Représentation et analyse automatique des discontinuités syntaxiques dans les corpus arborés en constituants du français

Maximin Coavoux, Benoît Crabbé

Représentation et analyse automatique des discontinuités syntaxiques dans les corpus arborés en constituants du français.

Maximin Coavoux[1, 2] Benoît Crabbé[1, 3]
(1) Laboratoire de Linguistique Formelle (LLF, CNRS)
(2) Univ Paris Diderot, Sorbonne Paris Cité (SPC)
(3) Institut Universitaire de France (IUF)
{mcoavoux,bcrabbe}@linguist.univ-paris-diderot.fr

RÉSUMÉ

Nous présentons de nouvelles instanciations de trois corpus arborés en constituants du français, où certains phénomènes syntaxiques à l'origine de dépendances à longue distance sont représentés directement à l'aide de constituants discontinus. Les arbres obtenus relèvent de formalismes grammaticaux légèrement sensibles au contexte (LCFRS). Nous montrons ensuite qu'il est possible d'analyser automatiquement de telles structures de manière efficace à condition de s'appuyer sur une méthode d'inférence approximative. Pour cela, nous présentons un analyseur syntaxique par transitions, qui réalise également l'analyse morphologique et l'étiquetage fonctionnel des mots de la phrase. Enfin, nos expériences montrent que la rareté des phénomènes concernés dans les données françaises pose des difficultés pour l'apprentissage et l'évaluation des structures discontinues.

ABSTRACT

Representation and parsing of syntactic discontinuities in French constituent treebanks.

The article introduces novel instanciations of three French constituent treebanks in which certain syntactic phenomena responsible for long-distance dependencies are represented with discontinuous constituents. Resulting trees are mildly context-sentitive structures, and can be modeled with, e.g. LCFRS. We show that such structures can be parsed efficiently, by introducing a neural transition-based discontinuous parser, that also performs morphological analysis and functional tagging. Our experiments show that the sparsity of these phenomena in French treebanks makes learning and evaluation of discontinuous structures difficult.

MOTS-CLÉS : Constituants discontinus, analyse syntaxique, *deep learning*.

KEYWORDS: Discontinuous constituents, parsing, deep learning.

1 Introduction

On accepte généralement que les grammaires non contextuelles (CFG) ne sont pas suffisamment expressives pour décrire adéquatement les langues naturelles (Joshi, 1985), contrairement à la classe, plus générale, des grammaires légèrement sensibles au contexte. Ainsi des systèmes de réécriture comme les grammaires d'arbres adjoints (TAG) ou les systèmes linéraires de réécriture hors contexte (LCFRS) autorisent l'utilisation de règles qui permettent de générer des arbres dont les empans (angl. *spans*) sont discontinus (Kallmeyer, 2010). Ce type de système permet non seulement de modéliser

élégamment la syntaxe de langues à ordre des mots relativement libre comme l'allemand mais peut être également utilisé pour modéliser directement des phénomènes d'extraction ou de dislocation avec dépendance à longue distance dans des langues à ordre des mots relativement figé comme le français ou l'anglais. Par exemple, de telles discontinuités se manifestent naturellement en français lorsqu'il s'agit de traiter des questions partielles ou des propositions relatives. La figure 1 (partie droite) illustre chacun de ces deux cas d'extraction où les dépendances à longue distance sont représentées par des constituants discontinus.

Néanmoins, les analyseurs en constituants ne sont pas généralement conçus pour prédire ce type de structure. De fait, l'analyse de structures discontinues a longtemps été réputée trop coûteuse en calculs (et restreinte à des phrases courtes). La représentation de ces phénomènes dans les corpus fait éventuellement appel à une couche d'annotation sous forme de traces indexées (voir section 2) qui est généralement supprimée par les prétraitements de la majorité des analyseurs en constituants qui par hypothèse se restreignent aux grammaires CFG.

Cet article propose une méthode de traitement direct de la discontinuité en analyse syntaxique pour le français. On présente (a) trois corpus en constituants discontinus pour le français et (b) on montre qu'en réutilisant un algorithme d'analyse syntaxique lexicalisé qui permet de traiter le problème efficacement et une méthode d'apprentissage profond multi-tâche qui tire parti d'une modélisation des interfaces de l'analyseur (morphologie et structure fonctionelle), on représente dans le même modèle des constituants discontinus et des dépendances non projectives typées.

On commence par introduire, en section 2, trois corpus en constituants discontinus pour le français obtenus par conversion de corpus existants : le French Treebank (Abeillé *et al.*, 2003, FTB), le Sequoia (Candito & Seddah, 2012a, SEQ) et le French Question Bank (Seddah & Candito, 2016, FQB).

Nous présentons ensuite en section 3 un algorithme d'analyse syntaxique basé sur un système de transitions par décalage-réduction augmenté d'une action GAP (Coavoux & Crabbé, 2017a). L'analyseur utilise un système de pondération basé sur une méthode d'apprentissage profond multi-tâche qui réalise conjointement l'analyse morphologique, l'analyse syntaxique et l'étiquetage fonctionnel. À notre connaissance, il s'agit du premier analyseur en constituants discontinus basé sur un réseau de neurones profond toutes langues confondues [1].

Finalement, nous proposons une série d'expériences en section 4 qui mettent en évidence que la modélisation des dépendances à longue distance pose non seulement un problème théorique de complexité d'analyse mais aussi un problème statistique d'estimation de données éparses.

2 Représentations discontinues pour le français

Sans qu'ils exhibent nécessairement des propriétés de grammaires légèrement sensibles au contexte, les phénomènes d'extraction à longue distance restent difficiles à représenter avec des grammaires non contextuelles. Ainsi, Gazdar (1981) montre que les phénomènes d'extraction se représentent dans une grammaire hors contexte en utilisant un codage connu sous le nom de trait *SLASH*. Dans les corpus arborés, le Penn Treebank (Marcus *et al.*, 1993) utilise par exemple des traces indexées : une catégorie vide (trace) est ajoutée dans la position canoniquement occupée par un élément extrait, et cette catégorie est coindexée avec l'élément déplacé.

1. Le code de l'analyseur syntaxique est librement téléchargeable à l'adresse `github.com/mcoavoux/mtg`.

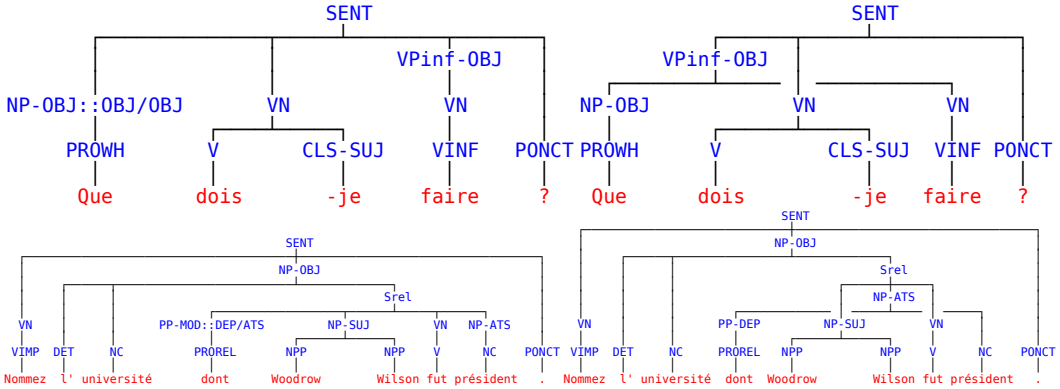


FIGURE 1 – Illustration de la transformation pour deux phrases tirées du FQB.

Si le French Treebank (Abeillé *et al.*, 2003) prend le parti de ne pas coder explicitement les relations d'extraction, Candito & Seddah (2012a) ont ajouté plus récemment l'information en utilisant des chemins fonctionnels (Bresnan, 2001, LFG) : certains constituants sont annotés avec un chemin permettant de retrouver leur position canonique (figure 1, à gauche).

À l'instar de la méthode de Gazdar (1981), ce type d'annotation indirecte permet de représenter les relations d'extraction à longue distance tout en conservant l'hypothèse forte que les structures syntaxiques représentent des dérivations d'une grammaire non contextuelle. En revanche, ces informations sont souvent ignorées des analyseurs syntaxiques en constituants, qui réalisent des prétraitements sur les données (suppression des traces ou des annotations fonctionnelles) et qui s'évaluent sur les données ainsi transformées. Ce choix est en général justifié par des facilités de traitement : comme nous le détaillons ci-dessous, les algorithmes d'analyse syntaxique exacts capables de traiter les discontinuités ont une complexité théorique nettement plus élevée que les algorithmes d'analyse pour les grammaires non contextuelles.

Par contraste, deux corpus en constituants allemands, Negra (Skut *et al.*, 1997) et Tiger (Brants, 1998) ont fait le choix d'annoter directement les discontinuités en autorisant des arbres à empans discontinus, comme illustré sur l'exemple français en figure 1 à droite. Les arbres à constituants discontinus représentent des dérivations de grammaires LCFRS dont la classe de langage est légèrement sensible au contexte, c'est-à-dire des grammaires qui couvrent le cas général pour l'analyse des langues naturelles. Ils permettent notamment de modéliser des phénomènes de variation d'ordre des mots, comme les extrapositions (McCawley, 1982; Levy, 2005).

Dans cet article, nous proposons de représenter directement des phénomènes linguistiques, comme les extractions à longue distance, par des constituants discontinus. Dans la suite de cette section, nous présentons les phénomènes concernés puis nous décrivons la procédure de conversion à partir de corpus existants vers des structures discontinues.

Observations qualitatives Étant données les annotations de Candito & Seddah (2012a), les phénomènes linguistiques pour lesquels l'utilisation de représentations discontinues semble pertinente dans les corpus arborés français relèvent essentiellement de cas d'extraction. Candito & Seddah (2012a) en proposent la classification suivante : les propositions relatives (i), les questions directes (ii) ou

Phénomène	FTB-TRAIN		FQB		SEQUOIA	
Propositions relatives	183	72%	4	5%	36	77%
Questions	8	3%	83	95%	2	4%
Constructions clivées	5	2%	0	0%	4	9%
Dislocations	1	$< 1\%$	0	0%	1	2%
en	57	22%	0	0%	4	9%
Total	254	100%	87	100%	47	100%

TABLE 1 – Phénomènes linguistiques à l'origine des discontinuités.

indirectes (iii), les constructions clivées (iv), les dislocations (v), certains syntagmes prépositionnels cliticisés avec *en* (vi). On illustre chacun de ces cas par un exemple où le syntagme extrait et son gourverneur sont en gras :

(i) *Conseiller municipal socialiste, il était déjà cependant directeur général de la banque **qu**'il va **présider**.* FTB
(ii) ***Pour quel type de logement** puis-je **bénéficier** d'une aide au logement ?* FQB
(iii) *Je suis seulement très curieux de savoir **comment** la Commission pense **rassembler** cet argent.* SEQ
(iv) *C'est donc toute la vie industrielle du bassin de Saint-Dizier, sans oublier les papeteries de Jeand'Heurs, les carrières de Savonnières, **que** les visiteurs du lavoir pourront **découvrir**.* SEQ
(v) ***A un "déterminisme technologique", développé notamment par Alain Touraine, où l'histoire des techniques s'impose à l'organisation du travail et à l'emploi,** on doit **opposer** "une dialectique à trois termes, technologie, organisation et travail".* FTB
(vi) *La crise, tout le monde la sentait, mais ce mois terrible **en** fait prendre la **mesure**.* FTB

Notons que toutes les occurrences de ces phénomènes ne produisent pas nécessairement de discontinuités. En effet, pour qu'une extraction produise une discontinuité, il faut qu'elle soit non locale à la règle de grammaire. Considérons par exemple les deux phrases suivantes :

(i-a) Nommez l'université **dont** Woodrow Wilson fut **président**. (FQB)
(i-b) `(S ...(Srel dont NP-suj VN NP-ats))`
(ii-a) Nommez un tigre **dont la race** est éteinte. (FQB)
(ii-b) `(S ...(Srel dont NP-suj VN ADJ))`

Dans la phrase (i-a), dont l'arbre est donné également en figure 1 (partie inférieure), le site d'extraction de *dont* est situé dans le NP attribut du sujet (ats). Il y a deux syntagmes entre le site d'extraction et la position finale de *dont* (i-b). Par contraste, dans la phrase (ii-a), *dont* est contigu au syntagme dont il est extrait (ii-b).

Nous présentons en table 1 les fréquences des occurrences de chaque phénomène à l'origine d'une discontinuité pour chaque corpus. On observe que les propositions relatives et le clitique *en* sont les deux principales sources de discontinuités dans le FTB et dans le SEQUOIA. Comme cela était attendu, ce sont les questions qui sont à l'origine de la quasi totalité des discontinuités du FQB.

Précisons que d'autres phénomènes que ceux cités ci-dessus sont à l'origine de discontinuités syntaxiques, par exemple les propositions relatives extraposées, ou les incises. Nous nous sommes restreints dans cet article aux extractions pour lesquelles nous disposons des annotations de Candito & Seddah (2012a) et laissons le traitement de ces phénomènes à des travaux futurs.

	FTB-TRAIN	FTB-DEV	FTB-TEST	FQB-ALL	FQB-TRAIN	FQB-DEV	FQB-TEST	SEQUOIA
Tokens	443113	38820	75216	23222	13616	4566	5040	67038
Phrases	14759	1235	2541	2289	1289	500	500	3099
Phrases avec discontinuité	253 (1.71%)	27 (2.19%)	32 (1.26%)	88 (3.84%)	69 (5.35%)	6 (1.2%)	13 (2.6%)	46 (1.48%)
Rang	34	23	30	16	16	7	15	32
Gap-degree	1	1	1	2	2	1	1	1
Constituants	298025	26180	50590	15966	9416	3189	3361	47586
Constituants discontinus	374 (0.13%)	35 (0.13%)	44 (0.09%)	94 (0.59%)	73(0.77%)	7 (0.22%)	14 (0.42%)	70 (0.15%)

TABLE 2 – Données statistiques sur les corpus obtenus.

Méthode de conversion pour le français Nous décrivons succintement la procédure de conversion des données du français depuis l'annotation de Candito & Seddah (2012a) vers une représentation en arbres discontinus avant de présenter quelques statistiques sur les données obtenues. Nous nous intéressons à trois corpus arborés du français : le French Treebank (Abeillé *et al.*, 2003), le Sequoia (Candito & Seddah, 2012b) et le French Question Bank (Seddah & Candito, 2016). Les schémas d'annotations des deux derniers sont basés sur celui du French Treebank et n'en diffèrent que marginalement, de sorte qu'ils peuvent être utilisés facilement pour des expériences d'adaptation de domaine.

Candito & Seddah (2012a) ont annoté des dépendances à longue distance dans ces corpus, sous la forme de chemins fonctionnels. La figure 1 (partie gauche) illustre une telle annotation. Dans la phrase de la partie supérieure de la figure, le NP projeté par le pronom interrogatif *que* est annoté avec le chemin fonctionnel `OBJ/OBJ`. Dans l'arbre projectif, le gouverneur local de *que* est *dois*. Le chemin fonctionnel spécifie le parcours entre ce gouverneur local et le gouverneur longue distance *faire* : le NP doit être rattaché comme objet de l'objet de *dois* (partie droite). La procédure de conversion consiste simplement à changer l'attachement des nœuds annotés par un chemin fonctionnel, en interprétant ce chemin.

Nous présentons quelques statistiques sur les données obtenues en table 2. En proportion, c'est le French Question Bank qui contient le plus de phrases avec une discontinuité (4% des phrases). Pour les autres corpus, 1 à 2 % des phrases sont concernées. Le degré de discontinuité d'un constituant est mesuré par le nombre d'empans contigus (*spans*) dominés par ce constituant (appelé *fan-out* dans le vocabulaire LCFRS). Le *gap-degree* $G(c)$ d'un constituant c est défini comme $G(c) = \text{fan-out}(c) - 1$. Par exemple, le constituant `SENT` dans l'arbre de la figure 1 (partie supérieure droite) a un *gap-degree* de 0, car les terminaux qu'il domine forment une chaîne continue. En revanche, le `VPinf` dans le même arbre a un *gap-degree* de 1, car il domine 2 chaînes, *Que* et *faire* séparées par d'autres terminaux. Soit C l'ensemble des constituants observés dans un treebank, on définit le *gap degree* $G(C)$ d'un corpus arboré par $G(C) = \max_{c \in C} G(c)$. Le rang d'un constituant $R(c)$ mesure le nombre de constituants dominés immédiatement par c. On définit $R(C)$ le rang d'un corpus arboré par $R(C) = \max_{c \in C} R(c)$.

Les discontinuités dans les corpus que nous avons utilisés sont relativement rares. Par comparaison, environ 29 % des phrases du corpus de l'allemand Tiger (Brants, 1998) contiennent une discontinuité (Maier & Lichte, 2011) et environ 20 % de celles de la version discontinue du Penn Treebank (Evang & Kallmeyer, 2011). Une explication possible à cela est que, comme dit plus haut, nous n'avons pas pris en compte tous les phénomènes linguistiques susceptibles de produire des discontinuités syntaxiques. Enfin, les choix d'annotations peuvent aussi avoir un effet sur le nombre de discontinuités. Notamment,

les syntagmes verbaux n'étant pas annotés dans le French Treebank (à l'exception des syntagmes verbaux infinitifs et participiaux), beaucoup d'extractions ne produisent pas de discontinuités [2].

Du point de vue de l'analyse syntaxique automatique, la rareté des arbres discontinus dans les données pose des problèmes en termes d'apprentissage : tant pour l'estimation de valeurs de paramètres que pour l'évaluation qui peut présenter une variance importante (section 4).

3 Analyse syntaxique discontinue

L'analyse automatique des arbres en constituants discontinus pose des problèmes computationnels. En effet, les systèmes linéaire de réécriture hors contexte (LCFRS) binaires, un type de grammaire formel suffisamment expressif pour générer des arbres discontinus, ont une complexité d'analyse exacte en $\mathcal{O}(n^{3f})$ où f est le *fan-out* de la grammaire et n le nombre de mots de la phrase (Kallmeyer, 2010). Pour cette raison, les analyseurs tabulaires exacts se sont souvent restreints aux phrases courtes ou ont eu recours à des heuristiques pour accélérer l'inférence (Kallmeyer & Maier, 2013; van Cranenburgh *et al.*, 2016).

Outre les analyseurs tabulaires, des méthodes fondées sur l'analyse par transitions ont été proposées, soit dans le cadre *easy-first* (Versley, 2014), soit avec des extensions de l'algorithme *shift-reduce* (Maier, 2015; Coavoux & Crabbé, 2017a). Ces méthodes permettent généralement une inférence approximative efficace [3] et peuvent utiliser facilement des classifieurs discriminants avec des traits riches. Elles sont par ailleurs intrinsèquement robustes, c'est-à-dire qu'elles renvoient une analyse pour toute séquence de symboles donnée en entrée et ne permettent pas de rejeter une phrase considérée comme aggrammaticale.

L'analyseur présenté ici adapte le système de transitions SR-GAP que Coavoux & Crabbé (2017a) ont introduit pour l'analyse de l'allemand, en utilisant un système de pondération des analyses qui s'appuie sur un perceptron, au cas de l'apprentissage profond multi-tâches (Caruana, 1997; Collobert *et al.*, 2011).

Après avoir présenté le système de transitions SR-GAP qui traite le cas discontinu en section 3.1, nous introduisons une méthode de pondération des analyses qui s'appuie sur un réseau de neurones profond pour prédire les actions à chaque étape de l'analyse (section 3.2). L'architecture du réseau comporte une étape de codage des mots par un réseau de neurones récurrent bi-directionnel inspiré de Cross & Huang (2016a). Ce modèle neuronal permet de réduire considérablement le nombre de patrons de traits à définir par rapport au perceptron de Coavoux & Crabbé (2017a). Le modèle neuronal instancie également une architecture inspirée du paradigme multi-tâche qui permet de réaliser de manière simultanée l'analyse syntaxique, l'analyse morphologique (comprenant la prédiction des parties-du-discours ainsi que des attributs morphologiques de chaque mot, comme le genre, le nombre…) et un étiquetage fonctionnel des mots.

2. Par exemple, le syntagme `(NP ce (Srel (NP-obj que) (NP-suj le FMI) (VN avait anticipé)))` (FTB) exhiberait une discontinuité si l'on supposait l'existence d'un syntagme verbal constitué de *que* et du VN.

3. Théoriquement, leur complexité est quadratique, mais en pratique, on observe des temps d'analyse linéaires.

3.1 Système de transitions

Le système de transitions SR-GAP pour le cas discontinu que nous décrivons ici suppose que tous les arbres du corpus sont binaires et lexicalisés, c'est-à-dire que pour chaque constituant, la forme lexicale de sa tête syntaxique est concaténée à sa catégorie. On présente en section 4.1 les prétraitements nécessaires pour que le corpus d'entraînement satisfasse à ces hypothèses. Ce système de transition est une extension de l'algorithme d'analyse par décalage et réduction (*shift-reduce*) suffisamment expressif pour dériver des arbres discontinus. L'analyseur utilise trois structures de données. Une file B contient les mots de la phrase qu'il reste à analyser. Une pile S et une file à double entrée D contiennent des arbres partiels. Les réductions vont porter sur les sommets respectifs de ces deux structures.

Par contraste avec l'algorithme *shift-reduce* où les réductions s'appliquent toujours aux deux éléments au sommet de la pile, SR-GAP utilise S et D pour partitionner la pile traditionnelle de l'automate d'analyse et permettre une réduction entre le sommet de la pile et un élément situé arbitrairement profondément dans la pile.

Une configuration d'analyse est un triplet $\langle S, D, B\rangle$ à un moment donné de l'analyse. Initialement, S et D sont vides et B contient la phrase à analyser sous la forme d'une séquence de tokens. L'algorithme procède de manière incrémentale en dérivant de nouvelles configurations en appliquant itérativement une action parmi l'ensemble des actions possibles. Dans les définitions suivantes, nous notons la concaténation $|$, les sommets de S, D, B respectivement s_0, d_0, b_0, et $X[h]$ un symbole non terminal X lexicalisé par le mot h.

$$\textsc{Shift}(\langle S, D, b_0|B\rangle) = \langle S|D, b_0, B\rangle$$
$$\textsc{ReduceLeft-X}(\langle S|s_0[h], D|d_0[h'], B\rangle) = \langle S|D, X[h], B\rangle$$
$$\textsc{ReduceRight-X}(\langle S|s_0[h], D|d_0[h'], B\rangle) = \langle S|D, X[h'], B\rangle$$
$$\textsc{ReduceUnary-X}(\langle S, d_0[h], B\rangle) = \langle S, X[h], B\rangle$$
$$\textsc{Gap}(\langle S|s_0, D, B\rangle) = \langle S, s_0|D, B\rangle$$

L'action SHIFT permet d'intégrer le token suivant à D. Les réductions droite et gauche construisent un nouveau constituant étiqueté X ayant pour descendants les sommets de S et D et lui assigne sa tête. Les réductions unaires permettent de gérer les réécritures unaires qui génèrent des pré-terminaux. Enfin, l'action GAP permet de rendre accessible un constituant de S pour une réduction en enfilant le sommet de S sur D.

L'analyse se termine quand S et B sont vides et quand D contient un seul élément : la racine de l'arbre prédit. En pratique, il est nécessaire de poser certaines préconditions sur chacune des actions pour assurer que les arbres produits sont bien formés (Coavoux & Crabbé, 2017a). On présente en table 3 un exemple de dérivation pour l'arbre de la figure 3 (partie droite).

3.2 Méthode de pondération

Pour une phrase donnée x_1^n, il existe un nombre exponentiel de séquences d'actions, et donc d'arbres, possibles. Pour pondérer chacune des dérivations, nous utilisons un classifieur. Le poids d'une

S	D	B	Action
		PROWH[Que] V[dois] CLS[-je] VINF[faire] PONCT[?]	SHIFT
	PROWH[Que]	V[dois] CLS[-je] VINF[faire] PONCT[?]	REDUCEUNARY-NP
	NP[Que]	V[dois] CLS[-je] VINF[faire] PONCT[?]	SHIFT
NP[Que]	V[dois]	CLS[-je] VINF[faire] PONCT[?]	SHIFT
NP[Que] V[dois]	CLS[-je]	VINF[faire] PONCT[?]	REDUCELEFT-VN
NP[Que]	VN[dois]	VINF[faire] PONCT[?]	SHIFT
NP[Que] VN[dois]	VINF[faire]	PONCT[?]	REDUCEUNARY-VN
NP[Que] VN[dois]	VN[faire]	PONCT[?]	GAP
NP[Que]	VN[dois] VN[faire]	PONCT[?]	REDUCERIGHT-VPINF
VN[dois]	VPinf[faire]	PONCT[?]	REDUCERIGHT-SENT:
	SENT:[dois]	PONCT[?]	SHIFT
SENT:[dois]	PONCT[?]		REDUCELEFT-SENT
	SENT[dois]		

TABLE 3 – Exemple d'exécution du système de transitions.

dérivation a_1^n se décompose par le produit des probabilités des actions $(a_1, \dots a_n)$ qui la constituent :

$$\text{score}(a_1^n) = \prod_{i=1}^{n} P(a_i | a_1^{i-1}, x_1^n) \tag{1}$$

Pour déterminer la séquence de score maximal, nous effectuons une recherche gloutonne, c'est-à-dire que nous exécutons à chaque étape d'analyse l'action de plus forte probabilité. Cette recherche est approximative, elle ne donne aucune garantie d'optimalité, mais elle permet d'obtenir de très bons résultats en pratique.

Le classifieur que nous utilisons pour déterminer les probabilités des actions est un réseau de neurones. Son architecture comprend deux modules connectés entre eux et entraînés simultanément. Le premier de ces modules est une cascade de codeurs LSTM bidirectionnels (Hochreiter & Schmidhuber, 1997) chargé d'une part de construire des représentations contextuelles pour chaque mot d'une phrase, et d'autre part de réaliser l'analyse morphologique et fonctionnelle des mots. Le second module est un réseau à propagation avant qui score les actions possibles étant donnée une configuration, en utilisant les représentations contextuelles des mots comme entrée.

Cette architecture n'est pas nouvelle en soi et a été proposée récemment pour l'analyse en constituants (Cross & Huang, 2016a) et l'analyse en dépendances (Kiperwasser & Goldberg, 2016). En revanche, à l'instar de Coavoux & Crabbé (2017b), nous y intégrons un mécanisme pour réaliser de manière simultanée l'analyse morphologique et l'étiquetage fonctionnel dans un cadre d'apprentissage multi-tâches.

3.2.1 Codeur LSTM bidirectionnel

Un codeur LSTM bidirectionnel (bi-LSTM) est un type de réseau de neurones récurrent qui permet de représenter une séquence par un vecteur de réels de taille fixe. À partir d'une séquence $x_1^n = (x_1, x_2, \dots, x_n)$, il procède en construisant itérativement des représentations $\text{LSTM}_f(x_1^i) \in \mathbb{R}^d$ pour chacun des préfixes $x_1^1, x_1^2, \dots$ de la séquence. Un second codeur LSTM, droite-gauche construit des représentations $\text{LSTM}_b(x_i^n) \in \mathbb{R}^d$ pour chacun des suffixes $x_{n-1}^n, x_{n-2}^n, \dots, x_1^n$. La représentation du token d'indice i est alors la concaténation $\mathbf{h}_i = [\text{LSTM}_f(x_1^i); \text{LSTM}_b(x_i^n)]$.

Dans notre cas, l'entrée du codeur consiste en la concaténation de deux vecteurs $[\mathbf{v}_i, \mathbf{c}_i]$ pour chacun des tokens x_i. $\mathbf{v}_i$ est un vecteur de mot standard (*word embedding*). $\mathbf{c}_i$ est une représentation obtenue à partir de la séquence des caractères du token, à l'aide d'un second bi-LSTM (Plank *et al.*, 2016).

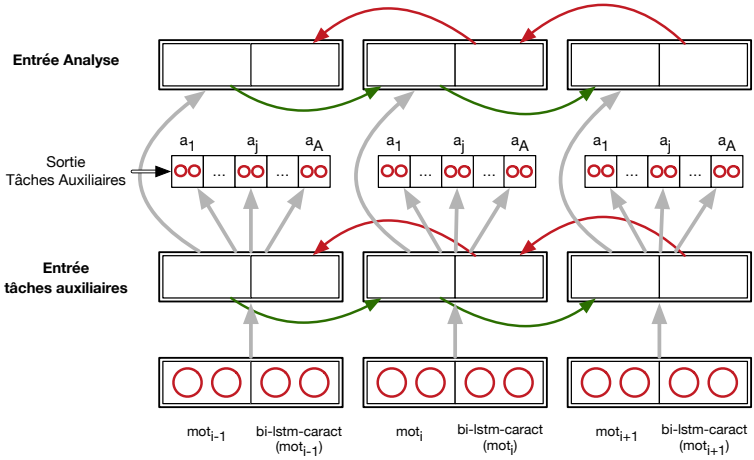


FIGURE 2 – Architecture du module bi-LSTM.

En analyse syntaxique par transitions, la motivation principale pour utiliser des bi-LSTMs est d'obtenir des représentations de taille fixe pour des structures de données qui peuvent être arbitrairement grandes. En particulier, une représentation adéquate de la file d'attente B permet de réduire les biais de localité des décisions inhérents à l'analyse incrémentale et aux méthodes de recherche approximatives.

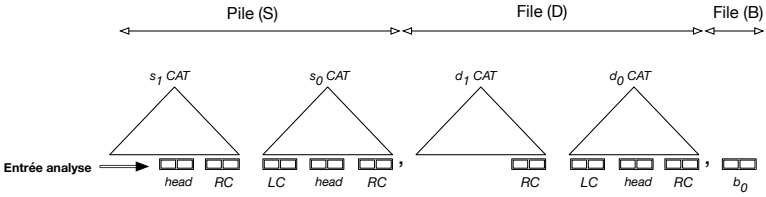
Analyse morphologique et étiquetage fonctionnel Le second rôle du module bi-LSTM est de réaliser l'analyse morphologique et l'étiquetage fonctionnel de chacun des tokens. L'analyse morphologique consiste en la prédiction de la partie-du-discours ainsi que d'un ensemble d'attributs pour un token donné. Les attributs morphologiques utilisés comprennent le mode, le temps, la personne, le genre, la sous-catégorie, ainsi que deux attributs indiquant si le token est la tête d'une expression polylexicale et s'il fait partie d'une expression polylexicale. Enfin l'étiquetage fonctionnel consiste à prédire la fonction syntaxique du token telle qu'elle serait annotée dans un corpus en dépendances.

Pour réaliser les prédictions, nous utilisons un classifieur par attribut à prédire. Par exemple, pour prédire l'attribut j du token i, nous calculons une distribution sur les classes possibles :

$$\text{Softmax}(W^{(j)} \cdot \mathbf{h}_i + \mathbf{b}^{(j)})$$

où $W^{(j)}$ et $\mathbf{b}^{(j)}$ sont des paramètres. Nous choisissons ensuite la classe de plus forte probabilité. Cette façon de procéder peut être vue comme comme une généralisation de l'architecture de tagging de Plank *et al.* (2016) au cas où l'on prédit un nombre arbitraire d'étiquettes pour chaque token.

L'architecture du module bi-LSTM est donnée en figure 2. Nous utilisons un bi-LSTM à deux couches : la première sert d'entrée pour les tâches d'étiquetage, et la seconde construit les représentations utilisées par le classifieur d'actions.



Traits de catégories syntaxiques	$s_0.CAT, s_1.CAT, d_0.CAT, d_1.CAT$
Traits lexicaux	$b_0, s_0.\{head, LC, RC\}, d_0.\{head, LC, RC\}, s_1.\{head, RC\}, d_1.\{RC\}$

TABLE 4 – Sous-structure du réseau à propagation avant utilisée pour prédire les actions. Les x_i adressent des positions dans une configuration. On utilise une notation pointée pour adresser la tête ($head$), le coin gauche (LC) et le coin droit (RC) d'un constituant. Les traits lexicaux sont instanciés par les représentations contextuelles (du bi-LSTM) correspondantes. Les traits syntaxiques sont instanciés par des vecteurs (*embedding*) de non-terminaux, appris lors de l'entraînement.

3.2.2 Classification des actions

Le second module du réseau est un réseau à propagation avant. Il contient deux couches cachées qui utilisent un rectifieur linéaire comme fonction d'activation (ReLU : $x \mapsto \max\{0, x\}$). Enfin, sa couche de sortie utilise la fonction Softmax pour calculer les probabilités $P(\cdot\,|a_1^{i-1}, x_1^n)$ de chacune des actions possibles (cf. équation 1). Ce réseau est semblable à ceux utilisés depuis Chen & Manning (2014) en analyse syntaxique. La différence principale est qu'au lieu d'utiliser des vecteurs de mots en entrée, il utilise les représentations contextuelles construites par le module bi-LSTM. On présente en table 4 l'ensemble des symboles qui forment son entrée.

4 Expériences

Nos expériences présentent quelques premiers résultats d'analyse en constituants discontinus pour le français avec la méthode décrite jusqu'ici. Elles contrastent des résultats obtenus essentiellement sur les corpus French Treebank, Sequoia et French Question Bank. Elles permettent de comparer le comportement de la méthode proposée sur des données qui présentent des proportions variables de discontinuités, le French Treebank comportant en proportion deux fois moins de discontinuités que le French Question Bank.

4.1 Protocole

Données et prétraitements Nous avons utilisé les versions discontinues, présentées plus haut, de trois corpus arborés du français. Nous avons effectué les prétraitements standards suivants. Les têtes de chacun des syntagmes ont été annotées à l'aide de règles adaptées de précédents travaux (Arun & Keller, 2005). Comme le système de transitions que nous utilisons ne permet de dériver que des arbres binaires, nous avons binarisé les arbres en appliquant une procédure standard appelée binarisation par la tête avec markovisation d'ordre 0 (Klein & Manning, 2003). Cette procédure consiste à binariser

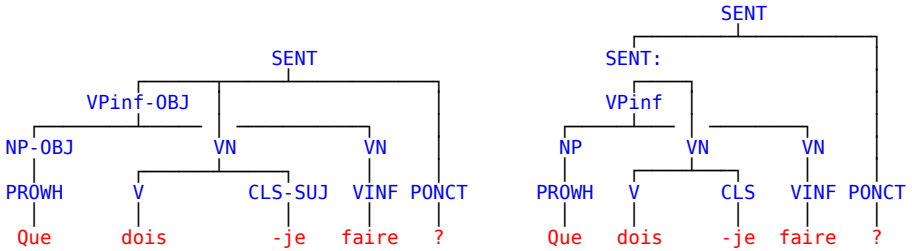


FIGURE 3 – Résultat du prétraitement appliqué à l'arbre en partie supérieure droite de la figure 1, reproduit ici à gauche. Le symbole SENT: est un symbole temporaire introduit par la binarisation.

un syntagme n-aire en partant de sa tête de sorte que chacun des symboles non-terminaux temporaires ajoutés lors de la binarisation contienne la tête du syntagme. Par ailleurs, nous avons fusionné les productions unaires [4]. Enfin, nous ignorons les étiquettes fonctionnelles annotées sur les constituants. On présente en figure 3 un arbre obtenu après pré-traitement.

Expériences Nous avons réalisé deux séries d'expériences. Pour la première, nous utilisons le FTB avec la division standard entre corpus d'entraînement, de développement et de test (SPMRL). Nous entraînons l'analyseur sur FTB-TRAIN et l'évaluons sur FTB-TEST ainsi que sur l'intégralité du Sequoia et du French Question Bank. La deuxième série d'expériences s'intéresse uniquement au French Question Bank. Nous utilisons les 500 dernières phrases de ce corpus pour l'évaluation, les 500 précédentes pour le développement et le reste pour l'entraînement.

Pour ces deux séries d'expériences, nous utilisons l'analyseur de Coavoux & Crabbé (2017a) pour servir de comparaison. Il s'agit d'un perceptron structuré avec recherche par faisceau, qui a obtenu des résultats état-de-l'art sur l'allemand. Comme il n'effectue pas d'étiquetage morphosyntaxique, nous utilisons pour le French Treebank les tags prédits fournis par les organisateurs de la campagne d'évaluation SPMRL, et pour les autres corpus les tags prédits par l'analyseur bi-LSTM.

Entraînement Pour une phrase w, la séquence d'actions de référence correspondante $a = (a_1, a_2, \ldots, a_n)$ et la matrice $m = (m_{i,j})$ d'étiquettes morphologiques et fonctionnelle de référence, nous utilisons la fonction de coût suivante :

$$\mathcal{L}(w, a, m; \boldsymbol{\theta}) = -\sum_{i=1}^{n} \log P(a^i | w, a_1^{i-1}; \boldsymbol{\theta}) - \sum_{i=1}^{n} \sum_{j=1}^{k} \log P(m_{i,j} | w; \boldsymbol{\theta})$$

où $\boldsymbol{\theta}$ est l'ensemble des paramètres du modèles (comprenant les vecteurs de mots et les vecteurs de caractères), $m_{i,j}$ est l'attribut j du token d'indice i, et k est le nombre d'étiquettes à prédire pour chaque mot. Cette fonction s'écrit comme la somme de deux termes $\mathcal{L}(w, a, m; \boldsymbol{\theta}) = \mathcal{L}_p(w, a; \boldsymbol{\theta}) + \mathcal{L}_t(w, m; \boldsymbol{\theta})$ Le premier terme $\mathcal{L}_p$ est la log vraisemblance de la séquence d'actions de référence, c'est-à-dire la fonction de coût du parseur. Le second terme $\mathcal{L}_t$ est la log vraisemblance des étiquettes de référence pour tous les tokens de la phrase.

4. à l'exception de celles qui produisent des pré-terminaux de manière à faciliter l'interface avec le module d'analyse morphologique.

Nous optimisons cette fonction de coût avec une descente de gradient stochastique moyennée (Polyak & Juditsky, 1992), en procédant en deux étapes. Pour une phrase donnée, nous calculons le gradient de $\mathcal{L}_t$ par rapport à $\boldsymbol{\theta}$, puis nous mettons à jour les paramètres. Ensuite, nous réalisons la même opération [5] avec $\mathcal{L}_p$.

Hyperparamètres Notre analyseur comprend plusieurs hyperparamètres. Pour chaque expérience, nous avons sélectionné le modèle qui maximisait la mesure F1 sur le corpus de développement, en explorant les combinaisons possibles parmi les valeurs suivantes :

Hyperparamètres d'optimisation		Hyperparamètres d'architecture	
Pas d'apprentissage	0.02	Dim. des vecteurs de mots et de non-terminaux	16
Constante de décroissement du pas d'apprentissage	$\{10^{-6}, 10^{-5}\}$	Dim. des états LSTM (pour chaque direction)	$\{128, 256\}$
Borne sur la valeur absolue du gradient (*hard gradient clipping*)	$\{5, 10\}$	Dim. des vecteurs de caractères	$\{8, 16\}$
Nombre d'itérations sur le corpus d'entraînement	$\{16, 20, 24, 25\}$	Dim. des états LSTM pour les caractères	16
		Dim. des couches cachées du classifieur d'actions	128

Pour obtenir des estimations pour les mots inconnus, suivant Kiperwasser & Goldberg (2016), nous remplaçons de manière stochastique un token par un pseudo-mot "INCONNU" avec une probabilité $p(w) = \frac{\alpha}{\#\{w\}+\alpha}$ où $\#\{w\}$ est le nombre d'occurrence de w dans le corpus d'entraînement. Suivant Cross & Huang (2016b), nous avons choisi $\alpha = 0.8375$.

Évaluation Pour l'évaluation de l'analyse en constituants, nous utilisons le logiciel DISCODOP (van Cranenburgh *et al.*, 2016) avec le paramétrage de la campagne d'évaluation SPMRL (Seddah *et al.*, 2013) [6]. Ce paramétrage prend en compte la ponctuation. Nous reportons une métrique standard (F1) calculée sur l'ensemble des constituants (All) et une métrique qui ne prend en compte que les constituants discontinus (Disc). Dans chacun des cas, nous reportons également la précision (P) et le rappel (R). Notons que la précision et le rappel sont très proches quand ils sont calculés sur l'ensemble des constituants, mais manifestent de très fortes divergences lorsqu'on ne prend en compte que les constituants discontinus.

Comme nous utilisons un modèle lexicalisé, les arbres en constituants prédits encodent implicitement des arbres en dépendances. Nous évaluons donc aussi notre système sur les arbres en dépendances étiquetés qu'il produit, à l'aide de l'évaluateur fourni pour la campagne d'évaluation SPMRL.

4.2 Résultats

Nous donnons les résultats pour les corpus de développement en table 5. Pour les métriques d'évaluation des constituants, nous reportons quelques statistiques sur l'ensemble des 64 modèles utilisés lors de la calibration des hyperparamètres. On remarque que les métriques calculées sur l'ensemble des constituants sont assez stables d'un modèle à l'autre. En revanche, les métriques calculées seulement sur les constituants discontinus manifestent une très forte variance, et ce sur les deux corpus. La rareté des constituants discontinus (table 2) dans les corpus explique ce résultat. Cela rend l'évaluation quantitative problématique.

Nous présentons les résultats pour les corpus de test en table 6. Nous observons tout d'abord que notre modèle entraîné sur le French Treebank obtient de bons résultats généraux en terme de F1, de

5. En apprentissage multi-tâches, il est d'usage de sélectionner aléatoirement une seule tâche à chaque étape d'optimisation. En pratique, nous traitons toutes les tâches d'étiquetage comme une seule, pour accélérer l'apprentissage.

6. `spmrl.prm` téléchargeable à l'url `http://pauillac.inria.fr/~seddah/evalb_spmrl2013.tar.gz`

	Constituants						Dépendances		Tagging
		All			Disc.		UAS	LAS	
	F1	P	R	F1	P	R			
FTB-DEV – Entraînement sur FTB-TRAIN									
Meilleur	82.33	82.27	82.39	17.39	36.36	11.43	88.85	84.30	97.63
Maximum	82.33	82.3	82.39	32.0	60.0	22.86			
Minimum	80.2	80.11	80.3	3.85	5.88	2.86			
Écart-type	0.428	0.431	0.433	6.931	12.8	4.853			
FQB-DEV – Entraînement sur FQB-TRAIN									
Meilleur	95.18	95.09	95.26	62.50	55.56	71.43	95.53	92.25	96.82
Maximum	95.18	95.23	95.26	75.0	71.43	85.71			
Minimum	93.75	93.73	93.76	40.0	30.77	57.14			
Écart-type	0.317	0.324	0.33	8.838	11.42	7.785			

TABLE 5 – Résultats sur les corpus de développement. Nous donnons le modèle qui maximise le score F1 (Meilleur), ainsi que, pour chaque métrique les valeurs minimales, maximales, et la variance calculées sur l'ensemble des 64 modèles testés (voir le paragraphe Hyperparamètres).

tagging et de LAS [7]. En revanche, le perceptron se comporte mieux sur un autre domaine et obtient en général de meilleurs résultats sur les constituants discontinus. De manière générale, les résultats obtenus sur les constituants discontinus sont plus faibles que ceux obtenus typiquement sur les corpus allemands (Coavoux & Crabbé, 2017a), ce qui peut s'expliquer par la rareté des discontinuités dans les données françaises.

Lorsqu'on entraîne les analyseurs sur le French Question Bank, où les discontinuités sont plus fréquentes en proportion, ils obtiennent de bien meilleurs résultats sur les constituants discontinus (plus proches de ceux obtenus par ailleurs sur l'allemand). Cela suggère qu'on peut espérer améliorer la prédiction des discontinuités en entraînant sur des données où elles sont plus fréquentes. Des méthodes issues de l'adaptation de domaine, comme le suréchantillonage des phrases contenant des structures discontinues lors de l'entraînement pourraient permettre d'améliorer notre système.

5 Conclusion

Cet article propose de nouvelles instanciations de trois corpus arborés du français où certains phénomènes syntaxiques responsables de dépendances à longue distance sont représentés directement à l'aide de structures discontinues. Ces données nous ont permis d'entraîner un analyseur syntaxique par transitions efficace, lexicalisé et qui réalise à la fois l'analyse syntaxique en constituants discontinus, l'analyse morphologique et l'étiquetage fonctionnel. Ce système permet également d'extraire immédiatement une représentation en dépendances typées non-projectives de phrases du français.

Nos expériences mettent en évidence que la difficulté principale de l'analyse des structures discontinues n'est pas uniquement liée à la complexité des calculs théoriquement ajoutée par les représenta-

7. Nous reportons en table 6 les deux résultats publiés d'analyse en dépendances non projectives qui semblent les plus comparables à nos conditions expérimentales.

	Constituants						Dépendances		Tagging
		All			Disc.		UAS	LAS	
	F1	P	R	F1	P	R			
Entraînement sur FTB-TRAIN									
Analyseur bi-LSTM, faisceau=1 (glouton)									
FTB-TEST[a]	82.04	81.93	82.14	14.46	15.38	13.64	88.24	83.40	97.66
FQB-ALL	85.14	83.79	86.53	11.43	54.55	6.38	89.08	79.39	93.89
SEQUOIA	77.46	77.75	77.16	26.53	46.43	18.57	85.18	77.94	94.79
Perceptron structuré, faisceau=16									
FTB-TEST[a]	79.42	79.27	79.57	19.05	31.58	13.64	-	-	97.35
FQB-ALL	85.93	84.67	87.22	15.53	88.89	8.51	-	-	93.89
SEQUOIA	76.74	76.89	76.60	26.83	91.67	15.71	-	-	94.79
FTB-TEST : Michalon *et al.* (2016)	-	-	-	-	-	-	86.6	83.3	
FQB-ALL : Seddah & Candito (2016)	-	-	-	-	-	-	87.70	76.48	
Entraînement sur FQB-TRAIN									
Analyseur bi-LSTM, faisceau=1 (glouton)									
FQB-TEST	92.96	92.67	93.25	60.00	56.25	64.29	94.23	90.48	96.05
Perceptron structuré, faisceau=16									
FQB-TEST	93.28	93.43	93.13	83.33	100	71.43	-	-	93.39

TABLE 6 – Résultats sur les corpus de test. [a]Notre modèle bi-LSTM traite environ 480 tokens par seconde contre 920 pour le perceptron (en utilisant 1 CPU).

tions discontinues mais aussi à leur rareté dans les données. Cette rareté rend l'apprentissage difficile et l'évaluation des structures discontinues plus difficiles dans le cas du français que dans le cas de l'allemand tel que décrit par Coavoux & Crabbé (2017a).

Remerciements

Les auteurs remercient Marie Candito et Djamé Seddah pour nous avoir fourni les treebanks, ainsi que 3 relecteurs anonymes pour leurs nombreuses remarques.

Références

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a Treebank for French. In *Treebanks : Building and Using Parsed Corpora*, p. 165–188. Springer.

ARUN A. & KELLER F. (2005). Lexicalization in crosslinguistic probabilistic parsing : The case of French. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, p. 306–313, Ann Arbor, Michigan : Association for Computational Linguistics.

BRANTS T. (1998). *The NeGra Export Format for Annotated Corpora*. Rapport interne 98, Universität des Saarlandes, Saarbrücken.

BRESNAN J. (2001). *Lexical-Functional Syntax*. Oxford : Blackwell Publishers.

CANDITO M. & SEDDAH D. (2012a). Effectively long-distance dependencies in French : annotation and parsing evaluation. In *TLT 11 - The 11th International Workshop on Treebanks and Linguistic Theories*, Lisbon, Portugal.

CANDITO M. & SEDDAH D. (2012b). Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 321–334, Grenoble, France : ATALA/AFCP.

CARUANA R. (1997). Multitask learning. *Mach. Learn.*, **28**(1), 41–75.

CHEN D. & MANNING C. D. (2014). A fast and accurate dependency parser using neural networks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

COAVOUX M. & CRABBÉ B. (2017a). Incremental discontinuous phrase structure parsing with the gap transition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 1259–1270, Valencia, Spain : Association for Computational Linguistics.

COAVOUX M. & CRABBÉ B. (2017b). Multilingual lexicalized constituency parsing with word-level auxiliary tasks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, p. 331–336, Valencia, Spain : Association for Computational Linguistics.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, **12**, 2493–2537.

CROSS J. & HUANG L. (2016a). Incremental parsing with minimal features using bi-directional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 32–37, Berlin, Germany : Association for Computational Linguistics.

CROSS J. & HUANG L. (2016b). Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1–11, Austin, Texas : Association for Computational Linguistics.

EVANG K. & KALLMEYER L. (2011). Plcfrs parsing of english discontinuous constituents. In *Proceedings of the 12th International Conference on Parsing Technologies*, IWPT '11, p. 104–116, Stroudsburg, PA, USA : Association for Computational Linguistics.

GAZDAR G. (1981). Unbounded dependencies and coordinate structure. *Linguistic Inquiry*, **12**, 155–184.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Comput.*, **9**(8), 1735–1780.

JOSHI A. (1985). How much context sensitivity is necessary for characterizing structural descriptions. In D. DOWTY, L. KARTTUNEN & A. ZWICKY, Eds., *Natural Language Processing, Theoretical Computational and Psychological Perspectives*. Cambridge University Press.

KALLMEYER L. (2010). *Parsing Beyond Context-Free Grammars*. Springer Publishing Company, Incorporated, 1st edition.

KALLMEYER L. & MAIER W. (2013). Data-driven parsing using probabilistic linear context-free rewriting systems. *Computational Linguistics*, **39**(1), 87–119.

KIPERWASSER E. & GOLDBERG Y. (2016). Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association of Computational Linguistics – Volume 4, Issue 1*, p. 313–327.

KLEIN D. & MANNING C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, p. 423–430, Stroudsburg, PA, USA : Association for Computational Linguistics.

LEVY R. (2005). *Probabilistic Models of Word Order and Syntactic Discontinuity*. PhD thesis, Stanford University.

MAIER W. (2015). Discontinuous incremental shift-reduce parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 1202–1212, Beijing, China : Association for Computational Linguistics.

MAIER W. & LICHTE T. (2011). Characterizing discontinuity in constituent treebanks. In *Formal Grammar. 14th International Conference, FG 2009. Bordeaux, France, July 25-26, 2009. Revised Selected Papers*, volume 5591 of *LNCS/LNAI*, p. 167–182, Berlin, Heidelberg, New York : Springer-Verlag.

MARCUS M. P., SANTORINI B. & MARCINKIEWICZ M. A. (1993). Building a large annotated corpus of english : The penn treebank. *Computational Linguistics*, **19**(2), 313–330.

MCCAWLEY J. D. (1982). Parentheticals and discontinuous constituent structure. *Linguistic Inquiry*, **13**(1), 91–106.

MICHALON O., RIBEYRE C., CANDITO M. & NASR A. (2016). Deeper syntax for better semantic parsing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, p. 409–420, Osaka, Japan : The COLING 2016 Organizing Committee.

PLANK B., SØGAARD A. & GOLDBERG Y. (2016). 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 (Volume 2 : Short Papers)*, p. 412–418, Berlin, Germany : Association for Computational Linguistics.

POLYAK B. T. & JUDITSKY A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, **30**(4), 838–855.

SEDDAH D. & CANDITO M. (2016). Hard time parsing questions : Building a questionbank for french. In N. C. C. CHAIR), K. CHOUKRI, T. DECLERCK, S. GOGGI, M. GROBELNIK, B. MAEGAARD, J. MARIANI, H. MAZO, A. MORENO, J. ODIJK & S. PIPERIDIS, Eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France : European Language Resources Association (ELRA).

SEDDAH D., TSARFATY R., KÜBLER S., CANDITO M., CHOI J. D., FARKAS R., FOSTER J., GOENAGA I., GOJENOLA GALLETEBEITIA K., GOLDBERG Y., GREEN S., HABASH N., KUHLMANN M., MAIER W., NIVRE J., PRZEPIÓRKOWSKI A., ROTH R., SEEKER W., VERSLEY Y., VINCZE V., WOLIŃSKI M., WRÓBLEWSKA A. & DE LA CLERGERIE E. V. (2013). Overview of the SPMRL 2013 shared task : A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, p. 146–182, Seattle, Washington, USA : Association for Computational Linguistics.

SKUT W., KRENN B., BRANTS T. & USZKOREIT H. (1997). An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*, Washington, DC.

VAN CRANENBURGH A., SCHA R. & BOD R. (2016). Data-oriented parsing with discontinuous constituents and function tags. *J. Language Modelling*, **4**(1), 57–111.

VERSLEY Y. (2014). Experiments with easy-first nonprojective constituent parsing. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, p. 39–53, Dublin, Ireland : Dublin City University.