Semi-Automatic Deep Syntactic Annotations of the French Treebank

Corentin Ribeyre, Marie Candito, Djamé Seddah

Semi-Automatic Deep Syntactic Annotations of the French Treebank

Corentin Ribeyre° Marie Candito° Djamé Seddah◇
°Univ. Paris Diderot, Sorbonne Paris Cité, Alpage, INRIA
◇ Université Paris Sorbonne, Alpage, INRIA
firstname.lastname@inria.fr

Abstract

We describe and evaluate the semi-automatic addition of a deep syntactic layer to the French Treebank (Abeillé and Barrier [1]), using an existing scheme (Candito et al. [6]). While some rare or highly ambiguous deep phenomena are handled manually, the remainings are derived using a graph-rewriting system (Ribeyre et al. [22]). Although not manually corrected, we think the resulting Deep Representations can pave the way for the emergence of deep syntactic parsers for French.

Introduction

Syntactic parsing has been the focus of intensive international research over the last decades, leading to current state-of-the-art parsers to provide quite high performance (on well-formed English text at least). However, extracting more semantically-oriented information from syntactic parses, be them of high quality, is as not straightforward, given the abundant syntax-semantic divergences and the idiosyncratic nature of syntax itself. "Deep syntax" is generally intended as an intermediate representation between what is actually observable (surface syntax) and a semantic representation, which abstracts away from *syntactic variation*, such as diathesis alternations or non-canonical word order, and which can thus serve as an easier basis for semantic analysis. Such view forms, for example, the basis of the Meaning-Text Theory, MTT, (Melčuk [16]).

Several initiatives have been proposed to obtain "deep" syntactic treebanks, with various meanings attached to the term "deep". For instance for Spanish, the AnCORA-UPF multi-layer corpus (Mille et al. [17]) includes a deep syntactic layer, inspired by the MTT. For English, the Penn Treebank (PTB, (Marcus et al. [15])) contains a certain amount of "deep" annotations (such as traces for subjects of infinitives, long-distance dependencies and so on). Initially encoded with traces and co-indexes through constituent structures, the processing and recovery of these phenomena entailed complicated algorithms and methods. Nevertheless, the emergence of various conversion algorithms and enrichment processes from the PTB phrase structures to deep syntax representation (e.g LFG F-Structures as in (Cahill et al. [4]), HPSG feature structures (Miyao et al. [18]), or CCG complex lexical types and derivations (Hockenmaier and Steedman [13])) have made these complex syntactic phenomenon more straightforwardly available.

Recently, more semantically oriented "Deep" treebanks have been made available (Čmejrek et al. [7], Flickinger et al. [10]) and their use was popularized through the Semeval 2014 broad semantic shared task (Oepen et al. [20]) which simplified these data set by providing mostly graph-based predicate-argument structure instances of these treebanks (Miyao et al. [19]). It worth noting that providing access to different representation layers of the same source, each having a different degree of granularity in term of syntax-to-semantic interface was, among others such as the MTT, formalized through the Prague Dependency Bank line of work (Böhmová et al. [3], Hajic et al. [12]). Inspired by the LFG theory, the various Stanford dependency schemes (De Marneffe and Manning [8], de Marneffe et al. [9]) are also a milestone in making deep syntax structures easily processable for further downstream semantic processing.

For French, which is the language we focus on, the annotations of the largest treebank available for French (the French Treebank (Abeillé and Barrier [1]), hereafter FTB) are surface-only. However, earlier attempts at deriving deeper representations were carried out by Schluter and Van Genabith [25] within a treebank-based LFG framework, using an heavily modified subset of the initial FTB release. Focusing on delivering a free data set based on structures as close as possible from the current FTB, Candito et al. [6] have defined a deep dependency syntactic annotation scheme for French, and added a deep syntactic layer to the Sequoia Treebank (Candito and Seddah [5]), a freely available corpus, made of 3,099 sentences . Although this resource can be used to train statistical deep parsers for French, we anticipate that its size will be insufficient to train accurate models given the additional complexity of deep syntax with respect to surface syntax.[1] We have thus undertaken to semi-automatically annotate the FTB with deep syntactic annotations, leading to a "silver" deep treebank of 18,500 sentences.

In the following, we start by describing the Deep Syntactic Representations (hereafter DSRs) of (Candito et al. [6]) in section 1, and the methodology used to obtain such representations for the sentences of the FTB. Section 3 is devoted to the tool we designed to convert surface dependency trees into such deep syntactic representations: we describe both the graph-rewriting system (section 3.1) and the hand-craft rules (section 3.2). We provide an evaluation of the DSRs obtained using this tool in section 4, and conclude.

[1] As shown by the mixed level of performance obtained by Ballesteros et al. [2] on a comparable parsing task for Spanish.

1 Target Deep Representations

In order to describe the Deep Syntactic Representations (DSRs) that we target, we sum up their description by Candito et al. [6]. As mentioned in the introduction, deep representations are intended to be an intermediary step between surface syntax and semantic representations. The DSRs make explicit three major types of information with respect to the surface representations:

- First, DSRs make explicit the deep syntactic arguments of verbs and adjectives and "subjects" of adjectives (predicates with other part-of-speech are left for future work). The deep syntactic arguments include those arguments that are syntactically dependent of another head (e.g. the subject of infinitival verbs) or that appear as the surface governor of the predicate (e.g. in the case of an attributive participle: *des personnes parlant italien ((some) people speaking italian)*).
- Second, following Relational Grammar (Perlmutter [21]), predicates are taken to subcategorize for dependents with certain *canonical* grammatical functions, potentially different from their effective *final* functions. The deep arcs are thus labeled with both canonical and final functions (at least for the grammatical functions that can be involved in syntactic alternations). For instance in Figure 1, while *Jean* is both the final and canonical subject of *semble*, it is the final subject and canonical object of the passive form *respecté* (written with a *suj:obj* label).
- Third, the semantically-void tokens are discarded, and dependencies coming in or out from these tokens are shifted to semantically full tokens (e.g. semantically void prepositions or complementizers are bypassed, auxiliaries are discarded and replaced by features on full verbs).

In order to capture syntactic alternations, DSRs make use of the distinction between *canonical* grammatical function (canonical GF) and *final* grammatical function (final GF)[2], and between *canonical subcategorization frames* (canonical SF) and *final subcategorization frames* (final SF). The final SF of an occurrence of a verb is defined as the list of GFs associated to its expressed arguments, plus the GFs that would be associated with the linguistic expressions that would appear as argument, if the verb were used in finite mode and in a non elliptical construction. This formulation accounts for the subject of infinitives, the subject of coordinated verbs or more generally any argument shared by several predicates. For instance, in Figure 1, the final SF both for *compter (to matter)* and for *respecté (respected)* is [subject=*Jean*].

The *deep* syntactic arguments of a verb are defined as the set of linguistic expressions that bear a final GF with respect to that verb, and that are not semantically empty. Syntactic alternations are viewed as redistributions of the grammatical functions associated to the syntactic arguments. Following Relational Gram-

[2] We use the term *canonical* instead of the Relational Grammar term *initial*.

mar (Perlmutter [21]), the final SF is considered as resulting from the application of 0 to *n* redistributions to a canonical SF. A simple case is for instance a passive occurrence of a transitive verb: the final SF is [subject, (by-object)] while the corresponding canonical SF is [objet, (subject)]. So for instance in Figure 1, the canonical SF of *respecté* is [object=*Jean*]. This is shown in the figure with double labels on the arcs of the form *final_function:canonical_function* (hence the label *suj:obj* between *Jean* and *respecté*).

Candito et al. [6] only considered redistributions that are morpho-syntactically marked (for instance with an auxiliary for passives, or a void reflexive clitic *se* for middle or neuter alternations). Unmarked redistributions are not accounted for, because disambiguating them resorts to semantic analysis. For instance, for the verb *couler ('to sink')*, the non-marked causative/inchoative alternation gives rise to two canonical SFs: the two constructions *X coule Y (X sinks Y)* and *Y coule (Y sinks)* are not related in the deep syntactic representation. They get the two distinct canonical SF [subject, object] and [subject] respectively, and for both occurrences, the canonical SF is identical to the final SF. On the contrary, the neuter and middle alternations, which are marked by a void reflexive clitic *se*, are represented using redistributions. For instance, for both *(Paul cassa la vase) Paul broke the vase* and *le vase se brisa* (litt. *the vase SE broke* for *the vase broke*), *vase* is canonical object.

From the formal point of view, the DSRs are graphs, whose nodes are the non-void tokens of the sentence. The arcs are labeled using the canonical functions (hence for instance, the *suj:obj* arc between *Jean* and *respecté* is labeled with *obj* only in the DSR). The DSRs may contain cycles, for instance in the case of an adjective or participle modifying a noun : the modifier dependency is retained in the deep representation, and an inverse arc is added (the noun is a deep syntactic argument of the modifying adjective or participle).

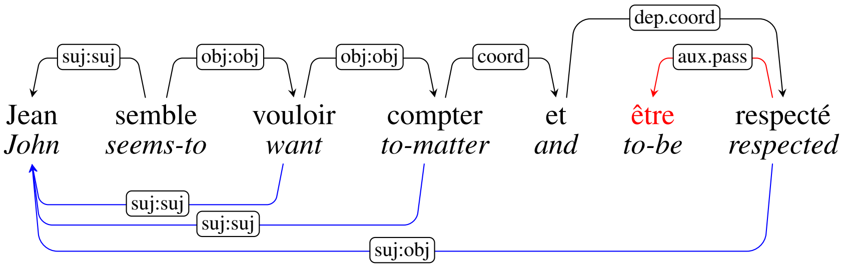


Figure 1: A dependency graph containing both the surface syntactic tree and the deep syntactic representation. The black arcs belong to both representations. The red arc belongs to the surface representation only. The blue arcs below the sentence are "deep-only": they belong to the DSR only.

2 Methodology to obtain more DSRs

In order to obtain pseudo-gold DSRs for the FTB, we used as starting point the surface dependency version of the FTB, as released for the SPMRL Shared Task (Seddah et al. [26]), which contains $18,535$ newspaper sentences. To obtain DSRs for these sentences, we partially re-used the annotation methodology of the Deep Sequoia Treebank, which consisted of three steps (Candito et al. [6]):

(i) Manual annotation of certain phenomena,

(ii) Automatic pre-annotation using two independently designed graph-rewriting systems (Grew (Guillaume et al. [11]) and OGRE (Ribeyre et al. [22]))

(iii) Double manual correction plus adjudication of the divergences.

We applied this methodology on the FTB, but we skipped the last step, which currently seems out of reach given the corpus' size. We retained the dichotomy between manual annotation of certain phenomena (step (i)) and automatic annotation (step (ii)), this time using OGRE only.

The focus of this paper is on the evaluation of step (ii) mainly, so we only briefly list the phenomena that were manually annotated during step (i)[3]: long-distance dependencies, impersonal subjects, causative alternations, cleft sentences, and finally the status of the *se* clitic, which can either be part of a lexicalized pronominal verb (like *s'apercevoir (to realize)*), or mark a reflexive construction (as in *Anna se soigne tout seule (Anna cures herself on her own)*), or mark a middle diathesis alternation (*Ces livres se vendent bien (These books sell well)*) or a neuter diathesis alternation (*Le vase s'est rompu (The vase broke)*). All these phenomena are either highly ambiguous (clitic *se*) and/or rare (long-distance dependencies, causatives, cleft sentences), and their disambiguation resorts to semantic properties that are notoriously difficult to capture in a rule-based approach. By contrast, phenomena which exhibit more systematic syntactic properties, such as raising and control or the passive alternation, are handled automatically at step (ii).

We can now turn to the description of the graph rewriting system and the handcraft rules used at step (ii).

3 Surface to deep tool

3.1 OGRE

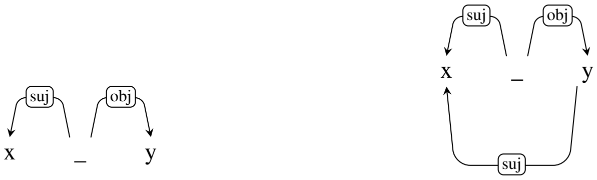
OGRE (for Optimized Graph Rewriting System) is a two-stage graph rewriting system (Ribeyre et al. [22]). The first stage is based on the Single Pushout approach described at length in (Rozenberg [24]) while the second has his roots in the constraint programming paradigm (Rossi et al. [23]).

OGRE uses a set of rules, applied in two stages. A rule is defined by a subgraph pattern called a match, a set of rewriting commands (performed at first stage)

[3] The manual annotations were mainly performed by the second author of this paper, and other colleagues. We hope to be able to describe the manual annotations in another publication.

```
rule add_suj_edge{
    match{
        x <-[label:"suj"]- [] -[label:"obj"]-> y[cat:"VINF"]
    }
    negative{
        y -[]-> x
    }
    commands{
        add_edge(y, x, [label:"suj"]
    }
}
```

(a) Textual form of the rule.



(b) Subgraph pattern (match).

(c) Transformed subgraph.

Figure 2: Example of rule which adds a *suj* edge, in text format (a), with graphical format for the match pattern (b) and for the resulting graph after application (c). The rule contains a Negative Application Condition (NAC), which blocks the application if the *x* node depends on the *y* node.

and/or a set of "triggers"[4] (instantiated at first stage, but activated in the second stage). In addition, a rule may contain negative application conditions (NAC defined in Lambers et al. [14]) which block matches based on certain conditions. An example of rule is given in Figure 2.

During the first stage, rules are applied sequentially. The rewriting commands can add, modify and remove properties on nodes (token, features, POS, ...) and edges (labels, surfacic or deep status), and remove or add edges. In the surface-to-deep rules, removal of edges or features is not used though. Importantly, the match is always performed on the input graph only, independently of the added arcs/features, so the order of the rules does not matter. The set of rewriting commands is applied to the input graph and triggers are instanciated on pairs of nodes, to be used during the second stage.

In the second stage, triggers instantiated at first stage apply until no more edges are added. In the surface-to-deep rules, we only use one of the several trigger types available in OGRE, namely the share triggers[5], which add edges. A share trigger *share(l)* is defined for a pair of nodes (y,z) and a label l. It states that if a $y \xrightarrow{l} x$ arc belongs to the current graph (i.e. if it either belongs to the modified graph from the first stage, or was added by a share trigger), then the arc $z \xrightarrow{l} x$ should be

[4] In (Ribeyre et al. [22]), the term “constraints” was used instead of “triggers”.

[5] Formerly defined as share constraints.

added to it. For example in Figure 3, three share triggers *share(suj)* sequentially add the orange, purple and green edges, in that order, each new edge triggering the applicability of the subsequent trigger. As will be seen in the next section, this system allows to express in a compact way general linguistic constraints such as cascaded control or raising verbs.

Termination of the second stage is guaranteed by the absence of multi-edges with the same label. Moreover, the iterative process combined with the fact that triggers can only add edges ensures the confluence of the system.

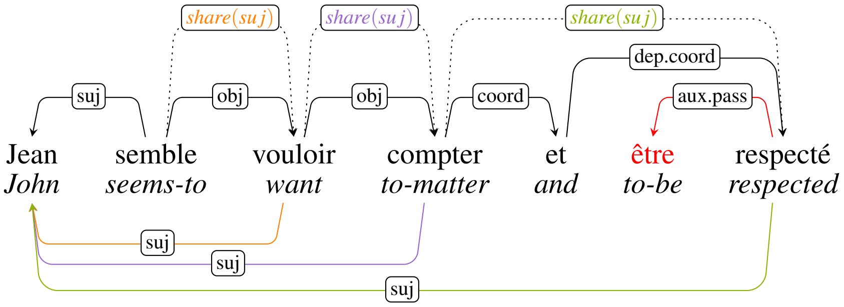


Figure 3: Share-triggers (dotted arcs) for a cascade of raising/control verbs and coordinated verbs. Each share-trigger instance adds the deep-only arc of the same color.

3.2 Rules

The rules for surface to deep syntax conversion are organized into five *modules*, namely sets of rules, designed to be applied sequentially. The rules are partitioned into modules so that arcs or features added within one module can serve as matches for the rules of a subsequent modules : while the rules within a module need not be ordered, the modules themselves are sequentially ordered.

The first module makes verbal tense and mood explicit, converting tense auxiliaries into appropriate features on the lexical verb. For instance, in example 1, the verb *respecté* is a past participle at the surface level, but it bears infinitival mood and past tense at the deep level. This normalization facilitates the writing of rules in subsequent modules.

The second module marks the final subjects of non finite verbs (and by extension, of adjectives also, whether used as predicative complements or noun modifiers). It uses the constraint propagation system of OGRE to handle embeddings involving cascades of predicates and/or coordination. For instance the rule for raising or subject control verbs introducing infinitives contains a share-constraint stating that their subject should also be the final subject of the infinitive. This constraint instantiates for two pairs of nodes in Figure 3 (the orange and purple constraint instantiations), which add *Jean* as final subject of *vouloir* and in turn as final subject of *compter*. We extracted control and raising verbs from the Dicovalence lexicon (van den Eynde and Mertens [27]), and subsequently extended the list during rule tuning on the DeepSequoia dev corpus.

VP coordination is handled through another constraint, stating that for two coordinated verbs, if a final subject exists for the first conjunct, then it must also be added as the final subject of the second conjunct (provided the latter does not initially have a final subject). This is displayed as the green constraint in Figure 3, which adds the final subject of *respecté* as soon as *compter* gets a final subject.

Syntactic alternations are mainly handled in the third module, which identifies canonical functions for arguments of verbs (whether these arguments were already in the surface tree, or added by the second module). For instance, a rule states that if a passive verb has a final subject, then that is its canonical object. Such a rule applies in sentence 1 to identify *Jean* as the canonical object of the passive verb *respecté*. This module interacts with some manual annotations performed at step (i) : while passive verbs are automatically identified, other highly ambiguous alternations are first manually identified at step (i) (causatives, impersonal, middle and neuter alternations), then the rules interpret the manual annotations to correctly derive the canonical functions of the arguments (including the cases of alternation interaction such as impersonal passives). A clear-cut separation between the module for final subjects and the module for syntactic alternations is not possible in particular because of control verbs specificities. The syntactic generalization applicable to control verbs mixes canonical and final functions. Indeed, a given control verb imposes which of its *canonical* argument is the *final* subject of the infinitive. For instance, the verb *condamner (to condemn)* is an object-control verb, meaning that its *canonical* object is the *final* subject of the infinitival clause it introduces. So, in *La cour a condamné Jean à être incarcéré (The court condemned Jean to be incarcerated)*, the object *Jean* is the final subject of the passive verb *incarcéré*. When *condamner* is passive, then the controler of the infinitive, it still holds that its *canonical* object (but final subject) is the final subject of the infinitive, as in *Jean a été condamné à être incarcéré (Jean was condemned to be incarcerated)*. We resolved this interaction by explicitely distinguishing rules for active and passive control verbs in the final-subject module.

A fourth module handles comparative and superlative constructions, mostly. It also adds morphological features such as definiteness in case of determiners, and identifies the clause types (interrogative, imperative...). Finally the last module exclusively deals with the removal and bypassing of semantically empty words. Incoming and outgoing edges of these words are attached to semantically full words.

To give an idea of the degree of complexity of the system, the five modules contain 19, 40, 21, 39 and 36 rules respectively, for a total of 155 rules. While being a reasonable figure, we must admit that the understanding and maintenance of this rule set requires training.

4 Evaluation

4.1 Quantative analysis

We now turn to the evaluation of the Surface-to-deep conversion module and of the quality of the DSRs we obtain for the FTB. The current version of the surface-to-deep conversion module was designed in two stages. As mentioned in section 2, the conversion module was first designed to pre-annotate the DeepSequoia, which has been subsequently manually corrected. More precisely, in order to build the DeepSequoia, the two research teams who produced the DeepSequoia (namely ours and the Sémagramme team) first manually annotated a subset of 247 sentences (called the MINIREF), and then both tuned a conversion tool on this subset. Then two pre-annotated versions of the treebank could be produced, manually corrected by each team, and finally conflicts were manually adjudicated. For the current work, we subsequently improved our conversion rules using another subset of the DeepSequoia. More precisely, we split the DeepSequoia into four parts, as shown in Table 1. We used the DEV2 set to improve the rules' coverage, while setting aside a training set for future experiments, and a test set for final evaluation. Further, in order to evaluate the quality of the DSRs obtained for the FTB, we manually annotated the first 200 sentences from the FTB development set.

The top part of Table 2 concerns the evaluation of the conversion rules on the DeepSequoia test set. We report labeled and unlabeled precision, recall and F-measure when considering either the set of deep edges (first row of Table 2) , or the set of deep-only edges (second row). Performance on this test set is rather high, although a little lower on the deep-only edges.

Sets	#Sent.	#Tokens	#Deep Tokens
TRAIN	2,202	47,415	40,792
DEV-1 (Miniref)	247	5,852	5,038
DEV-2	250	5,360	4,606
TEST	400	8,411	7,264

Table 1: Experimental Split

The bottom part of the table concerns the evaluation on the 200 sentences from the FTB. We proceeded as follows: we applied the surface-to-deep rules on the *reference surface trees*, augmented with the deep manual annotations (cf. step (i) mentioned in section 2), and obtained predicted DSRs (hereafter *Predicted Deep 1*). We manually corrected these predicted DSRs, and also manually corrected some errors in the reference surface trees. We thus obtained *corrected deep* representations and *corrected surface* trees. The line "REFERENCE VS CORRECTED SURFACE" in Table 2 shows the evaluation of the reference surface trees against the corrected surface trees. The next line provides an evaluation of the *Predicted Deep 1* representations against the corrected deep ones. It shows that the overall quality of the resulting deep syntactic corpus is rather good. It can be anticipated that the DSRs obtained for the FTB will have sufficient quality to serve as training data.

Yet, while the evaluation of *Predicted Deep 1* (penultimate row) provides an estimation of the quality of the full set of predicted DSRs for the whole FTB, it mixes errors due to the rules, and errors in the reference surface trees. In order to evaluate the former more precisely, we applied the conversion rules on the *corrected surface* trees, and obtained a second version of predicted DSRs (hereafter *Predicted Deep 2*). The results are shown in the last row of Table 2. We obtain no drop in performance with respect to the evaluation of the DeepSequoia, which indicates that our rule set has a good coverage, and generalizes well to other corpora.

DeepSequoia (test set)	# gold edges	LP	LR	LF	UP	UR	UF
DEEP EDGES	8259	99.5	99.2	99.4	99.5	99.3	99.4
DEEP ONLY EDGES	1806	98.1	97.3	97.7	98.3	97.5	97.9
FTB (200 sent. dev.)	# gold edges	LP	LR	LF	UP	UR	UF
REFERENCE VS CORRECTED SURFACE	6170	98.7	98.0	98.4	100.0	99.4	99.7
PREDICTED DEEP 1	6012	97.5	97.1	97.3	98.9	98.4	98.7
PREDICTED DEEP 2	6012	99.5	99.3	99.4	99.6	99.4	99.5

Table 2: Rules' evaluation (Labelled/Unlabelled recall, precision, F-measure).

4.2 Qualitative analysis

We checked the errors on the 200 sentences from the FTB. A qualitative evaluation reveals that some phenomena are not (properly) handled by the rules, because of their complexity and ambiguity. For example, nominal predicative complements in sentences such as *C'est une femme Capitaine.* (*It's a female captain.*), where an *arg* edge should be added between *femme* and *Capitaine*, are not automatically annotated. Elliptic coordination is another unhandled phenomena, in particular head gapping and argument clusters.

Finally, automatic annotation of infinitive subjects leads to the highest rate of errors. We can distinguish two types: (i) **Control or raising verbs** not present in our lexical resources: *annoncer (to announce)* or *continuer de (to continue to)* are two examples (*continuer* was present, but with preposition *à*). The same goes, for "control nouns". For instance, the noun *idée (idea)* was missing in the rules, which thus fail to assign the possessive as subject of the infinitive verb in *D'où son idée de calmer le jeu. (Hence his idea to calm things down).* (ii) **Arbitrary control**, for certain modifying prepositions introducing infinitive clauses, the rules arbitrarily choose the subject of the main verb as subject of the infinitive, though it is clear that such a simple and systematic rule will fail in some cases. For instance, in *Ils ont re cu les élèves pour visiter le fournil (they received the pupils to visit the bakery)*, the subject of *visiter (to visit)* is not properly found.

Conclusion

In this paper, we described the methodology we used to add a deep syntax annotation layer to the French Treebank. Based on the work carried out by Candito et al. [6] to develop and the DeepSequoia treebank, we enhanced the conversion process from surface trees to obtain state-of-the-art results in term of expected quality as shown by our evaluation on a small gold standard we built from the FTB. Furthermore, we manually corrected a reduced set of difficult constructions. This evaluation suggests that the resulting new data set, a deep syntax version of the FTB, can be used as pseudo-gold data to train deep syntactic parsers, or to extract syntactic lexicons augmented with quantitative information. The Deep French Treebank will be released with the paper (following the original license).

References

[1] Anne Abeillé and Nicolas Barrier. Enriching a French treebank. In *Proc. of LREC*, Lisbon, Portugal, 2004.

[2] Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. Deep-syntactic parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1402–1413, Dublin, Ireland, August 2014.

[3] Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. The prague dependency treebank. In *Treebanks*, pages 103–127. Springer, 2003.

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

[5] Marie Candito and Djamé Seddah. Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Proc. of TALN*, Grenoble, France, 2012.

[6] Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, and Éric De La Clergerie. Deep Syntax Annotation of the Sequoia French Treebank. In *International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Islande, 2014.

[7] Martin Čmejrek, Jan Hajič, and Vladislav Kuboň. Prague czech-english dependency treebank: Syntactically annotated resources for machine translation. In *In Proceedings of EAMT 10th Annual Conference*. Citeseer, 2004.

[8] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.

[9] Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R Bowman, Timothy Dozat, and Christopher D Manning. More constructions, more genres: Extending stanford dependencies. *DepLing 2013*, page 187, 2013.

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

[11] Bruno Guillaume, Guillaume Bonfante, Paul Masson, Mathieu Morey, and Guy Perrier. Grew : un outil de réécriture de graphes pour le TAL. In *Proc. of TALN*, Grenoble, France, 2012. URL `http://hal.inria.fr/hal-00760637`.

[12] Jan Hajic, Jarmila Panevová, Eva Hajicová, Petr Sgall, Petr Pajas, Jan Štepánek, Jiří Havelka, Marie Mikulová, Zdenek Zabokrtskỳ, and Magda Ševcıková Razımová. Prague dependency treebank 2.0. *CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia*, 98, 2006.

[13] Julia Hockenmaier and Mark Steedman. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

[14] Leen Lambers, Hartmut Ehrig, and Fernando Orejas. Conflict detection for graph transformation with negative application conditions. In *Proceedings of the Third International Conference on Graph Transformations*, ICGT'06, pages 61–76, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-38870-2, 978-3-540-38870-8.

[15] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: annotating predicate argument structure. In *Proc. of the workshop on Human Language Technology*, pages 114–119, Stroudsburg, USA, 1994. ISBN 1-55860-357-3. doi: 10.3115/1075812.1075835. URL `http://dx.doi.org/10.3115/1075812.1075835`.

[16] Igor Melčuk. *Dependency syntax: theory and practice*. State University Press of New York, 1988.

[17] Simon Mille, Alicia Burga, and Leo Wanner. AnCora-UPF: A Multi-Level Annotation of Spanish. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 217–226, Prague, Czech Republic, August 2013.

[18] Yusuke Miyao, Takashi Ninomiya, and Junichi Tsujii. Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *Lectures notes in Computer Sciences (proc. of IJCNLP 2004)*, volume 3248, pages 684–693. Springer, 2005.

[19] Yusuke Miyao, Stephan Oepen, and Daniel Zeman. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 335–340, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL `http://www.aclweb.org/anthology/S14-2056`.

[20] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL `http://www.aclweb.org/anthology/S14-2008`.

[21] D.M. Perlmutter. *Studies in Relational Grammar 1*. Studies in Relational Grammar. University of Chicago Press, 1983. ISBN 9780226660509. URL `http://books.google.ca/books?id=EcBbfP1zSLMC`.

[22] Corentin Ribeyre, Djamé Seddah, and Éric Villemonte De La Clergerie. A Linguistically-motivated 2-stage Tree to Graph Transformation. In Chung-Hye Han and Giorgio Satta, editors, *Proc. of TAG+11*, Paris, France, 2012. INRIA.

[23] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006. ISBN 0444527265.

[24] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1997. ISBN 98-102288-48.

[25] Natalie Schluter and Josef Van Genabith. Dependency parsing resources for french: Converting acquired lexical functional grammar f-structure annotations and parsing f-structures directly. In *Proc. of NODALIDA 2009*, Odense, Denmark, 2009.

[26] Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. Overview of the spmrl 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proc. of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA, 2013.

[27] Karel van den Eynde and Piet Mertens. *Le dictionnaire de valence DICOVALENCE : manuel d'utilisation*, 2006. URL `http://bach.arts.kuleuven.be/dicovalence/manuel 061117.pdf`.