Because Syntax Does Matter: Improving Predicate-Argument Structures Parsing with Syntactic Features

Corentin Ribeyre[⋆] [◦] Eric Villemonte de la Clergerie[⋆] Djamé Seddah[⋆] [⋄]
[⋆]Alpage, INRIA
[◦]Univ Paris Diderot, Sorbonne Paris Cité
[⋄] Université Paris Sorbonne
firstname.lastname@inria.fr

Abstract

Parsing full-fledged predicate-argument structures in a deep syntax framework requires graphs to be predicted. Using the DeepBank (Flickinger et al., 2012) and the Predicate-Argument Structure treebank (Miyao and Tsujii, 2005) as a test field, we show how transition-based parsers, extended to handle connected graphs, benefit from the use of topologically different syntactic features such as dependencies, tree fragments, spines or syntactic paths, bringing a much needed context to the parsing models, improving notably over long distance dependencies and elided coordinate structures. By confirming this positive impact on an accurate 2nd-order graph-based parser (Martins and Almeida, 2014), we establish a new state-of-the-art on these data sets.

1 Introduction

For the majority of the state-of-the-art parsers that routinely reach ninety percent performance plateau in capturing tree structures, the question of *what next* crucially arises. Indeed, it has long been thought that the bottleneck preventing the advent of accurate syntax-to-semantic interfaces lies in the quality of the preceding phase of analysis: the better the parse, the better the output. The truth is that most of the structures used to train current parsing models are degraded versions of a more informative data set: the Wall Street journal section of the Penn treebank (PTB, (Marcus et al., 1993)) which is often stripped of its richer set of annotations (*i.e.* traces and functional labels are removed), while, for reasons of efficiency and availability, projective dependency trees are often given preference over richer graph structures (Nivre and Nilsson, 2005; Sagae and Tsujii, 2008). This led to the emergence of *surface* syntax-based parsers (Charniak, 2000; Nivre, 2003; Petrov et al., 2006) whose output cannot by themselves be used to extract full-fledged predicate-argument structures. For example, control verb constructions, it-cleft structures, argument sharing in ellipsis coordination, etc. are among the phenomena requiring a graph to be properly accounted for. The dichotomy between what can usually be parsed with high accuracy and what lies in the deeper syntactic description has initiated a line of research devoted to closing the gap between surface syntax and richer structures. For most of the previous decade, the term *deep syntax* was used for rich parsing models built upon enriched versions of a constituency treebank, either with added HPSG or LFG annotation or CCG (almost) full rewrites (Miyao and Tsujii, 2005; Cahill et al., 2004; Hockenmaier, 2003). Its use now spreads by misnomer to models that provide more abstract structures, capable of generalizing classical functional labels to more semantic (in a logical view) arguments, potentially capable of neutralizing diathesis distinctions and of providing accurate predicate-argument structures. Although the building of syntax-to-semantic interface seems inextricably linked to an efficient parsing stage, inspirational works on semantic role labelling (Toutanova et al., 2005) and more recently on broad coverage semantic parsing (Du et al., 2014) that provide state-of-the-art results without relying on surface syntax, lead us to question the usefulness of syntactic parses for predicate-argument structure parsing.

In this study, we investigate the impact of syntactic features on a transition-based graph parser by testing on two treebanks. We take advantage of the recent release for the *SemEval 2014 shared task* on semantic dependency parsing, by Oepen et

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 64–74,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

al. (2014) of two semantic-based treebanks, derived from two HPSG resources, the DeepBank (DM, (Flickinger et al., 2012)) and the Enju's predicate argument structure (PAS, (Miyao and Tsujii, 2005)), to investigate the impact of syntactic features on a transition-based graph parser. Our results show that surface syntactic features significantly improve the parsing of predicate-argument structures. More specifically, we show that adding syntactic context improves the recognition of long distance dependencies and elliptical constructions. We finally discuss the usefulness of our approach, when applied on a second-order model based on dual decomposition (Martins and Almeida, 2014), showing that our use of syntactic features enhances this model accuracy and provides state-of-the-art performance.

2 Deep Syntax and Underspecified Semantic Corpora

DeepBank Corpus Semantic dependency graphs in the DM Corpus are the result of a two-step simplification of the underspecified logical-form meaning representations, based on Minimal Recursion Semantic (MRS, (Copestake et al., 1995; Copestake et al., 2005)), derived from the manually annotated DeepBank treebank (Flickinger et al., 2012). First, Oepen and Lønning (2006) define a conversion from original MRS formulae to variable-free Elementary Dependency Structures (EDS), which (a) maps each predication in the MRS logical-form meaning representation to a node in a dependency graph and (b) transforms argument relations represented by shared logical variables into directed dependency links between graph nodes. Then, in a second conversion step, the EDS graphs are further reduced into strict bi-lexical form, *i.e.* a set of directed, binary dependency relations holding exclusively between lexical units (Ivanova et al., 2012). Even though both conversion steps are, by design, lossy, DM semantic dependency graphs present a true subset of the information encoded in the full, original MRS data set.

Predicate-Argument Structure Corpus Enju Predicate-Argument Structures (PAS Corpus) are derived from the automatic HPSG-style annotation of the Penn Treebank (Miyao and Tsujii, 2004) that was primarily used for the development of the Enju parsing system (Miyao and Tsujii, 2005). The PAS data set is an extraction of predicate-argument structures from the Enju HPSG treebank and contains word-to-word semantic dependencies. Each dependency type is made of two elements: a coarse part-of-speech of the head predicate dependent (e.g. verb and adjective), and the argument (e.g. ARG1 and ARG2).

Although both are derived from HSPG resources (a hand-crafted grammar for DM, a treebank-based one for PAS), they differ in their core linguistic choices (functional heads vs lexical heads, coordination scheme, *etc.*) leading to different views of the predicate argument structure for the same sentence (Ivanova et al., 2012). Thus, even though both corpora may appear to contain a similar number of dependency labels, as shown in Table 1, their annotation schemes depict a deeply divergent linguistic reality exposed by two very different distributions. In DM, 9 labels account for almost 95% of all dependencies whereas a label set twice as large covers the same percentage for PAS, as shown in Table 2. Furthermore, semantically empty elements are widespread in the DeepBank (around 21.5%), compared to a low rate of 4.3% in PAS. In other words, the latter is somewhat more *dense* and consequently more syntactic. This is due to the fact that PAS integrates markers for infinitives, auxiliaries, and most punctuation marks into its graphs, whereas DM considers them as semantically void. DM corpus is clearly heading toward more semantic analysis while the PAS corpus aims at providing a more abstract deep syntax analysis than regular surface syntax trees. Both treebanks are used in their bilexical dependency formats.

	DM Corpus		PAS Corpus	
	Train	Dev	Train	Dev
# Sentences	32,389	1,614	32,389	1,614
# Tokens	742,736	36,810	742,736	36,810
% Void tokens	21.63	21.58	4.30	4.25
# Planar graphs	18,855	972	17,477	953
# Non planar	13,534	642	14,912	661
# Edges	559,975	27,779	723,445	35,573
% Crossing edges	4.24	4.05	5.69	4.46
Label set	52	36	43	40

Table 1: DM and PAS treebank properties

DM LABELS	%	PAS LABELS	%
ARG1	37.89	adj_ARG1	13.46
ARG2	23.08	noun_ARG1	9.54
compound	11.01	prep_ARG2	9.51
BV	10.39	prep_ARG1	9.37
root	5.77	verb_ARG2	9.34
poss	2.23	verb_ARG1	9.23
-and-c	2.02	det_ARG1	9.13
loc	1.38	punct_ARG1	5.23
ARG3	1.21	root	4.48
times	*0.87*	aux-ARG2	3.06
mwe	*0.85*	aux-ARG1	3.05
appos	*0.72*	coord-ARG2	2.35
conj	*0.57*	coord-ARG1	2.35
neg	*0.47*	comp-ARG1	1.85
subord	*0.43*	conj-ARG1	1.20
-or-c	*0.31*	poss-ARG2	0.89
-but-c	*0.20*	poss-ARG1	0.85
total	**94.98**	**total**	**94.89**

Table 2: Breakdown of Label Statistics.
Cell values in italics not counted in the DM *total.*

3 Transition-based Graphs Parsing

$$\begin{array}{lll}
(\sigma, w_i|\beta, A) & \vdash (\sigma|w_i, \beta, A) & \text{(shift)} \\
(\sigma|w_j|w_i, \beta, A) & \vdash (\sigma|w_i, \beta, A \cup (w_i, r, w_j)) & \text{(lR)} \\
(\sigma|w_j|w_i, \beta, A) & \vdash (\sigma|w_j, \beta, A \cup (w_j, r, w_i)) & \text{(rR)} \\
(\sigma|w_j|w_i, \beta, A) & \vdash (\sigma|w_j|w_i, \beta, A \cup (w_i, r, w_j)) & \text{(lA)} \\
(\sigma|w_j|w_i, \beta, A) & \vdash (\sigma|w_j, w_i|\beta, A \cup (w_j, r, w_i) & \text{(rA)} \\
(\sigma|w_i, \beta, A) & \vdash (\sigma, \beta, A) & \text{(pop0)}
\end{array}$$

Figure 1: Set of transitions for dependency graphs.

Shift-reduce transition-based parsers essentially rely on *configurations* formed of a stack and a buffer, with stack transitions used to move from a configuration to the next one, until reaching a final configuration. Following Kübler et al. (2009), we define a configuration by $c = (\sigma, \beta, \mathcal{A})$ where σ denotes a stack of words w_i, β a buffer of words, and $\mathcal{A}$ a set of dependency arcs of the form (w_i, r, w_j), with w_i the head, w_j the dependent, and r a label in some set R. As shown in Figure 1, besides the usual *shift* and *reduce* transitions (lR & rR) of the *arc-standard* strategy, we introduced the new left and right *attach* (lA & rA) transitions for adding new dependencies (while keeping the dependent on the stack) and a *pop0* transition to remove a word from the stack after attachment of its dependents. All the transitions that add an edge must also satisfy the condition that the newly created edge does not introduce a cycle or multiple edges between the same pair of nodes. It is to be noted that the *pop0* action may also be used to remove words with no heads.

Word$_{\sigma_1,\sigma_2,\sigma_3}$	Lemma$_{\sigma_1,\sigma_2,\sigma_3}$	POS$_{\sigma_1,\sigma_2,\sigma_3}$
Word$_{\beta_1,\beta_2}$	Lemma$_{\beta_1,\beta_2}$	POS$_{\beta_1,\beta_2,\beta_3}$
leftPOS$_{\sigma_1,\sigma_2}$	rightPOS$_{\sigma_1,\sigma_2}$	leftLabel$_{\sigma_1,\sigma_2}$
rightLabel$_{\sigma_1,\sigma_2}$	a	d_{12} d'_{11}

Table 3: Baseline features for the parser.
X$\sigma_i, \ldots, \sigma_j$ stands for X$\sigma_i, \ldots,$ Xσ_j.

We base our work on the the DAG parser of Sagae and Tsujii (2008) (henceforth S&T) which we extended with the set of actions displayed above (Figure 1) to cope with partially connected planar graphs, and we gave it the ability to take advantage of an extended set of features. Finally, for efficiency reasons (memory consumption and speed), we replaced the original Maxent model with an averaged structured perceptron (Freund and Schapire, 1999; Collins, 2002).

4 Feature Design

4.1 Baseline Features

We define Word$_{\beta_i}$ (resp. Lemma$_{\beta_i}$ and POS$_{\beta_i}$) as the word (resp. lemma and part-of-speech) at position i in the queue. The same goes for σ_i, which is the position i in the stack. Let $d_{i,j}$ be the distance between Word$_{\sigma_i}$ and Word$_{\sigma_j}$. We also define $d'_{i,j}$, the distance between Word$_{\beta_i}$ and Word$_{\sigma_j}$. In addition, we define leftPOS$_{\sigma_i}$ (resp. leftLabel$_{\sigma_i}$) the part-of-speech (resp. the label if any) of the word immediately to the left of σ_i, and the same goes for rightPOS$_{\sigma_i}$ (resp. rightLabel$_{\sigma_i}$). Finally, a is the previous action predicted by the parser. Table 3 lists our baseline features. X$\sigma_i, \sigma_j, \sigma_k$ means that we use Xσ_i, Xσ_j, Xσ_k as unigram features as well as bigram and trigram features.

4.2 Syntactic Features

We combined the previous features with different types of syntactic features (constituents and dependencies), our intuition being that syntax and semantic are interdependent, and that syntactic features should therefore help predicate-argument parsing. In fact, we considered that the low density of syntactic information (compared to regular dependency treebanks) would be counterbalanced by adding more context. We considered the following pieces of information in particular.

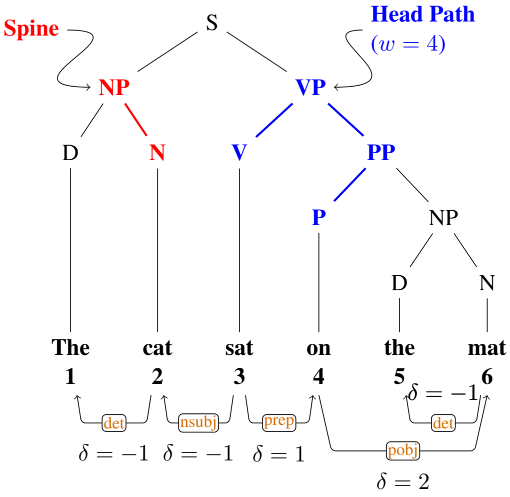


Figure 2: Schema of Syntactic Features

Constituent Tree Fragments These consist of fragments of syntactic trees predicted by the Petrov et al. (2006) parser in a 10-way jackknife setting. They can be used as enhanced POS or as features.

Spinal Elementary Trees A full set of parses was reconstructed from the tree fragments using a slightly tweaked version of the CoNLL 2009 shared task processing tools (Hajič et al., 2009). We then extracted a *spine grammar* (Seddah, 2010) using the head percolation table of the Bikel (2002) parser, slightly modified to avoid certain determiners being marked as heads in certain configurations. The resulting spines were assigned in a deterministic way (red part in Figure 2).

Predicted MATE Dependency Labels These consist of the dependency labels predicted by the MATE parser (Bohnet, 2010), trained on a Stanford surface dependency version of the Penn Treebank. We combined the labels with a distance $\delta = t - h$ where t is the token position and h the head position (brown labels and δ in Figure 2). In addition, we expanded these features with the part-of-speech of the head of a given token (HPOS). The idea is to evaluate the informativeness of more abstract syntactic features since a <LABEL,HPOS> pair can be seen as generalizing many constituent subtrees.

Constituent Head Paths. Inspired by Björkelund et al. (2013), we used MATE dependencies to extract the shortest path between a token and its lexical head and included the path length w (in terms of traversed nodes) as a feature (blue part in Figure 2). The global idea is to use the phrase-based features to provide different kinds of syntactic context and the dependency-based features to provide generalisations over the functional label governing a token. The spines are seen as deterministic supertags, bringing a vertical context.

We report, in Table 4, the counts for each syntactic feature on each set.

	TREE FRAG.	MATE LABELS+δ	SPINES TREES	HEAD PATHS
TRAIN	648	1305	637	27,670
DEV	272	742	265	3,320
TEST	273	731	268	2,389

Table 4: Syntactic features statistics (Counts).

5 Experiments

Experimental Setup Both DM and PAS treebanks consist of texts from the PTB and which were either automatically derived from the original annotations or annotated with a hand-crafted grammar (see above). We use them in their bi-lexical dependency format, aligned at the token level as provided by Oepen et al. (2014)[1]. The following split is used: sections 00-19 for training, 20 for the dev. set and 21 for test[2]. All predicted parses are evaluated against the gold standard with *labeled precision, recall and f-measure* metrics.

Results Our experiments are based on the evaluation of the combinations of the 4 main types of syntactic features described in section 4: tree fragments (BKY), predicted mate dependencies (BN) and their extension with POS heads (BN(HPOS)), spinal elementary trees (SPINES) and head paths (PATHS).

The results are shown in Tables 5 and 6. All improvements from the baseline are significant with a p-value $p < 0.05$. There was no significant difference of the same p value between our two best mod-

[1] This alignment entailed the removal of all unparsed sentences.

[2] We used the same unusual split as in (Oepen et al., 2014) to be able to conduct meaningful comparisons with others.

els for each of the treebanks. [3]

As expected from the rapid overview of our datasets exposed earlier in section 2, the use of each single feature alone increases the performance over the baseline by 0.5 points for the BN feature in DM to 1.44 for PATHS, and by 1.10 for the SPINES to 1.85 for the PATHS features in PAS. Looking at the conjunction of two classes in the DM table, it seems that dependency-based features benefit from the extra context brought by constituents features, reaching an increase of 2.21 points for BKY+BN(HPOS). Interestingly, the maximum gain is brought by the addition of topologically different phrase-based features such as SPINES (+2.80, inherently vertical) or BKY (+2.76, often wider) to the previous best. Regarding PAS, similar trends can be observed, although the gains are more distributed. As opposed to DM where the conjunction of more features led to inferior results, here using a four-features class provides the second best improvement (ALL(HPOS) = BKY+BN(HPOS)+SPINES+PATHS), +2.82) while removing the SPINES slightly increases the score (+2.92). In fact, adding too many features to the model slightly degrades our scores, at least with regard to DM which has a larger label set than PAS.

Results show that syntactic information improves our parser performances. As each feature represents one unique piece of information, they benefit from being combined in order to provide more structural information.

6 Results Analysis

Following Mcdonald and Nivre (2007), we conducted an error analysis based on the two best models and the baseline for each corpus. As shown in section 5, syntactic features greatly improve semantic parsing. However, it is interesting to explore more precisely what kind of syntactic information boosts or penalizes our predictions. We consider, among other factors, the impact in terms of distance between the head and the dependent (edge length) and the labels. We also explore several linguistic phenomena well known to be difficult to recover.

DM Corpus (dev. set)	LP	LR	LF	
BASELINE	83.66	80.33	81.97	
BN	84.12	80.91	82.48	+0.51
BKY	85.10	81.70	83.36	+1.39
SPINES	84.72	81.31	82.98	+1.01
PATHS	85.15	81.74	83.41	+1.44
BN(HPOS)	**85.63**	**82.19**	**83.88**	**+1.91**
BKY+SPINES	85.41	81.88	83.61	+1.64
SPINES+PATHS	85.49	82.01	83.71	+1.74
BKY+BN	85.47	82.08	83.74	+1.77
BKY+PATHS	85.70	82.22	83.92	+1.95
BN(HPOS)+SPINES	85.94	82.48	84.17	+2.20
BKY+BN(HPOS)	85.96	82.46	84.18	+2.21
BN(HPOS)+PATHS	85.97	82.59	84.25	+2.28
BN+SPINES	86.05	82.55	84.26	+2.29
BN+PATHS	**86.05**	**82.64**	**84.31**	**+2.34**
BKY+SPINES+PATHS	85.64	82.23	83.90	+1.93
BKY+BN+SPINES	85.88	82.50	84.16	+2.19
BKY+BN(HPOS)+SPINES	86.38	82.81	84.56	+2.59
BN(HPOS)+SPINES+PATHS	86.28	82.91	84.56	+2.59
BKY+BN(HPOS)+PATHS	86.49	82.94	84.68	+2.71
BKY+BN+PATHS	**86.55**	**82.98**	**84.73**	**+2.76**
BN+SPINES+PATHS	**86.59**	**83.02**	**84.77**	**+2.80**
ALL	85.73	82.27	83.96	+1.99
ALL(HPOS)	**86.13**	**82.64**	**84.35**	**+2.38**

Table 5: Best results and gains on DM corpus.

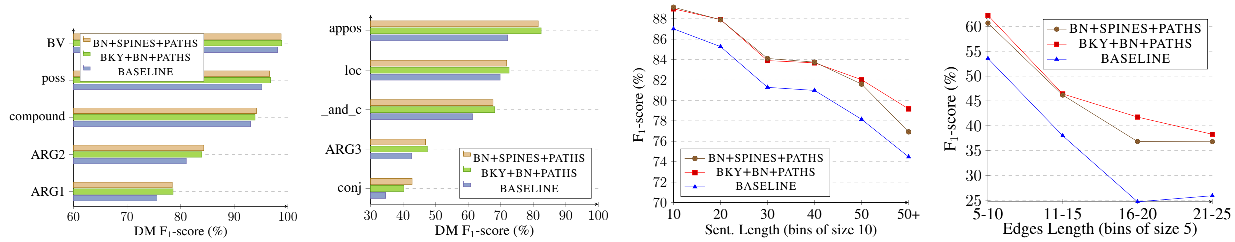
PAS Corpus (dev. set)	LP	LR	LF	
BASELINE	86.95	83.45	85.17	
SPINES	88.15	84.47	86.27	+1.10
BN	88.21	84.77	86.46	+1.29
BN(HPOS)	88.55	85.00	86.74	+1.57
BKY	88.63	84.97	86.76	+1.59
PATHS	**88.85**	**85.24**	**87.01**	**+1.84**
BKY+SPINES	88.84	85.20	86.98	+1.81
SPINES+PATHS	89.04	85.45	87.21	+2.04
BN(HPOS)+SPINES	89.18	85.49	87.30	+2.13
BN(HPOS)+PATHS	89.17	85.62	87.36	+2.19
BN+PATHS	89.32	85.74	87.49	+2.32
BKY+PATHS	89.44	85.72	87.54	+2.37
BKY+BN	89.30	85.87	87.55	+2.38
BN+SPINES	89.48	85.81	87.60	+2.43
BKY+BN(HPOS)	**89.49**	**85.80**	**87.61**	**+2.44**
BKY+SPINES+PATHS	89.35	85.54	87.40	+2.23
BKY+BN+SPINES	89.56	86.02	87.75	+2.58
BN(HPOS)+SPINES+PATHS	89.76	86.15	87.92	+2.75
BN+SPINES+PATHS	89.88	86.13	87.96	+2.79
BKY+BN+PATHS	89.82	86.20	87.97	+2.80
BKY+BN(HPOS)+PATHS	**89.93**	**86.32**	**88.09**	**+2.92**
ALL	89.70	86.11	87.87	+2.70
ALL(HPOS)	**89.91**	**86.14**	**87.99**	**+2.82**

Table 6: Best results and gains on PAS.

6.1 Breakdown by Labels

In Figures 3(a) and 4(a), we detail the scores for the five most frequent labels.

[3] We tested the statistical significance between our best models and the baseline with the paired bootstrap test (Berg-Kirkpatrick et al., 2012).



(a) Most frequent labels.

(b) Best improved labels.

(c) Sentence length.

(d) Long-distance dependencies.

Figure 3: Error analysis on DM (dev. set).

As observed in the charts, the scores are higher for the most frequent labels on both corpora, especially when dealing with verbal arguments. There are also two interesting cases for DM: the predictions of *_and_c* and *ARG3* edges show an improvement by at least 5 points (Figures 3(b) & 4(b)), showing that the recovery of coordination structures and the disambiguation of less frequent or more distant arguments is achieved by adding non-local features.

6.2 Length Factor

Longer sentences are notoriously difficult to parse for most parsing models. Figures 3(c) and 4(c) show the F_1-measure of our models with respect to sentence length (in bins of size 10: 1-10, 11-20, etc.) for the DM and PAS corpora.

It is worth noting that we greatly improve the scores for longer sentences. The use of paths and of the output of a graph-based parser (Bohnet, 2010) favors the capture of complex dependencies and enhances the learning of these constructions for our local transition-based parser. However, we also observe that the features are not able to completely stop the loss of F_1-score for longer sentences. The slopes of the curves in the different charts show the same trend: the longer the sentence, the lower the score.

6.3 Linguistic Factors

We now center our analysis on long-distance dependencies (LDDs), by focusing our attention on edges length, *i.e.* the distance between two words linked by an edge. We will then concentrate on subject ellipsis, in a treatment of LDDs more similar to the linguistic definition of Cahill et al. (2004).

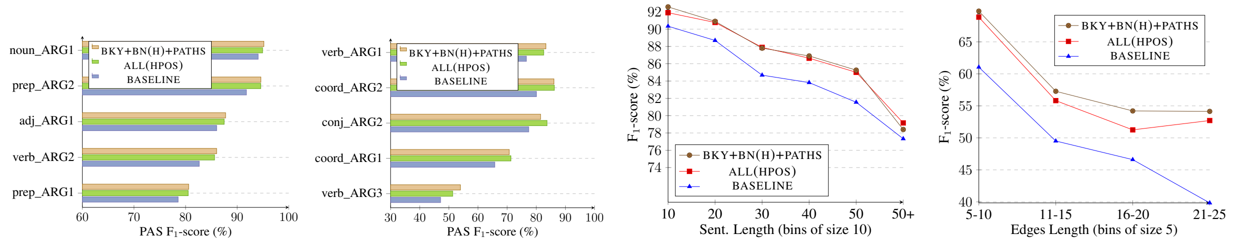
Long-distance Dependencies (LDDs) For many systems, LDDs are difficult to recover because they are generally under-represented in the training corpus and the constructions involved in LDDs often require deep linguistic knowledge to be recovered. In Figure 7, we report the distribution of long-distance dependencies by bins of size 5 up to 40. They only account for 15% of all the dependencies in both corpora. The longest dependencies consist of the first and second arguments of the verb as well as coordination links. In the case of elided coordination structures, we have long-distance dependencies when two coordinated verbs share the same first or second argument, which explains the distribution of lengths.

BINS	5-10	11-15	16-20	21-25	26-40
DM	2907	734	329	141	92
PAS	3705	1007	408	175	127

Table 7: Number of LDDs edges (dev. set).

As outlined in Figures 3(d) and 4(d), we can see that without structural information such as spines, surfacic dependencies or paths, the longest dependencies have low F_1-scores. When using these features, our models tend to perform better, with a gain of up to 25 points for high-dependency lengths (bins between 16-20 and 21-25).

In Table 8, we show the global improvement when considering edge lengths between 5 and 40. For both corpora, the improvement is the same (around 9 points), showing that structural information is the key to better predictions. Looking into this improvement more closely, we found that PATHS combined with BN tend to be crucial, whereas SPINES



(a) Most frequent labels.

(b) Best improved labels.

(c) Sentence length.

(d) Long-distance dependencies.

Figure 4: Error analysis on PAS (dev. set).
BKY+BN(H)+PATHS stands for BKY+BN(HPOS)+PATHS.

may sometimes penalize the models. Even though, BN+SPINES+PATHS is the best model for DM, a spine is only a *partial projection* which lacks attachment information. Spines alone only therefore provide a local context and are unable to cope well with LDDs.

Coordination Structures We now focus on structures with subject ellipsis. We extracted them by using a simple graph pattern, *i.e.* two verbs with a shared *ARG1* and a coordination dependency.

Our best models' scores are displayed in Tables 9. Once again, our models improve the F_1 score, but not in the same proportion. DM considers the conjunction as a semantically empty word and attaches an edge *_and_c* between the two verbs to mark the coordination. Consequently this edge is more difficult to predict, because it is less informative, our baseline model relying on tokens, lemmas and POS.

We note that the difference in the number of evaluated dependencies in both corpora comes from an annotation scheme divergence between PAS and DM regarding subject ellipsis. DM opts for coordinate structures with a chain of dependencies rooted at the first conjunct, the coordinating conjunctions being therefore semantically empty. In PAS, the final coordinating conjunction and each coordinating conjunction is a two-place predicate, taking left and right conjuncts as its arguments.

The gain of 6.30 points for DM (Table 9(a), resp. +3 for PAS) indicates that, when an annotation scheme is designed to have many semantically empty words, using syntactic information tends to enhance the parser accuracy. This gives a clear insight into what type of information is required to parse semantic graphs: the greater the distance between the head and the dependent, the larger the context needed to disambiguate the attachments.

6.4 Ruling out the Structural Factor Bias

It may argued that the improvement we noticed could stem from a potentially strong overlap between surface trees and predicate-argument structures, both in terms of edges and labels. In fact, the conversion from surfacic parses into predicate-argument structures requires a large amount of edges relabeling (for instance, when *nsubj* is relabeled to *ARG1*). We tested this hypothesis by computing the number of common edges between MATE predictions and DM and PAS. The overlap corresponds to about 22% of all edges in PAS and 27% in DM. Although important, it does not represent the majority of dependencies in our corpora, because most of edges are not present in surface predictions. We evaluated the improvement of the overlap as well as for the rest. Results show that our best models perform roughly the same on both sets. Interestingly, as opposed to PAS's model, DM's model performs better on the non-overlap part. This suggests that the use of PTB-based features is somehow not optimal when applied on a none PTB-based treebank, such as DM which comes from a handcrafted grammar.

	PAS	DM
Overlap	+2.87	+2.67
Rest	+2.70	+2.74

7 Discussion

Our point was to prove that providing more syntactic context, in the form of phrased-based tree fragments and surface dependencies, helps transition-

	LP	LR	LF	
BASELINE	54.95	42.53	47.95	
BN+SPINES+PATHS	64.23	50.55	56.57	+8.62
BKY+BN+PATHS	64.88	50.90	57.05	**+9.10**

(a) DM Corpus (dev. set).

	LP	LR	LF	
BASELINE	66.62	50.17	57.23	
ALL(HPOS)	74.03	57.58	64.78	+7.55
BKY+BN(HPOS)+PATHS	74.62	58.95	65.86	**+8.73**

(b) PAS Corpus (dev. set).

Table 8: Long-distance dependencies eval. (dev sets).

	LP	LR	LF	
BASELINE	90.00	48.57	63.09	
BN+SPINES+PATHS	96.02	53.65	68.84	+5.85
BKY+BN+PATHS	96.07	54.29	69.37	**+6.28**

(a) on DM (dev. set, 315 dependencies).

	LP	LR	LF	
BASELINE	97.51	61.48	75.41	
ALL(HPOS)	97.86	64.78	77.96	+2.55
BKY+BN(HPOS)+PATHS	98.57	65.09	78.41	**+3.00**

(b) on PAS (dev. set, 636 dependencies).

Table 9: Shared subjects coordinations eval. (dev sets).

based parsers to predict predicate-argument structures, especially for LDDs. Yet, compared to state-of-the-art systems, our results built on the S&T parser score lower than the top performers (Table 10).

However, we are currently extending a more advanced lattice-aware transition-based parser (DSR) with beams (Villemonte De La Clergerie, 2013) that takes advantage of cutting-edge techniques (dynamic programming, averaged perceptron with early updates, *etc.* following (Goldberg et al., 2013; Huang et al., 2012)) [4], which proves effective by reaching the state-of-the-art on PAS, outperforming Thomson et al. (2014) and second to the model of Martins and Almeida (2014). [5]

The point here is that using the same syntactic features as our base system exhibits the same improvement over a now much stronger baseline. We can conjecture that the ambiguities added by the relative scarcity of the deep annotations is efficiently handled by a more complete exploration of the search space, made possible by beam optimization.

We can also wonder whether the lower improvement brought to DM parsing by the PTB-based syntactic features does not come from the fact that the DM corpus and the PTB have divergent annotation schemes. In that aspect, PTB syntactic features may add some noise to the learning process, because they give more weight to conflicting decisions that led to correct structures in one but not in the other scheme.

By using features which, to a certain extent, (i) extend the domain of locality available at a given node and (ii) generalize some structural and functional contexts otherwise unavailable, we tried to overcome the main issue of transition-based parsers: they remain local in the sense that they lack a global view of the whole sentence.

Impact Beyond Transition-based Parser Of course, it can be argued that improving over a somewhat weak baseline is of limited interest. Our point was to investigate how the direct parsing of relatively sparse graph structures would benefit from the inclusion of more context via the use of topologically different syntactic pieces of information. However in that work, we mostly focused on transition based-parsing, which raises the question of the impact of our feature-set on a much more powerful and state-of-the-art model such as the TurboSemanticParser developed by Martins and Almeida (2014).

To this end, we extended the T.Parser so that it could cope with our syntactic features and studied the interaction of our best feature set with second order features (*i.e.* grand-parents and co-parents). Results in Table 11 show that the gain brought by adding syntactic features (+2.14 on DM over the baseline) is higher than the sole use of second order ones (+1.09). Furthermore, the gain brought by

[4] It uses a different set of transitions, notably *pop* actions instead of *left* and *right reduce*, and a *swap* that allow limited amount of non-planarity. Such a set raises issues with beams (several paths leading to a same item, final items reached with paths of various lengths, . . .), overcome by adding a 'noop' action only applied on final items to balance path lengths.

[5] Leaving aside the multiple (19) ensemble models of Du et al. (2014), because of the impracticability of the approach.

	PAS	DM
(T.Parser+features, this paper)	92.11	89.70
(Du et al., 2014)	92.04	89.40
(Martins and Almeida, 2014)	91.76	89.16
(DSR, this paper)	90.13	85.66
(Thomson et al., 2014)	89.63	83.97
(S&T, this paper)	87.5	83.84
(DSR, this paper, no feat)	87.02	83.91
(S&T, this paper, no feat)	84.18	81.17

Table 10: Comparison with the State-of-the-Art.

the second-order features is reduced by half when used jointly with our feature set (+1.09 vs +0.57 with them). However, although we could assess that the need of second order models is thus alleviated, the conjunction of both types of features still improves the parser performance by an overall gain of 1.62 points on DM (1.18 on PAS), suggesting that both feature sets contribute to different types of "structures". In short, the use of syntactic features is also relevant with a strong baseline, as they provide a global view to graph-based models, establishing a new state-of-the-art on these corpora.

	-Synt. Feat.	+Synt. Feat.	δ
DM, baseline	86.99	89.13	*+2.14*
+grandparent	87.66	89.43	*+1.77*
+co-parents	88.08	89.7	**+1.62**
PAS, baseline	89.73	91.68	*+1.95*
+grandparent	90.15	91.92	*+1.77*
+co-parents	90.93	92.11	**+1.18**

Table 11: LF Results for T.Parser (test set).
Baseline = arc-factored + siblings

Related Work A growing interest for semantic parsing has emerged over the past few years, with the availability of resources such as PropBank and NomBank (Palmer et al., 2005; Meyers et al., 2004) built on top of the Penn Treebank. The shallow semantic annotations they provide were among the targets of successful shared tasks on semantic role labeling (Surdeanu et al., 2008; Carreras and Màrquez, 2005). Actually, the conjoint use of such annotations with surface syntax dependencies bears some resemblance with predicate-argument structure parsing like we presented here. However, they diverge in that Propbank/Nombank annotations do not form connected graphs by themselves, as they only cover argument identification and nominal predicates. The range of phenomena they describe is also limited, compared to a full predicate-argument analysis as provided by DM and PAS (Oepen et al., 2014). More importantly, as pointed out by Yi et al. (2007), being verb-specific, Propbank's roles do not generalize well beyond the Arg0 argument (i.e. the subject/agent role) leading to inconsistencies.
However, the advent of such semantic-based resources have ignited a fruitful line of research, of which the use of heterogeneous sources of information to boost parsing performance has been investigated over the past decade (Chen and Rambow, 2003; Tsuruoka et al., 2004) with a strong regain of interest raised by the work of Moschitti et al. (2008), Henderson et al. (2008), Sagae (2009).

8 Conclusion

We described the use and combination of several kinds of syntactic features to improve predicate-argument parsing. To do so, we tested our approach of injecting surface-syntax features by thoroughly evaluating their impact on one transition-based graph parser, then validating on two more efficient parsers, over two deep syntax and semantic treebanks. Results of the syntax-enhanced *semantic* parsers exhibit a constant improvement, regardless of the annotation scheme and the parser used.
The question is now to establish whether will this be verified in other semantic data sets? From the parsing of deep syntax treebanks *a la* Meaning Text Theory (Ballesteros et al., 2014), to Framenet semantic parsing (Das et al., 2014) or data-driven approaches closer to ours (Flanigan et al., 2014), it is difficult to know which models will predominate from this bubbling field and what kind of semantic data sets will benefit the most from syntax.

Acknowledgements

We would like to thank Kenji Sagae and André F. T. Martins for making their parsers available and for kindly answering our questions. We also thank our anonymous reviewers for their comments. This work was partly funded by the Program "Investissements d'avenir" managed by Agence Nationale de la Recherche ANR-10-LABX-0083 (Labex EFL).

References

Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. 2014. Deep-syntactic parsing. In *In Proc. of COLING*, Dublin, Ireland.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005.

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proc. of the second international conference on Human Language Technology Research*, pages 178–182. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proc. of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, October.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proc. of the 23rd International Conference on Computational Linguistics*, pages 89–97.

Aoife Cahill, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proc. of ACL*, pages 320–327.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proc. of the Ninth Conference on Computational Natural Language Learning*, pages 152–164.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL)*, Seattle.

John Chen and Owen Rambow. 2003. Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proc. of the 2003 conference on Empirical methods in natural language processing*, pages 41–48. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, pages 1–8.

Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. Translation using minimal recursion semantics. In *Proc. of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 15–32. Citeseer.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Yantao Du, Fan Zhang, Weiwei Sun, and Xiaojun Wan. 2014. Peking: Profiling syntactic tree parsing techniques for semantic graph parsing. In *Proc. of the 8th International Workshop on Semantic Evaluation*, pages 459–464.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *in Proc. of ACL*, Baltimore, US.

Daniel Flickinger, Yi Zhang, and Valia Kordoni. 2012. DeepBank: a dynamically annotated treebank of the wall street journal. In *Proc. of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296.

Yoav Goldberg, Kai Zhao, and Liang Huang. 2013. Efficient implementation of beam-search incremental parsers. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sophia, Bulgaria.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proc. of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18.

James Henderson, Paola Merlo, Gabriele Musillo, and Ivan Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proc. of the Twelfth Conference on Computational Natural Language Learning*, pages 178–182.

Julia Hockenmaier. 2003. *Data and models for statistical parsing with Combinatory Categorial Grammar*. Ph.D. thesis.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proc. of HLT-NAACL 2012*, pages 142–151.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In *Proc. of the sixth linguistic annotation workshop*, pages 2–11.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Morgan and Claypool Publishers.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

T. André F. Martins and C. Mariana S. Almeida. 2014. Priberam: A turbo semantic parser with second order features. In *Proc. of the 8th International Workshop on Semantic Evaluation*, pages 471–476.

Ryan Mcdonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for nombank. In *LREC*, volume 4, pages 803–806.

Yusuke Miyao and Jun'ichi Tsujii. 2004. Deep Linguistic Analysis for the Accurate Identification of Predicate-Argument Relations. In *Proc. of the 18th International Conference on Computational Linguistics*, pages 1392–1397, Geneva, Switzerland.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proc. of ACL 2005*, pages 83–90.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2):193–224.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proc. of the 8th International Workshop on Parsing Technologies (IWPT*. Citeseer.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based mrs banking. In *Proc. of the 5th international conference on language resources and evaluation (lrec 2006)*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proc. of the 8th International Workshop on Semantic Evaluation*, pages 63–72.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Kenji Sagae and Jun'ichi Tsujii. 2008. Shift-reduce dependency DAG parsing. In *Proc. of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 753–760.

Kenji Sagae. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proc. of the 11th International Conference on Parsing Technologies*, pages 81–84. Association for Computational Linguistics.

Djamé Seddah. 2010. Exploring the spinal-stig model for parsing french. In *Proc. of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177. Association for Computational Linguistics.

Sam Thomson, Brendan O'Connor, Jeffrey Flanigan, David Bamman, Jesse Dodge, Swabha Swayamdipta, Nathan Schneider, Chris Dyer, and A. Noah Smith. 2014. CMU: Arc-Factored, Discriminative Semantic Dependency Parsing. In *Proc. of the 8th International Workshop on Semantic Evaluation*, pages 176–180.

Kristina Toutanova, Aria Haghighi, and Christopher D Manning. 2005. Joint learning improves semantic role labeling. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 589–596.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2004. Towards efficient probabilistic hpsg parsing: integrating semantic and syntactic preference to guide the parsing. In *Proc. of the IJCNLP-04 Workshop on Beyond Shallow Analyses*. Citeseer.

Éric Villemonte De La Clergerie. 2013. Exploring beam-based shift-reduce dependency parsing with DyALog: Results from the SPMRL 2013 shared task. In *4th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL'2013)*.

Szu-Ting Yi, Edward Loper, and Martha Palmer. 2007. Can semantic roles generalize across genres? In *HLT-NAACL*, pages 548–555.