de constituer un treebank pour cette langue. Ce travail sera effectué dans le cadre du projet ParCoLaF, qui vise l'ouverture du corpus ParCoLab aux langues de France, dont l'occitan sera le premier. Cette initiative est soutenue par un projet DGLFLF dans le cadre de l'APN « Langues et numérique » 2017.

En ce qui concerne la mise à l'épreuve des ressources que nous avons créées, les résultats ont été positifs. Nos expériences en exploitation des traits morphosyntaxiques confirment encore une fois que ce type d'information est utile au parsing des langues à morphologie flexionnelle riche et à ordre des constituants flexible. Nous rejoignons donc les observations d'Agić & Ljubešić (2015) par rapport au croate et au serbe, et plus globalement, la majorité des travaux sur ce sujet. Nos évaluations du parsing des structures non projectives ont montré que Talismane et le parser MST sont complémentaires, ce qui est en accord avec les observations de McDonald & Nivre (2011).

Quant aux exploitations en linguistique théorique, nous souhaitons souligner deux apports principaux. Premièrement, l'utilisation du corpus ParCoTrain-Synt nous a permis de

faire avancer deux questions linguistiques jusqu'alors peu explorées en serbe : la variation de la position du groupe adjectival et les constructions non projectives. Deuxièmement, ces deux études illustrent l'intérêt méthodologique de ParCoTrain-Synt. Dans leur réalisation, les couches d'annotation dont est doté le corpus ont été utilisées avec succès pour repérer différents phénomènes linguistiques, en extraire les occurrences et en recenser et analyser des propriétés. Ce fait montre que ParCoTrain-Synt ouvre la porte aux études empiriques basées sur des analyses quantitatives dans le domaine de la linguistique serbe.

Dans la suite de ce travail, nous explorerons deux pistes principales en TAL. En premier lieu, nous souhaitons mettre en place une évaluation explicite des conditions d'apprentissage sur ParCoTrain-Synt et d'autres corpus, notamment SETimes et les treebanks serbe et croate du projet UD. Cela sous-entend une évaluation systématique de plusieurs parsers différents sur les corpus sélectionnés. Ce qui nous intéresse en particulier, c'est de savoir si les jeux d'étiquettes syntaxiques plus restreints présentent un avantage par rapport au nôtre.

Nous allons également consacrer un effort à l'optimisation de l'étiquetage automatique. Comme nous l'avons vu dans le chapitre 9, Talismane atteint une exactitude de 94 % avec un entraînement de base. Pour pallier l'effet de la propagation d'erreurs de ce niveau au parsing, nous souhaitons améliorer ses performances. Après une analyse d'erreurs détaillée, nous pourrons envisager différentes mesures correctives, telles l'élaboration de ressources lexicales complémentaires, l'optimisation des traits d'apprentissage au niveau de l'étiquetage, ou encore l'élaboration des règles symboliques visant des problèmes précis. Les expériences d'A. Urieli avec les deux dernières méthodes ont donné des résultats prometteurs sur le français (Urieli, 2013).

Si l'on revient à la dimension linguistique de ce travail, une question importante se pose par rapport aux deux études effectuées : les effets observés sont-ils dus aux propriétés du corpus utilisé (et notamment à son genre) ou bien s'agit-il de phénomènes généralisables ? L'extension de ParCoTrain-Synt par un échantillon journalistique mentionnée ci-dessus nous permettra d'apporter une partie de la réponse à cette interrogation.

En attendant la finalisation de l'échantillon journalistique, nous souhaitons mettre en place une étude globale du poids syntaxique et de la minimisation de la longueur des dépendances en serbe. Il s'agira d'une méthode quantitative et computationnelle, qui nous permettra d'évaluer les effets de ces deux principes dans notre corpus. Quant au travail sur les structures non projectives, un prolongement naturel consistera à décrire plus précisément l'importance des clitiques pour ce phénomène. Il s'agira notamment de quantifier leur participation à différentes constructions détectées.

Pour conclure, nous constatons que les résultats de cette thèse ont fait avancer le traitement automatique du serbe, que ce soit du point de vue de la disponibilité des ressources ou des performances en traitement automatique. Il en est de même avec la linguistique

théorique, dans laquelle ce corpus permet des recherches nouvelles, et facilite notamment l'application d'une approche empirique et quantitative, dont l'application est rare pour l'étude du serbe. Cela confirme que le TAL et la linguistique ne sont pas forcément impossibles à concilier : une annotation équilibrée et raisonnée ouvre des possibilités intéressantes pour des recherches théoriques, tout en maintenant des conditions d'apprentissage propices à l'entraînement d'un parser.

À la clôture de ce travail, nous allons nous engager dans une étape de valorisation des ressources constituées afin qu'elles soient exploitées et enrichies au-delà du cadre de cette thèse. Leur diffusion libre est un premier pas dans ce sens. Cependant, nous accordons une importance particulière à l'appropriation de nos ressources par la communauté des linguistes travaillant sur le serbe. Dans ce but, dans le cadre du projet ParCoLab, une interface d'interrogation est en cours de développement, qui permettra de faire appel aux différentes annotations présentes dans notre corpus. Ceci facilitera l'accès aux informations linguistiques encodées dans nos ressources, les rendant ainsi utiles à une communauté plus large.

Bibliographie

Anne Abeillé and Nicolas Barrier. Enriching a French Treebank. In *Proceedings of the Fourth International Conference on Language Ressources and Evaluation (LREC2004)*, pages 2233–2236, Lisbon, Portugal, 2004. European Language Resources Association (ELRA).

Anne Abeillé and Danièle Godard. The syntactic structure of French auxiliaries. *Language*, 78(3) :404–452, 2002.

Anne Abeillé and Danielle Godard. La position de l'adjectif épithète en français : le poids des mots. *Recherches linguistiques de Vincennes*, (28) :9–32, 1999.

Anne Abeillé, Lionel Clément, and R. Reyes. Talana annotated corpus : The first results. In *International Conference on Language Ressources and Evaluation (LREC1998)*, pages 993–998, Granada, Spain, 1998. European Language Resources Association (ELRA).

Anne Abeillé, Lionel Clément, and François Toussenel. Building a treebank for French. In *Treebanks*, pages 165–187. Springer, 2003.

Gilles Adda, J Mariani, P Paroubek, and M Rajman. Action GRACE – Mise en place du paradigme d'Evaluation – Application au domaine de l'analyse morpho-syntaxique. In *Proceedings of the 2nd Conférence sur le Traitement Automatique des Langues Naturelles (TALN95)*, Marseille, France, 1995. Association pour le Traitement Automatique des Langues (ATALA).

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers) (ACL2013)*, volume 2, pages 206–211, Sofia, Bulgaria, 2013. Association for Computational Linguistics (ACL).

Stergos Afantenos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Lydia-Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Péry-Woodley, Laurent Prévot, et al. An empirical resource for discovering cognitive principles of discourse

organisation : the ANNODIS corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC2012)*, pages 2727–2734, Istanbul, Turkey, 2012. European Language Resources Association (ELRA).

Željko Agić and Nikola Ljubešić. The SETimes.HR Linguistically Annotated Corpus of Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC2014)*, pages 1725–1727, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Željko Agić and Nikola Ljubešić. Universal Dependencies for Croatian (that work for Serbian, too). In *The Proceedings of the 5th Workshop on Balto-Slavic Natural Language Processing (BSNLP2015)*, pages 1–8, Hissar, Bulgaria, 2015. Special Interest Group on Slavic Natural Language Processing (SIGSLAV).

Željko Agić and Danijela Merkler. Three syntactic formalisms for data-driven dependency parsing of Croatian. In *The Proceedings of the 16th International Conference on Text, Speech and Dialogue (TSD2013)*, pages 560–567, Pilsen, Czech Republic, 2013. Springer.

Željko Agić, Daša Berović, Danijela Merkler, and Marko Tadić. Croatian Dependency Treebank 2.0 : New Annotation Guidelines for Improved Parsing. In *Ninth International Conference on Language Resources and Evaluation (LREC2014)*, pages 2313–2319, Reykjavik, Iceland, 2014.

Željko Agić, Nikola Ljubešić, and Daša Berović. Lemmatization and morphosyntactic tagging of Croatian and Serbian. In *4th Biennial International Workkshop on Balto-Slavic Natural Language Processing (BSNLP 2013)*, pages 48–57, Sofia, Bulgaria, 2013a.

Željko Agić, Danijela Merkler, and Daša Berović. Parsing Croatian and Serbian by using Croatian dependency treebanks. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL2013)*, pages 22–33, Seattle, Washington, USA, 2013b.

Bea Alex, Claire Grover, Rongzhou Shen, and Mijail Kabadjov. Agile corpus annotation in practice : An overview of manual and automatic annotation of CVs. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 29–37, Uppsala, Sweden, 2010. Association for Computational Linguistics.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 1 : Long Papers (ACL2016)*, pages 2442–2452, Berlin, Germany, 2016. ACL.

Juri Apresjan, Igor Boguslavsky, Boris Iomdin, Leonid Iomdin, Andrei Sannikov, and Victor Sizov. A syntactically and semantically tagged corpus of Russian : State of the art and prospects. In *Proceedings of the 5th International Language Resources and Evaluation Conference (LREC2006)*, pages 1378–1381, Genoa, Italy, 2006.

Jurij Apresjan, Igor Boguslavskij, Leonid Iomdin, Alexandre Lazurskij, Vladimir Sannikov, and Leonid Tsinman. ETAP-2 : the linguistics of a machine translation system. *Meta : Journal des traducteurs/Meta : Translators' Journal*, 37(1) :97–112, 1992.

Jurij D Apresjan, Igor M Boguslavskij, Leonid L Iomdin, Alexandre V Lazurskij, Vladimir Z Sannikov, and Leonid L Tsinman. Système de traduction automatique {ETAP}. *La Traductique P. Bouillon et A. Clas, ed. Montréal, Les Presses de l'Université de Montréal,-AUPELF/UREF*, pages 377–391, 1993.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4) :555–596, 2008.

Nart B Atalay, Kemal Oflazer, Bilge Say, et al. The annotation process in the Turkish treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreteted Corpora (LINC2003)*, volume 37, pages 33–38, Budapest, Hungary, 2003. Association for Computational Linguistics (ACL).

Giuseppe Attardi. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL2006)*, pages 166–170, New York City, NY, USA, 2006. Special Interest Group on Natural Language Learning (SIGNLL).

Collin F Baker, Charles J Fillmore, and John B Lowe. The Berkeley Framenet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL1998) and 17th International Conference on Computational Linguistics – Volume 1*, pages 86–90, Montréal, Quebec, Canada, 1998. Association for Computational Linguistics (ACL).

Miguel Ballesteros and Joakim Nivre. Maltoptimizer : an optimization tool for MaltParser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–62, Avignon, France, 2012. Association for Computational Linguistics (ACL).

Eugenija Barić, Mijo Lončarić, Dragica Malić, Slavko Pavešić, Mirko Peti, Vesna Zečević, Marija Znika, et al. *Hrvatska gramatika*. Zagreb : Školska knjiga, 1995.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb) :1137–1155, 2003.

Daša Berović, Željko Agić, and Marko Tadić. Croatian dependency treebank : Recent development and initial experiments. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1902–1906, Istanbul, Turkey, 2012.

Riyaz Ahmad Bhat and Dipti Misra Sharma. A dependency treebank of Urdu and its evaluation. In *Proceedings of the Sixth Linguistic Annotation Workshop (LAW 2012)*, pages 157–165, Jeju Island, South Korea, 2012a. Association for Computational Linguistics (ACL).

Riyaz Ahmad Bhat and Dipti Misra Sharma. Non-projective structures in Indian language treebanks. In *Proceedings of the 11th Workshop on Treebanks and Linguistic Theories (TLT11)*, pages 25–30, Lisbon, Portugal, 2012b.

Alan W Black and Keiichi Tokuda. The Blizzard Challenge-2005 : Evaluating corpus-based speech synthesis on common datasets. In *The Proceedings of the Ninth European Conference on Speech Communication and Technology*, pages 77–80, Lisbon, Portugal, 2005. International Speech Communication Association (ISCA).

Manuel Bodirsky, Marco Kuhlmann, and Mathias Möhl. Well-nested drawings as models of syntactic structure. In James Rogers, editor, *Tenth Conference on Formal Grammar and Ninth Meeting on Mathematics of Language*, pages 195–213. CSLI Publications, 2005.

Igor Boguslavsky. SynTagRus–a Deeply Annotated Corpus of Russian. *Les émotions dans le discours-Emotions in Discourse*, page 367, 2014.

Igor Boguslavsky, Ivan Chardin, Svetlana Grigorieva, Nikolai Grigoriev, Leonid L Iomdin, Leonid Kreidlin, and Nadezhda Frid. Development of a Dependency Treebank for Russian and its Possible Applications in NLP. In *Proceedings of the 3rd International Conference on Language Ressources and Evaluation (LREC2002)*, pages 852–856, Las Palmas, Canary Islands, Spain, 2002a. LREC.

Igor Boguslavsky, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Kreidlin, and Nadezhda Frid. Dependency treebank for Russian : Concept, tools, types of information. In *Proceedings of the 18th conference on Computational linguistics-Volume 2 (COLING2002)*, pages 987–991, Taipei, Taiwan, 2002b. Association for Computational Linguistics (ACL).

Bernd Bohnet. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the 13th Conference on Computational Natural Language Learning : Shared Task (CoNLL2009)*, pages 67–72, Boulder, Colorado, USA, 2009. Association for Computational Linguistics.

Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING2010)*, pages 89–97, Beijing, China, 2010. Association for Computational Linguistics.

Francis Bond, Sanae Fujita, and Takaaki Tanaka. The Hinoki syntactic and semantic treebank of Japanese. *Language Resources and Evaluation*, 42(2) :243–251, 2008.

Johan Bos and Jennifer Spenader. An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, 45(4) :463–494, 2011.

Željko Bošković. Adjectival escapades. In *Proceedings of the 22nd Workshop on Formal Approaches to Slavic Linguistics (FASL2013)*, volume 21, pages 1–25, Hamilton, Ontario, Canada, 2013.

Denis Bouchard. The distribution and interpretation of adjectives in French : A consequence of Bare Phrase Structure. *Probus*, 10(2) :139–183, 1998.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT2002)*, volume 168, pages 1–17, Sozopol, Bulgaria, 2002.

Thorsten Brants. Estimating Markov Model Structures. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, volume 2, pages 893–896, Philadelphia, PA, USA, 1996. IEEE.

Thorsten Brants. Inter-annotator agreement for a german newspaper corpus. In *Proceedings of the 2nd International Conference on Language Ressources and Evaluation*, Athens, Greece, 2000a. European Language Resources Association (ELRA).

Thorsten Brants. TnT : a Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLC'00)*, pages 224–231, Seattle, Washington, 2000b. Association for Computational Linguistics (ACL).

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. Syntactic annotation of a German newspaper corpus. In *Treebanks*, pages 73–87. Springer, 2003.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2) :123–140, 1996.

Joan Bresnan. *Lexical-Functional Syntax*. Blackwell Publishers, Malden, MA, 2001.

Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL2006)*, pages 149–164, New York City, USA, 2006. Association for Computational Linguistics (ACL).

Alicia Burga, Simon Mille, and Leo Wanner. Looking behind the scenes of syntactic dependency corpus annotation : Towards a motivated annotation schema of surface-syntax in Spanish. In *Proceedings of the 1st International Conference on Dependency Linguistics (Depling2011)*, pages 104–114, Barcelona, Spain, 2011.

Marie Candito and Djamé Seddah. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL2010)*, pages 76–84, Los Angeles, CA, USA, 2010. Association for Computational Linguistics (ACL).

Marie Candito, Benoît Crabbé, and Mathieu Falco. Dépendances syntaxiques de surface pour le français. Technical report, Paris 7, 2009.

Marie Candito, Benoît Crabbé, and Pascal Denis. Statistical French dependency parsing : treebank conversion and first results. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC2010)*, pages 1840–1847, La Valetta, Malta, 2010a. European Language Resources Association (ELRA).

Marie Candito, Joakim Nivre, Pascal Denis, and Enrique Henestroza Anguiano. Benchmarking of statistical dependency parsers for French. In *Proceedings of the 23rd International Conference on Computational Linguistics : Posters (COLING2010)*, pages 108–116, Beijing, China, 2010b. Association for Computational Linguistics.

Jean Carletta. Assessing agreement on classification tasks : the kappa statistic. *Computational linguistics*, 22(2) :249–254, 1996.

Xavier Carreras. Experiments with a higher-order projective dependency parser. In *Proceedings of EMNLP-CoNLL 2007*, pages 957–961, 2007.

Xavier Carreras, Isaac Chao, Lluis Padró, and Muntsa Padró. FreeLing : An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004)*, pages 239–242, Lisbon, Portugal, 2004. European Language Resources Association (ELRA).

František Čermák and Alexandr Rosen. The case of InterCorp, a multilingual parallel corpus. *International Journal of Corpus Linguistics*, 17(3) :411–427, 2012.

Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL2000)*, pages 132–139, Seattle, Washington, USA, 2000. Association for Computational Linguistics (ACL).

Wanxiang Che, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. Multilingual dependency-based syntactic and semantic parsing. In *Proceedings of the 13th conference on computational natural language learning (CoNLL2009) : shared task*, pages 49–54, Boulder, CO, USA, 2009. Special Interest Group on Natural Language Learning (SIGNLL).

Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pages 740–750, Doha, Qatar, 2014.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pages 1724–1734, Doha, Qatar, 2014.

Noam Chomsky. On wh-movement. *Formal syntax*, pages 71–132, 1977.

Noam Chomsky. *Some Concepts and Consequences of the Theory of Government and Binding*. MIT press, 1982.

Noam Chomsky. *Lectures on Government and Binding : The Pisa lectures*. Walter de Gruyter, 1993.

Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, MA, 1995.

Grzegorz Chrupała. Simple data-driven context-sensitive lemmatization. *Procesamiento del Lenguaje Natural*, (37) :121–127, 2006.

Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. Learning morphology with Morfette. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC2008)*, pages 2362–2367, Marrakech, Morocco, 2008. European Language Resources Association (ELRA).

Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA, 1988. Association for Computational Linguistics.

Guglielmo Cinque. *The syntax of adjectives : A comparative study*, volume 57. MIT press, 2010.

Lionel Clement and Alexandra Kinyon. Chunking, marking and searching a morpho-syntactically annotated corpus for French. In *Proceedings of ACIDCA*, Sfax, Tunisia, 2000. IEEE & ELRA.

Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics (EACL1997)*, pages 16–23, Madrid, Spain, 1997. Association for Computational Linguistics (ACL).

Michael Collins. Discriminative training methods for Hidden Markov Models : Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (EMNLP2002)*, pages 1–8, Philadelphia, PA, USA, 2002. Association for Computational Linguistics (ACL).

Michael Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4) :589–637, 2003.

Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. A statistical parser for Czech. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL1999)*, pages 505–512, College Park, Maryland, USA, 1999. Association for Computational Linguistics (ACL).

Ronan Collobert and Jason Weston. A unified architecture for Natural Language Processing : Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, Helsinki, Finland, 2008. ACM.

Matthieu Constant and Anthony Sigogne. MWU-aware part-of-speech tagging with a CRF model and lexical resources. In *Proceedings of the Workshop on Multiword Expressions : from Parsing and Generation to the Real World*, pages 49–56, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

Greville Corbett. *The World's Major Languages*, chapter Serbo-Croat, pages 391–490. Oxford University Press, 1987.

Koby Crammer, Ofer Dekel, Shai Shalev-shwartz, and Yoram Singer. Online passive-aggressive algorithms. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1229–1236. MIT Press, 2004.

Emanuela Cresti and Massimo Moneglia. *C-ORAL-ROM : integrated reference corpora for spoken romance languages*, volume 15. John Benjamins Publishing, 2005.

Paul Cubberley. *Russian : A linguistic introduction*. Cambridge University Press, 2002.

Éric De La Clergerie, Benoît Sagot, Lionel Nicolas, and Marie-Laure Guénot. FRMG : évolutions d'un analyseur syntaxique TAG du français. In *Journée de l'ATALA sur : Quels analyseurs syntaxiques pour le français ?*, Paris, France, 2009.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. From raw text to Universal Dependencies – Look, no tags ! In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies (CoNLL2017)*, pages 207–217, Pisa, Italy, 2017. Special Interest Group on Natural Language Learning (SIGNLL).

Ralph Debusmann, Denys Duchier, and Geert-Jan M Kruijff. Extensible dependency grammar : A new methodology. In *Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar*, pages 70–76, Geneva, Switzerland, 2004.

Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC2009)*, pages 110–119, Hong Kong, 2009.

Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a lexicon for state-of-the-art POS tagging. *Language resources and evaluation*, 46(4) :721–736, 2012.

Matthew S Dryer. The Greenbergian word order correlations. *Language*, pages 81–138, 1992.

Denys Duchier and Ralph Debusmann. Topological dependency trees : A constraint-based account of linear precedence. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL2001)*, pages 180–187, Toulouse, 2001. Association for Computational Linguistics (ACL).

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing*, pages 334–343, Beijing, China, 2015. Association for Computational Linguistics (ACL).

Sašo Džeroski, Tomaž Erjavec, Nina Ledinek, Petr Pajas, Zdenek Žabokrtsky, and Andreja Žele. Towards a Slovene dependency treebank. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 1388–1391, Genoa, Italy, 2006.

Jason M Eisner. Three new probabilistic models for dependency parsing : An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345, Copenhagen, Denmark, 1996. Association for Computational Linguistics.

Tania Ellbogen, Florian Schiel, and Alexander Steffen. The BITS speech synthesis corpus for German. In *Proceedings of the 4th International Conference on Language Ressources and Evaluation*, volume 47, pages 40–44, Lisbon, Portugal, 2004.

Tomaž Erjavec. MULTEXT-East : morphosyntactic resources for Central and Eastern European languages. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC 2012)*, volume 46, pages 131–142, Istanbul, Turkey, 2012. Springer.

Tomaž Erjavec and Nancy Ide. The MULTEXT-East Corpus. In *First International Conference on Language Resources and Evaluation (LREC1998)*, volume 98, pages 971–974, Granada, Spain, 1998.

Tomaž Erjavec, Darja Fišer, Simon Krek, and Nina Ledinek. The JOS linguistically tagged corpus of Slovene. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*, La Valetta, Malta, 2010.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear : A library for large linear classification. *Journal of Machine Learning Research*, 9 :1871–1874, 2008.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. Apertium : a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2) :127–144, 2011.

Karën Fort. *Les ressources annotées, un enjeu pour l'analyse de contenu : vers une méthodologie de l'annotation manuelle de corpus.* PhD thesis, Université Paris-Nord-Paris XIII, 2012.

Karën Fort and Benoît Sagot. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the 4th Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden, 2010. Association for Computational Linguistics.

Kilian Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. Because size does matter : The Hamburg dependency treebank. In *Proceedings of the Language Resources and Evaluation Conference 2014*, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA) (2014).

W Nelson Francis and Henry Kučera. Brown corpus manual. Technical report, Brown University, 1979.

Alexander Fraser, Helmut Schmid, Richárd Farkas, Renjing Wang, and Hinrich Schütze. Knowledge sources for constituent parsing of German, a morphologically rich and less-configurational language. *Computational Linguistics*, 39(1) :57–85, 2013.

Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33) :10336–10341, 2015.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

Gerald Gazdar, Klein Ewald, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.

Kim Gerdes and Sylvain Kahane. Word order in German : A formal dependency grammar using a topological hierarchy. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 220–227. Association for Computational Linguistics, 2001.

Andrea Gesmundo and Tanja Samardžić. Lemmatising Serbian as category tagging with bidirectional sequence classification. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC 2012)*, pages 2103–2106, 2012.

Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009) : Shared Task*, pages 37–42, Boulder, Colorado, USA, 2009. Association for Computational Linguistics.

Daniel Gildea and David Temperley. Do grammars minimize dependency length ? *Cognitive Science*, 34(2) :286–310, 2010.

Jesús Giménez and Lluis Marquez. SVMTool : A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004)*, pages 43–46, Lisbon, Portugal, 2004.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard : Telephone speech corpus for research and development. In *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520. ICASSP-92, IEEE, 1992.

Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1) :1–309, 2017.

Yoav Goldberg and Reut Tsarfaty. A single framework for joint morphological segmentation and syntactic parsing. In *Proceedings of ACL*, 2008.

Yoav Goldberg, Reut Tsarfaty, Meni Adler, and Michael Elhadad. Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and EM-HMM-based lexical probabilities. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 327–335, Athens, Greece, 2009. Association for Computational Linguistics.

Annette M Green. Kappa statistics for multiple raters using categorical classifications. In *Proceedings of the 22nd annual SAS User Group International conference*, volume 2, page 4, 1997.

Spence Green, Marie-Catherine de Marneffe, and Christopher D Manning. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1) :195–227, 2013.

Thomas Groß. Clitics in Dependency Morphology. In *Proceedings of the 1st International Conference on Dependency Linguistics (DepLing 2011)*, pages 56–68, Barcelona, Spain, 2011.

Thomas Groß and Timothy Osborne. Toward a practical dependency grammar theory of discontinuities. *SKY Journal of Linguistics*, 22 :43–90, 2009.

Thomas Groß and Timothy Osborne. The Dependency Status of Function Words : Auxiliaries. In *Proceedings of the 3rd International Conference on Dependency Linguistics (DepLing2015)*, pages 111–120, Uppsala, Sweden, 2015.

Kristina Gulordava and Paola Merlo. Structural and lexical factors in adjective placement in complex noun phrases across Romance languages. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 247–257, Beijing, China, 2015.

Kristina Gulordava, Paola Merlo, and Benoit Crabbé. Dependency length minimisation effects in short spans : a large-scale analysis of adjective placement in complex noun phrases. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers), volume 2, pages 477–482, Beijing, China, 2015.

Nizar Habash, Reem Faraj, and Ryan Roth. Syntactic annotation in the Columbia Arabic treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, pages 125–132, Cairo, Egypt, 2009.

Jan Hajič. Morphological tagging : Data vs. dictionaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL2000)*, pages 94–101, Seattle, Washington, USA, 2000. Association for Computational Linguistics (ACL).

Jan Hajič. Complex corpus annotation : The Prague dependency treebank. *Insight into Slovak and Czech Corpus Linguistics. Veda Bratislava*, pages 54–73, 2005.

Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, and Alla Bémová. Annotations at analytical level. Instructions for annotators. *UK MFF ÚFAL, Praha, Czech Republic. URL http ://ufal. mff. cuni. cz/pdt2. 0/doc/manuals/en/a-layer/pdf/a-man-en. pdf (2012-03-18)*, 1999.

Jan Hajič, Otakar Smrz, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. Prague Arabic dependency treebank : Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117, Cairo, Egypt, 2004.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. The CoNLL-2009 shared task : Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning : Shared Task*, pages 1–18. Association for Computational Linguistics, 2009.

Jan Hajič. Building a syntactically annotated corpus : The Prague Dependency Treebank. *Issues of valency and meaning*, pages 106–132, 1998.

Eva Hajičová, Jiří Havelka, Petr Sgall, Kateřina Veselá, and Daniel Zeman. Issues of Projectivity in the Prague Dependency Treebank. *Prague Bull. Math. Linguistics*, 81 : 5–22, 2004.

Dilek Z Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4) :381–410, 2002.

Péter Halácsy, András Kornai, and Csaba Oravecz. HunPos : an open source trigram tagger. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 209–212, Prague, Czech Republic, 2007. Association for Computational Linguistics.

Nabil Hathout, Franck Sajous, and Basilio Calderone. GLÀFF, a large versatile French lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC2014)*, pages 1007–1012, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Jiří Havelka. Beyond Projectivity : Multilingual Evaluation of Constraints and Measures on Non-Projective Structures. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 608–615, Prague, Czech Republic, 2007.

John A Hawkins. *A performance theory of order and constituency*, volume 73. Cambridge University Press, 1994.

John A Hawkins. *Efficiency and complexity in grammars*. Oxford University Press on Demand, 2004.

David G Hays. Dependency theory : A formalism and some observations. *Language*, 40 (4) :511–525, 1964.

Peter Hellwig. Dependency unification grammar. In *Proceedings of the 11th Coference on Computational Linguistics*, COLING '86, pages 195–198, Stroudsburg, PA, USA, 1986. Association for Computational Linguistics.

Tomas Holan, Vladislav Kubon, Karel Oliva, and Martin Plátek. Two useful measures of word order complexity. *Processing of Dependency-Based Grammars*, 1998.

Rodney Huddleston. *Introduction to the Grammar of English*. Cambridge University Press, 1984.

Richard A Hudson. *Word grammar*. Blackwell Oxford, 1984.

Nancy Ide and Jean Véronis. MULTEXT : Multilingual text tools and corpora. In *Proceedings of the 15th conference on Computational linguistics-Volume 1 (COLING1994)*, pages 588–592, Kyoto, Japan, 1994. Association for Computational Linguistics.

Lidija Iordanskaja and Igor Mel'čuk. Establishing an inventory of surface-syntactic relations : Valence-controlled surface-syntactic dependents of the verb in French. *Studies in Language*, 2009 :151–234, 2009.

Milka Ivić, editor. *Sintaksa savremenog srpskog jezika*. Institut za srpski jezik SANU, Beograd, 2005.

Bojana Jakovljević, Aleksandar Kovačević, Milan Sečujski, and Maja Marković. A dependency treebank for Serbian : Initial experiments. In *International Conference on Speech and Computer*, pages 42–49, Novi Sad, Serbia, 2014. Springer.

Bart Jongejan and Hercules Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL2009) and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 145–153, Suntec, Singapore, 2009. Association for Computational Linguistics (ACL).

Matjaz Juršic, Igor Mozetic, Tomaz Erjavec, and Nada Lavrac. Lemmagen : Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9) :1190–1214, 2010.

Sylvain Kahane. Grammaires de dépendance formelles et théorie sens-texte. In *Actes de la 8e conférence sur le Traitement Automatique des Langues Naturelles (TALN2001)*, pages 1–62, Tours, France, 2001.

Sylvain Kahane, Alexis Nasr, and Owen Rambow. Pseudo-projectivity : a polynomially parsable non-projective dependency grammar. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 646–652. Association for Computational Linguistics, 1998.

Vladimir Karabalić. Hrvatski imenski predikatni proširak i njemački ekvivalenti. *Suvremena lingvistika*, 55-56 :85–101, 2003.

Radoslav Katičić. *Sintaksa hrvatskoga književnog jezika : nacrt za gramatiku*, volume 61. Jugoslavenska akademija znanosti i umjetnosti, 1986.

ed. Keith, Brown. Encyclopedia of language and linguistics. 2006.

Jin-Dong Kim, Sang-Zoo Lee, and Hae-Chang Rim. Hmm specialization with selective lexicalization. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 121–127, College Park, MD, USA, 1999.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1) :i180–i182, 2003.

Jong-Bok Kim and Peter Sells. *English syntax : An introduction.* CSLI publications, 2008.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, (4) :313–327, 2016.

Göran Kjellmer. *A dictionary of English collocations : Based on the Brown corpus.* Oxford University Press, USA, 1994.

Philipp Koehn. Europarl : A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit X*, volume 5, pages 79–86, Phuket, Thailand, 2005.

Iliyana Krapova. Auxiliaries and complex tenses in Bulgarian. In W. Browne, E. Domisch, N. Kondrašova, and D. Zec, editors, *Annual workshop on Formal approaches to Slavic linguistics. The Cornell meeting*, pages 320–344. Ann Arbor : Michigan Slavic Publications, 1995.

Klaus Krippendorff. *Content Analysis : An introduction to its methodology.* Sage, Beverly Hills, CA., USA, 1980.

Cvetana Krstev. *Processing of Serbian. Automata, Texts and Electronic Dictionaries.* Faculty of Philology of the University of Belgrade, 2008.

Cvetana Krstev and Duško Vitas. Corpus and lexicon-mutual incompletness. In *Proceedings of the Corpus Linguistics Conference*, pages 14–17, Birmingham, UK, 2005.

Cvetana Krstev and Duško Vitas. An aligned English-Serbian corpus. *ELLSIIR Proceedings (English Language and Literature Studies : Image, Identity, Reality)*, 1 :495–508, 2011.

Cvetana Krstev, Gordana Pavlovic-Lazetic, Duško Vitas, and Ivan Obradovic. Using textual and lexical resources in developing Serbian Wordnet. *Romanian Journal of Information Science and Technology*, 7(1-2) :147–161, 2004a.

Cvetana Krstev, Duško Vitas, and Tomaž Erjavec. MULTEXT-East resources for Serbian. In *Zbornik 7. mednarodne multikonference Informacijska druzba IS 2004 Jezikovne tehnologije 9-15 Oktober 2004, Ljubljana, Slovenija, 2004.* Erjavec, Tomaž and Zganec Gros, Jerneja, 2004b.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1) :1–127, 2009.

Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th International Conference on Natural language learning (CoNLL2002)*, volume Volume 20, pages 1–7, Taipei, Taiwan, 2002. Special Interest Group on Natural Language Learning (SIGNLL).

Marco Kuhlmann and Joakim Nivre. Mildly Non-Projective Dependency Structures. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 507–514, Sydney, Australia, 2006. Association for Computational Linguistics.

Marco Kuhlmann and Joakim Nivre. Transition-based techniques for non-projective dependency parsing. *Northern European Journal of Language Technology (NEJLT)*, 2(1) : 1–19, 2010.

Anna Kupść and Jesse Tseng. A new HPSG approach to Polish auxiliary constructions. In S. Müller, editor, *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, pages 253–273. Stanford : CSLI Publications, 2005.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, Williamstown, MA, USA, 2001.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL2010)*, pages 504–513, Uppsala, Sweden, 2010. Association for Computational Linguistics (ACL).

Pierre Le Goffic. Les mots qu- entre interrogation, indéfinition et subordination : quelques repères. *Lexique*, (18), 2007.

Joseph Le Roux, Benoit Sagot, and Djamé Seddah. Statistical parsing of Spanish and data driven lemmatization. In *ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages (SP-Sem-MRL 2012)*, pages 55–61, Jeju, South Korea, 2012.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. Finding function in form : Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, pages 1520–1530, Lisbon, Portugal, 2015. Association for Computational Linguistics.

Haitao Liu. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2) :159–191, 2008.

Nikola Ljubešić and Filip Klubička. {bs, hr, sr} WaC–web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35, Gothenburg, Sweden, 2014.

Nikola Ljubešić, Filip Klubička, Željko Agić, and Ivo-Pavao Jazbec. New inflectional lexicons and training corpora for improved morphosyntactic annotation of Croatian and Serbian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank : Building a large-scale annotated Arabic corpus. In *Proceedings of the NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467, Cairo, Egypt, 2004.

Wolfgang Maier, Sandra Kübler, Daniel Dakota, and Daniel Whyatt. Parsing German : How much morphology do we need ? In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 1–14, Dublin, Ireland, 2014.

Francesco Mambrini and Marco Passarotti. Non-Projectivity in the Ancient Greek Dependency Treebank. In *Proceedings of the 2nd Internantional Conference on Dependency Linguistics (DepLing 2013)*, pages 177–186, Prague, Czech Republic, 2013.

Prashanth Mannem, Himani Chaudhry, and Akshar Bharati. Insights into non-projectivity in Hindi. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 10–17, Singapore, 2009. Association for Computational Linguistics.

Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing (Fifth Printing 2002)*. The MIT Press, 1999.

Malgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiorkowski, and Anna Kupsc. Construction of an HPSG treebank for Polish. In *Proceedings of the ATALA conference*, Cargèse, France, 1999.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English : The Penn Treebank. *Computational linguistics*, 19(2) : 313–330, 1993.

André FT Martins, Dipanjan Das, Noah A Smith, and Eric P Xing. Stacking dependency parsers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP2008)*, pages 157–166, Honolulu, Hawaii, USA, 2008. Association for Computational Linguistics.

André FT Martins, Noah A Smith, and Eric P Xing. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 1-Volume 1*, pages 342–350, Singapore, 2009. Association for Computational Linguistics.

André FT Martins, Miguel B Almeida, and Noah A Smith. Turning on the turbo : Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*, pages 617–622, Sofia, Bulgaria, 2013. Association for Computational Linguistics (ACL).

Yuval Marton, Nizar Habash, and Owen Rambow. Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1) : 161–194, 2013.

Peter H Matthews. Syntax. *Cambridge textbooks in linguistics, Cambridge University Press*, 69 :75, 1981.

Ryan McDonald and Joakim Nivre. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1) :197–230, 2011.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL2005)*, pages 91–98, Sydney, Australia, 2005a. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Joint Conference on Human Language Technology (HLT) and Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–530, Vancouver, Canada, 2005b. Association for Computational Linguistics (ACL).

Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL2006)*, pages 216–220, New York City, USA, 2006. Association for Computational Linguistics.

Ryan T McDonald and Fernando CN Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL2006)*, pages 81–88, Trento, Italy, 2006. Association for Computational Linguistics (ACL).

Igor Aleksandrovič Mel'čuk. *Russkij Jazyk V Modeli "Smysl-Tekst" : The Russian Language in the Meaning-Text Perspective*. Škola" Jazyki russkoj kul'tury", 1995.

Igor Mel'čuk. *Dependency syntax : Theory and practice*. State University Press of New York, 1988.

Igor Mel'čuk and Nikolaj Pertsov. *Surface Syntax of English*. John Benjamins Publishing Company, 1987.

Igor Mel'čuk. Dependency in natural language. *Dependency in linguistic description*, 111 : 1–110, 2009.

Igor Mel'čuk. Levels of dependency in linguistic description : Concepts and problems. *Dependency and valency. An International handbook of contemporary research*, 1 :188–229, 2003.

Bernard Merialdo. Tagging English text with a probabilistic model. *Computational linguistics*, 20(2) :155–171, 1994.

Danijela Merkler, Željko Agić, and Ana Agić. Babel treebank of public messages in Croatian. *Procedia-Social and Behavioral Sciences*, 95 :490–497, 2013.

Christian M Meyer and Iryna Gurevych. OntoWiktionary : Constructing an ontology from the collaborative online dictionary wiktionary. *Semi-Automatic Ontology Development : Processes and Resources : Processes and Resources*, 131, 2012.

Aleksandra Miletic. Annotation morphosyntaxique semi-automatique d'un corpus litéraire serbe. Master's thesis, Université Charles de Gaulle - Lille 3, 2013.

Aleksandra Miletic. Building a morphosyntactic lexicon for Serbian using Wiktionary. In *Sixièmes Journées d'études Toulousaines (JéTou2017)*, pages 30–34, Toulouse, France, 2017.

Aleksandra Miletic and Assaf Urieli. Non-projectivity in Serbian : Analysis of Formal and Linguistic Properties. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling2017)*, pages 135–144, Pisa, Italy, 2017.

Aleksandra Miletic, Dejan Stosic, and Saša Marjanović. ParCoLab : A Parallel Corpus for Serbian, French and English. In K. Ekstein and V. Matousek, editors, *Text, Speech and Dialogue 2017, LNAI 10415*, pages 156–164, Prague, Czech Republic, 2017. Springer.

Jasmina Milićević. Serbian Auxiliary Verbs : Syntactic Heads or Dependents ? In W. Cichocki, editor, *Proceedings of the 31st Annual Conference of the Atlantic Provinces Linguistics Association*, pages 43–53. PAMAPLA 31, 2009.

Simon Mille, Alicia Burga, Gabriela Ferraro, and Leo Wanner. How does the granularity of an annotation scheme influence dependency parsing performance ? In *Proceedings of the 24th International Conference on Computational Linguistics (Posters) (COLING2012)*, pages 839–852, Mumbai, India, 2012.

Simon Mille, Alicia Burga, and Leo Wanner. AncoraUPF : A multi-level annotation of Spanish. In *Proceedings of the 2nd Dependency Linguistics Conference (DepLing2013)*, pages 201–226, Prague, Czech Republic, 2013.

Simonetta Montemagni, Francesco Barsotti, Marco Battista, Nicoletta Calzolari, Ornella Corazzari, Alessandro Lenci, Antonio Zampolli, Francesca Fanciulli, Maria Massetani, Remo Raffaelli, et al. *Treebanks*, chapter Building the Italian syntactic-semantic treebank, pages 189–210. Springer, 2003.

Pavica Mrazović. *Gramatika srpskog jezika za strance*. Izdavačka knjižarnica Zorana Stojanovića, 2009.

Thomas Müller and Hinrich Schütze. Robust morphological tagging with word representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT2015)*, pages 526–536, Denver, Colorado, USA, 2015.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP2013)*, pages 322–332, Seattle, USA, 2013.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. Joint lemmatization and morphological tagging with LEMMING. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal, 2015.

Emmanuel Navarro, Franck Sajous, Bruno Gaume, Laurent Prévot, Hsieh ShuKai, Kuo Tzu-Yi, Pierre Magistry, and Huang Chu-Ren. Wiktionary and NLP : Improving synonymy networks. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP : Collaboratively Constructed Semantic Resources*, pages 19–27, Suntec, Singapore, 2009. Association for Computational Linguistics.

Mariana Neves, Alexander Damaschun, Andreas Kurtz, and Ulf Leser. Annotating and evaluating text for stem cell research. In *Proceedings of the Third Workshop on Building*

and Evaluation Resources for Biomedical Text Mining (BioTxtM 2012) at Language Resources and Evaluation (LREC2012), pages 16–23, Istanbul, Turkey, 2012.

Joakim Nivre. Constraints on non-projective dependency parsing. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–80, Trento, Italy, 2006. ACL.

Joakim Nivre. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4) :513–553, 2008.

Joakim Nivre. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 1*, pages 351–359, Singapore, 2009a. Association for Computational Linguistics.

Joakim Nivre. Parsing Indian languages with MaltParser. In *Proceedings of the ICON09 NLP Tools Contest : Indian Language Dependency Parsing*, pages 12–18, 2009b.

Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08 : HLT)*, pages 950–958, Columbus, Ohio, 2008.

Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106, Sydney, Australia, 2005. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser : A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, volume 6, pages 2216–2219, Genoa, Italy, 2006.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*, pages 915–932. Association for Computational Linguistics (ACL), 2007a.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Maltparser : A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02) :95–135, 2007b.

Joakim Nivre, Igor M Boguslavsky, and Leonid L Iomdin. Parsing the SynTagRus treebank of Russian. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 641–648, Manchester, UK, 2008. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal Dependencies v1 : A multilingual treebank collection. In *Proceedings of the 10th International Conference on Linguistic Resources and Evaluation (LREC2016)*, pages 1659–1666, Portorož, Slovenia, 2016.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. Open-domain anatomical entity mention detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 27–36. Association for Computational Linguistics, 2012.

Csaba Oravecz and Péter Dienes. Efficient stochastic part-of-speech tagging for Hungarian. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC2002)*, pages 710–717, Las Palmas, Canary Islands, Spain, 2002. LREC2002.

György Orosz and Attila Novák. PurePos–an open source morphological disambiguator. In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*, pages 53–63, Wroclaw, Poland, 2012.

Petr Pajas and Jan Štěpánek. Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1 (COLING2008)*, pages 673–680, Manchester, UK, 2008. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank : An annotated corpus of semantic roles. *Computational linguistics*, 31(1) :71–106, 2005.

Patrick Paroubek. Evaluating part-of-speech tagging and parsing. *Evaluation of Text and Speech Systems*, 37 :99, 2007.

Gordana Pavlović-Lažetić, Duško Vitas, and Cvetana Krstev. Towards full lexical recognition. In *Text, Speech and Dialogue (TSD2004)*, pages 179–186. Springer, 2004.

Leticia Anton Pérez, Hugo Gonçalo Oliveira, and Paulo Gomes. Extracting lexical-semantic knowledge from the Portuguese Wiktionary. In *Proceedings of the 15th Portuguese Conference on Artificial Intelligence, EPIA*, pages 703–717, Lisbon, Portugal, 2011.

Mirko Peti. *Predikatni proširak*. Hrvatsko filološko društvo, 1979.

Mirko Peti. Glagolski predikat u imenskom predikatu. *Rasprave : Časopis Instituta za hrvatski jezik i jezikoslovlje*, 30(1) :163–171, 2005.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING2006) and the 44th annual meeting of the Association for Computational Linguistics (ACL2006)*, pages 433–440, Sydney, Australia, 2006. Association for Computational Linguistics (ACL).

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL2016)*, pages 412–418, Berlin, Germany, 2016. Association for Computational Linguistics (ACL).

Joël Plisson, Nada Lavrač, Dunja Mladenić, and Tomaž Erjavec. Ripple down rule learning for automated word lemmatisation. *AI Communications*, 21(1) :15–26, 2008.

Lucie Poláková, Pavlína Jínová, and Jirí Mírovskỳ. Genres in the Prague Discourse Treebank. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC2014)*, pages 1320–1326, Reykjavik, Iceland, 2014.

Carl Pollard and Ivan A Sag. *Head-driven Phrase Structure Grammar*. University of Chicago Press, 1994.

Zoran Popović. Taggers applied on texts in Serbian. *INFOtheca-Journal of Informatics and Librarianship*, 11(2), 2010.

Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3) :130–137, 1980.

Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, and Bonnie Webber. The Penn Discourse TreeBank as a resource for natural language generation. In *Proceedings of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pages 25–32, Birmingham, UK, 2005.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18) :i575–i581, 2012.

Zhu Qi-bo. A quantitative look at the Guangzhou Petroleum English Corpus. Technical report, Norwegian Comwting Centre for the Humanities, 1989.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive grammar of the English language*. Longman, London, 1985.

Martin Rajman, Josette Lecomte, and Patrick Paroubek. Format de description lexicale pour le français. partie 2 : Description morpho-syntaxique. *Rapp. Tech., EPFL & INaLF. GRACE GTR-3-2.1*, 1997.

Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, volume 1, pages 133–142. Philadelphia, PA, USA, 1996.

R Reyes. Un etiqueteur du français inspiré du taggeur de Brill. Technical report, Rapport de stage, Université Paris 7, 1997.

John Robert Ross. *Constraints on variables in Syntax*. PhD thesis, MIT, 1967.

Cornelis J Ruijgh. La place des enclitiques dans l'ordre des mots chez Homère d'après la loi de Wackernagel. *Sprachwissenschaft und Philologie. Jacob Wackernagel und die Indogermanistik heute*, pages 213–233, 1990.

Benoît Sagot. DeLex, a freely-avaible, large-scale and linguistically grounded morphological lexicon for German. In *Proceedings of the 9th International Conference Language Resources and Evaluation (LREC2014)*, pages 2778–2784, Reykjavik, Iceland, 2014.

Benoît Sagot and Héctor Martínez Alonso. Improving neural tagging with lexical information. In *Proceedings of the 15th International Conference on Parsing Technologies (IWPT2017)*, pages 25–31, Pisa, Italy, 2017.

Benoît Sagot. Etiquetage multilingue en parties du discours avec MElt. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2016*, Paris, France, 2016.

Franck Sajous, Nabil Hathout, and Basilio Calderone. Glàff, un gros lexique à tout faire du français. In *Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013)*, pages 285–298, Les Sables d'Olonne, France, 2013.

Tanja Samardžić, Mirjana Starović, Željko Agić, and Nikola Ljubešić. Universal Dependencies for Serbian in Comparison with Croatian and Other Slavic Languages. In *Proceedings of the he 6th Workshop on Balto-Slavic Natural Language Processing (BSNLP2017)*, pages 39–44, Valencia, Spain, 2017. Association for Computational Linguistics (ACL).

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, Royaume-Uni, 1994.

Milan Sečujski. *Automatic part-of-speech tagging of texts in the Serbian language*. PhD thesis, Faculty of Technical Sciences, Novi Sad, Serbia, 2009.

Djamé Seddah, Marie Candito, and Benoit Crabbé. Cross parser evaluation and tagset variation : A French treebank study. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 150–161, Paris, France, 2009. Association for Computational Linguistics.

Djamé Seddah, Grzegorz Chrupała, Özlem Çetinoğlu, Josef Van Genabith, and Marie Candito. Lemmatization and lexicalized statistical parsing of morphologically rich languages : the case of French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 85–93, Los Angeles, California, USA, 2010. Association for Computational Linguistics.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, et al. Overview of the SPMRL 2013 shared task : cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL2013)*, Seattle, Washington, USA, 2013. Association for Computational Linguistics.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. Introducing the SPMRL Shared Task on Parsing Morphologically-Rich Languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland, 2014. Dublin City University.

Rico Sennrich and Beat Kunz. Zmorge : A German Morphological Lexicon Extracted from Wiktionary. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1063–1067, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Petr Sgall, Ladislav Nebeskỳ, Alla Goralcıková, and Eva Hajicová. *A functional approach to syntax : in generative description of language.* Elsevier, New. York, 1969.

Petr Sgall, Eva Hajicová, and Jarmila Panevová. *The meaning of the sentence in its semantic and pragmatic aspects.* Springer Science & Business Media, 1986.

Libin Shen, Giorgio Satta, and Aravind Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL2007)*, volume 7, pages 760–767, Prague, Czech Republic, 2007.

Anna Siewierska and Ludmila Uhliřova. An overview of word order in Slavic languages. *Constituent Order in the Languages of Europe*, 20(1) :105, 1998.

Milorad Simić. Srpski elektronski rečnik. http ://www.rasprog.com, 2005.

Kiril Simov and Petya Osenova. Practical annotation scheme for an HPSG treebank of Bulgarian. In *Proceedings of the 4th International Workshop on Linguistically Interpreteted Corpora (LINC)*, pages 17–24, Budapest, Hungary, 2003.

Kiril Ivanov Simov, Petya Osenova, Milena Slavcheva, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krassimira Ivanova, Alexander Simov, and Milen Kouylekov. Building a linguistically interpreted corpus of Bulgarian : the BulTreeBank. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1729–1736, Las Palmas, Canary Islands, Spain, 2002. European Language Resources Association (ELRA).

John M Sinclair. *Looking up : An account of the COBUILD project in lexical computing and the development of the Collins COBUILD English language dictionary.* Collins Elt, 1987.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. Annotating unrestricted German text. *Fachtagung der Sektion Computerlinguistik der Deutschen Gesellschaft für Sprachwissenschaft*, 1997.

Otakar Smrž, Viktor Bielicky, and Jan Hajič. Prague Arabic dependency treebank : A word on the million words. In *Proceedings of the Workshop on Arabic and Local Languages (LREC 2008)*, pages 16–23, Marrakech, Morocco, 2008.

Živojin Stanojčić and Ljubomir Popović. *Gramatika srpskog jezika.* Zavod za udžbenike, 2012.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. The JRC-Acquis : A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Ressources and Evaluation (LREC2006)*, pages 2142–2147, Genoa, Italy, 2006.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT : a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL2012)*, pages 102–107, Avignon, France, 2012. Association for Computational Linguistics (ACL).

Dejan Stosic. Le rôle des préfixes dans l'expression des relations spatiales. eléments d'analyse à partir des données du serbo-croate et du français. *Cahiers de grammaire*, 26 : 207–228, 2001.

Michael Strube and Simone Paolo Ponzetto. WikiRelate ! Computing semantic relatedness using Wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with Bidirectional Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pages 14–25, Doha, Qatar, 2014.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL2008)*, pages 159–177, Manchester, UK, 2008. Association for Computational Linguistics.

Jan Svartvik. *The London-Lund corpus of spoken English : Description and research.* Number 82. Lund University Press, 1990.

Zsolt Szántó and Richárd Farkas. Special techniques for constituent parsing of morphologically rich languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL2014)*, pages 135–144, Gothenburg, Sweden, 2014. Association for Computational Linguistics (ACL).

Marko Tadić. Building the Croatian Dependency Treebank : the initial stages. *Suvremena lingvistika*, (63) :85–92, 2007.

Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 64–71. Association for Computational Linguistics (ACL), 1997.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. Ancora : Multilevel annotated corpora for catalan and spanish. In *Proceedings of the 6th International Conference on Language Ressources and Evaluation (LREC2008)*, Marrakech, Morocco, 2008.

Isabelle Tellier, Iris Eshkol-Taravella, Yoann Dupont, and Ilaine Wang. Peut-on bien chunker avec de mauvaises étiquettes POS ? In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN2014)*, pages 125–136, Marseilles, France, 2014. Association pour le Traitement Automatique des Langues (ATALA).

David Temperley. Minimization of dependency length in written English. *Cognition*, 105 (2) :300–333, 2007.

David Temperley. Dependency-length minimization in natural and artificial languages*. *Journal of Quantitative Linguistics*, 15(3) :256–282, 2008.

Lucien Tesnière. Eléments de syntaxe structurale. 1959.

Paul-Louis Thomas. Bilan des recherches sur l'aspect en serbo-croate. *Revue des Etudes Slaves*, 3(65) :537–550, 1993.

Paul-Louis Thomas. Serbo-croate, serbe, croate..., bosniaque, monténégrin : une, deux..., trois, quatre langues ? *Revue des études slaves*, 66(1) :237–259, 1994.

Paul-Louis Thomas. Remarques sur l'aspect en serbo-croate. In Andrée Borillo, Carl Vetters, and Marcel Vuillaume, editors, *Regards sur l'aspect*, pages 231–243. Rodopi, Amsterdam, 1998.

Juliette Thuilier. *Contraintes préférentielles et ordre des mots en français*. PhD thesis, Université Paris-Diderot-Paris VII, 2012.

Juliette Thuilier, Gwendoline Fox, and Benoît Crabbé. Prédire la position de l'adjectif épithète en français : approche quantitative. *Lingvisticae Investigationes*, 35(1) :28–75, 2012.

Jörg Tiedemann. News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In G. Angelova N. Nicolov, K. Bontchev and R. Mitkov, editors, *Recent advances in natural language processing*, volume 5, pages 237–248, 2009.

Sara Tonelli, Rodolfo Delmonte, and Antonella Bristot. Enriching the Venice Italian Treebank with dependency and grammatical relations. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC2008)*, pages 1920–1924, Marrakech, Morocco, 2008. European Language Resources Association (ELRA).

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180, Edmonton, Canada, 2003. Association for Computational Linguistics (ACL).

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. Statistical parsing of morphologically rich languages (SPMRL) : what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12, Los Angeles, California, USA, 2010. Association for Computational Linguistics (ACL).

Larraitz Uria, Ainara Estarrona, Izaskun Aldezabal, Maria Jesús Aranzabe, Arantza Díaz De Ilarraza, and Mikel Iruskieta. Evaluation of the syntactic annotation in EPEC, the

reference corpus for the processing of Basque. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 72–85. Springer, 2009.

Assaf Urieli. *Robust French syntax analysis : reconciling statistical methods and linguistic knowledge in the Talismane toolkit*. PhD thesis, Université Toulouse le Mirail-Toulouse II, 2013.

Assaf Urieli. Improving the parsing of French coordination through annotation standards and targeted features. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 26–38, Dublin, Ireland, 2014.

Miloš Utvić. Annotating the corpus of contemporary Serbian. In *Proceedings of the INFOtheca '12 Conference*, pages 36–47, 2011.

Leonoor Van der Beek, Gosse Bouma, Rob Malouf, and Gertjan Van Noord. The Alpino dependency treebank. *Language and Computers*, 45(1) :8–22, 2002.

Marianne Vergez-Couret and Assaf Urieli. Analyse morphosyntaxique de l'occitan languedocien : l'amitié entre un petit languedocien et un gros catalan. In *Actes de l'atelier Traitement Automatique des Langues Régionales de France et d'Europe au TALN2015*, Caen, France, 2015. Association pour le Traitement Automatique des Langues (ATALA).

Dusko Vitas and Cvetana Krstev. Intex and Slavonic morphology. *INTEX pour la linguistique et le traitement automatique des langues, Presses Universitaires de Franche-Comté*, pages 19–33, 2004.

Duško Vitas and Cvetana Krstev. Literature and aligned texts. *Readings in Multilinguality*, pages 148–155, 2006.

Ruprecht von Waldenfels. Compiling a parallel corpus of Slavic languages. Text strategies, tools and the question of lemmatization in alignment. In B. Brehmer B, V. Zdanova, and R. Zimny, editors, *Beiträge der Europäischen Slavistischen Linguistik*, volume 9, pages 123–138, 2006.

Holger Voormann and Ulrike Gut. Agile corpus creation. *Corpus Linguistics and Linguistic Theory*, 4(2) :235–251, 2008.

Jacob Wackernagel. Über ein Gesetz der Indogermanischen Wortstellung. *Indogermanische Forschungen*, 1 :333, 1892.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The Penn Chinese TreeBank : Phrase structure annotation of a large corpus. *Natural language engineering*, 11(02) : 207–238, 2005.

Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with Support Vector Machines. In *Proceedings of International Workshop on Parsing Technologies*, volume 3, pages 195–206. IWPT, 2003.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. Conll 2017 shared task : multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, 2017.

Torsten Zesch and Iryna Gurevych. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007)*, pages 1–8, Rochester, USA, 2007.

Torsten Zesch and Iryna Gurevych. Wisdom of crowds versus wisdom of linguists–measuring the semantic relatedness of words. *Natural Language Engineering*, 16(1) : 25–59, 2010.

Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC2008)*, volume 8, pages 1646–1652, Marrakech, Morocco, 2008.

Bojana P. Đorđević. *Izrada osnova formalne gramatike srpskog jezika upotrebom metagramatike (Construction of a formal grammar of Serbian using a metagrammar)*. PhD thesis, University of Belgrade, Serbia, 2017.

Résumé. Au début de cette thèse, aucun corpus annoté syntaxiquement (treebank) n'était disponible pour le serbe. Or, les treebanks annotés manuellement sont une condition *sine qua non* du développement (entraînement et évaluation) d'outils statistiques dédiés à l'annotation syntaxique automatique (parsers). L'existence des parsers performants permet à son tour l'annotation syntaxique de corpus plus larges, qui peuvent ensuite alimenter des recherches en linguistique théorique. De fait, l'absence de ces ressources pour le serbe freine le développement des recherches sur cette langue dans ces deux directions, et plus généralement les efforts visant l'informatisation et la valorisation du serbe.

Afin de combler cette lacune, nous avons constitué un ensemble de ressources pour le traitement automatique du serbe. Il s'agit en premier lieu du treebank ParCoTrain-Synt, qui contient 101 000 tokens annotés en morphosyntaxe, en lemmes et en syntaxe de dépendances. Nous avons également confectionné le lexique ParCoLex, doté de 7 millions d'entrées provenant de 157 000 lemmes différents. En exploitant ces deux ressources, nous avons développé des modèles pour le parsing, pour l'étiquetage et pour la lemmatisation. Toutes les ressources citées sont librement diffusées à l'adresse suivante : `https://github.com/aleksandra-miletic/serbian-nlp-resources`. Les ressources constituées ont également été exploitées dans le cadre de deux études linguistiques, montrant ainsi que le corpus ParCoTrain-Synt ouvre la porte aux études empiriques basées sur des analyses quantitatives dans le domaine de la linguistique serbe.

Abstract. At the beginning of this PhD, no treebank for Serbian was available. However, manually annotated treebanks are an essential resource for developing (training and evaluating) statistical tools for syntactic analysis (parsers). Efficient parsers, in turn, facilitate the annotation of large corpora, which can be used as a basis for research in theoretical linguistics. The lack of these resources for Serbian slows down the research in these two directions. It also hinders the creation of digital resources for Serbian in general.

In order to address this issue, we created a suite of NLP resources for Serbian. Firstly, we created the ParCoTrain-Synt treebank, a 101 000 token corpus, complete with morphosyntactic annotation, lemmatisation and syntactic dependency annotation. We also built the ParCoLex lexicon, containing 7 million entries for 157 000 different lemmas. Using these two resources, we trained models for parsing, morphosyntactic tagging and lemmatisation. All of the above resources are available at the following address : `https://github.com/aleksandra-miletic/serbian-nlp-resources`. We also used these resources in two experiments in Serbian linguistics, demonstrating that the ParCoTrain-Synt treebank is well suited to empirical studies based on quantitative data analysis.