

Enhancing PTB Universal Dependencies for Grammar-Based Surface Realization

David King and Michael White

Department of Linguistics

The Ohio State University

Columbus, OH 43210, USA

king.2138@osu.edu, mwhite@ling.osu.edu

Abstract

Grammar-based surface realizers require inputs compatible with their reversible, constraint-based grammars, including a proper representation of unbounded dependencies and coordination. In this paper, we report on progress towards creating realizer inputs along the lines of those used in the first surface realization shared task that satisfy this requirement. To do so, we augment the Universal Dependencies that result from running the Stanford Dependency Converter on the Penn Treebank with the unbounded and coordination dependencies in the CCGbank, since only the latter takes the Penn Treebank's trace information into account. An evaluation against gold standard dependencies shows that the enhanced dependencies have greatly enhanced recall with moderate precision. We conclude with a discussion of the implications of the work for a second realization shared task.

1 Introduction

Surface realization systems employing reversible, broad coverage constraint-based grammars together with statistical ranking mod-

els have achieved impressive results in multiple languages, using a variety of formalisms (HPSG, TAG, LFG, CCG). However, these systems all require somewhat different inputs, making comparative evaluation difficult. In the first surface realization shared task (Belz et al., 2011, henceforth SR-11), which aimed to ameliorate these difficulties, attempts to use grammar-based realizers were unsuccessful, as converting shared task inputs to systematic inputs turned out to be more difficult than anticipated. Subsequently, Narayan & Gardent (2012) demonstrated that grammar-based systems can be substantially improved with error mining techniques, and Gardent and Narayan (2013) showed that augmenting the (shallow) SR-11 representation of coordination to include shared dependencies can benefit grammar-based realizers. White (2014) then showed that even better results can be achieved by inducing a grammar (Kwiatkowski et al., 2011; Artzi and Zettlemoyer, 2013) that is directly compatible with (an enhanced version of) the SR-11 inputs. However, as explained below, subsequent analysis revealed substantial remaining issues with the data, which this paper takes a step towards addressing.

A common thread in work on reversible, constraint-based grammars is emphasis on properly representing unbounded dependencies and coordination. For parsing, this emphasis has been shown to pay off in improved recall of unbounded dependencies (Rimell et al., 2009; Nguyen et al., 2012; Oepen et al., 2014). For realization, however, it remains an open question as to whether approaches based on constraint-based grammars can likewise yield an empirical payoff, given the continuing lack of a common input representation that adequately treats unbounded dependencies and coordination, as these grammars require.

With this issue in mind, White (2014) experimented with a version of the shallow SR-11 inputs (created by Richard Johansson) which included extra dependencies for unbounded dependencies and coordination, yielding dependency graphs extending core dependency trees. Unlike the rewrite rules employed by Gardent and Narayan (2013), the extra dependencies were derived from the gold traces in the Penn Treebank (Marcus et al., 1993, PTB), which is necessary to adequately handle right node raising and relativization. However, this version was still found to be incomplete, in particular because it was missing cases where the extra dependencies are encoded structurally in the PTB.

Since then, Universal Dependencies (Nivre et al., 2016, UDs), which aim to represent syntactic dependencies similarly across languages, have become increasingly prominent. Building on the enhanced Stanford dependencies for English (de Marneffe et al., 2013)—which were designed to properly represent unbounded dependencies in dependency graphs—enhanced UDs for English have been partially implemented in the Stanford Dependency Converter (Schuster and Manning, 2016, SDC). The SDC transforms automatic or gold PTB-

style trees into UDs; unfortunately, however, it was not designed to take traces into account, and thus the treatment of unbounded dependencies and coordination is only heuristic. To address this impasse, in this paper we report on progress towards creating SR-11-style realizer inputs that are both based on enhanced UDs and which accurately represent unbounded dependencies and coordination. To do so, we augment the UDs that result from running the SDC on the PTB with the dependencies in the CCGbank (Hockenmaier and Steedman, 2007), since the latter includes lexicalized dependencies derived from gold PTB traces.

2 Background

Figures 1–2 show an example where the CCGbank preserves the information provided by the trace in a free relative clause along with a crucial structurally encoded dependency. In Figure 1 (left), the unbounded dependency between *what* and *achieve* is annotated via a trace in the PTB. Figure 1 (right) shows the SDC output for the sentence. While the SDC manages to capture the unbounded dependency in this case, *what* is not recognized as the head of the free relative clause and there is no direct dependency from the copula to *what*, contrary to de Marneffe et al.’s (2013) specifications. The inadequacy of the representation here—which is essentially the same as the SR-11 representation for the sentence—has serious implications for realization, as it will be difficult for any realizer to determine that *what* should appear at the start of the free relative clause rather than following *achieve*, where direct objects would normally appear (or perhaps sentence initially). By contrast, Figure 2 shows how the Combinatory Categorical Grammar (Steedman, 2000; Steedman and Baldridge, 2011, CCG) derivation yields

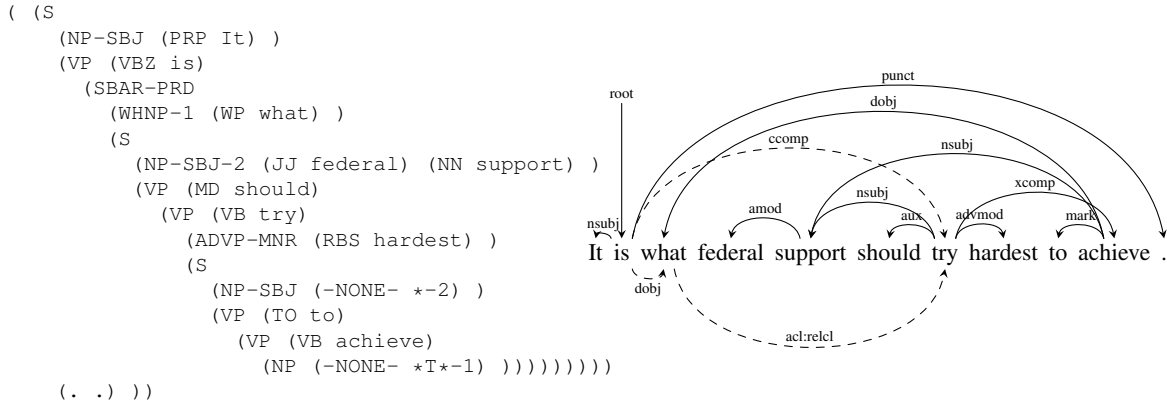


Figure 1: Left: An example of PTB annotation of a free relative clause (*wsj_2412.80*); note the co-indexation between the trace **T*-1* and *WHNP-1*. Right: Automatic SDC output of the gold annotated PTB structure treats *try* as the head of the copula’s complement and has no direct dependency from the copula to *what*, contrary to de Marneffe et al.’s (2013) specifications. Dashed lines show new (below) and deleted (above) dependencies in this work.

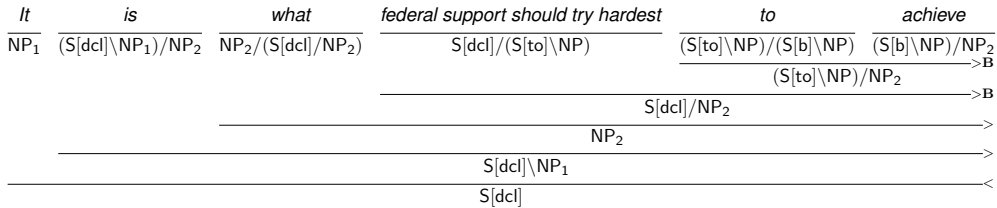


Figure 2: Abbreviated CCGbank derivation showing how PTB trace information is preserved via NP co-indexation; composition operations enable the missing object of *achieve* to be passed up to where its role can be filled by *what*.

a structural dependency between *what* and *is* along with the unbounded dependency between *what* and *achieve*. (More commonly, the SDC captures the structural dependency but not the unbounded one, especially in right node raising and object relatives.)

3 Using the CCGbank to Augment PTB Universal Dependencies

Unlike UD, CCGbank dependencies are numeric and depend on the lexical category of the functor (e.g. *what* fills the second argument of the category for *achieve* in Figure 2). To determine UD labels, we employ a maxent classifier taking information from CCGbank as in-

put. Comparing the CCGbank and SDC output, the classifier is trained where their dependencies overlap and predicts both a label and head-dependent direction. Features used are functor and argument categories; functor and argument tokens; functor and argument POS tags; and functor and argument relative directionality.

Our system for augmenting the SDC’s PTB output begins by combining the SDC basic and enhanced output, since the basic representation does not skip words while the enhanced representation already includes many correct extra dependencies. The system then scans the SDC output and CCGbank for 3 triggers: (i) shared arguments in coordination (e.g. shared objects

		All items	Without CCG gaps
exact	SDC	51	51
	System	82**	86**
unlabeled	SDC	57	57
	System	86**	91**

Table 1: Results from Rimell et al’s (2009) dev set (Wall Street Journal portion). ** – $p < 0.01$

		All items	Without CCG gaps
exact	SDC	42	44
	System	68***	70***
unlabeled	SDC	50	52
	System	75***	78***

Table 2: Results from Rimell et al’s (2009) test set (Wall Street Journal portion). *** – $p \leq 0.0001$

in right node raising), (ii) CCGbank unbounded dependency annotations, and (iii) underspecified SDC *dep* relations (i.e. instances where the SDC cannot determine the appropriate dependency relation). In each case, the maxent classifier is used to predict UD labels for the CCGbank dependencies in question. Predictions are only added to the corpus if there is no (non-*dep*) SDC dependency already present. In addition, *ccomp* and *csubj* relations that co-occur with free relatives are remapped to make the relative the head of the clause. Finally, structural changes for coordination and compounding along SR-11 lines are carried out.

4 Evaluation

The system’s recall was evaluated on Rimell et al.’s (2009) unbounded dependency corpus, a hand-curated corpus with gold annotations for constructions including object free relatives, right node raising, subject extraction, and object extraction. During the creation of CCGbank, some problematic sentences involving gapping were left out of the CCGbank. As a result, we evaluate the system using four different criteria: with and without the skipped CCG sentences,

and with both exact and unlabeled matches. Tables 1 and 2 show significant improvements across the board over the SDC.

Precision was evaluated by manually examining 401 predictions from the system’s output to see whether the proposed edits adhered to UD specifications. Precision from the converter is 70% for exact label matches and 91% for unlabeled matches.

5 Discussion and Future Work

We have adapted and extended White’s (2014) CCG induction algorithm to work with the augmented UDs that our system produces. White’s algorithm assumed CCG phrases are only rarely projected from a dependent rather than a head—e.g., where an NP is projected from a determiner, which is a dependent of the head noun—and thus could be easily handled by handcrafted lexical entries. Since such cases are very common in UDs, the algorithm needed to be extended to induce such categories automatically. Once this was done, the algorithm yielded complete derivations in most cases (approx. 94%). In particular, derivations were induced that captured all but one of the extra dependencies in Table 1 that appear in the CCGbank dev section, and realization experiments with the UD-based representations are underway.

With the augmented UD reported in this paper, we expect the resulting dependency graphs to serve as a promising basis for a second surface realization challenge (with using just the basic dependency trees as an option). A remaining obstacle, however, are the dependent cluster and gapping cases in the PTB, for which the SDC produces rather degenerate output. A promising avenue here would be to adapt Gardent and Narayan’s (2013) method of enhancing the SR-11 representations for these cases.

Acknowledgments

We thank Marie-Catherine de Marneffe, Michael Elsner, the OSU Clippers group and the anonymous reviewers for helpful feedback. This work was supported in part by NSF grant 1319318.

References

- Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *TACL*, 1:49–62.
- Anja Belz, Michael White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proc. ENLG*.
- Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R Bowman, Timothy Dozat, and Christopher D Manning. 2013. More Constructions, More Genres: Extending Stanford Dependencies. In *Proc. DepLing 2013*.
- Claire Gardent and Shashi Narayan. 2013. Generating elliptic coordination. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 40–50.
- Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.
- Shashi Narayan and Claire Gardent. 2012. Error mining with suspicion trees: Seeing the forest for the trees. In *Proc. COLING*.
- Luan Nguyen, Marten Van Schijndel, and William Schuler. 2012. Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of COLING 2012*, pages 2125–2140.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.
- 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 *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.
- Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded Dependency Recovery for Parser Evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 813–821.
- Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Mark Steedman and Jason Baldridge. 2011. Combinatory Categorial Grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell.
- Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.
- Michael White. 2014. Towards surface realization with CCGs induced from dependencies. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 147–151.