RRGbank—creating a Role and Reference Grammar resource through an automatic conversion of the Penn treebank

Tatiana Bladier[1], Andreas van Cranenburgh[2], Kilian Evang[1], Laura Kallmeyer[1], Robin Möllemann[1], Rainer Osswald[1]
[1]Heinrich Heine University Düsseldorf
[2]University of Groningen
{bladier, evang, kallmeyer, moellemann, osswald}@phil.hhu.de
a.w.van.cranenburgh@rug.nl

Introduction. Wide empirical coverage is a touchstone for every grammatical theory. The present paper describes an ongoing effort to develop annotated corpora for Role and Reference Grammar (RRG, [8, 7]). RRG is a functional theory of grammar strongly inspired by typological concerns and aiming at integrating syntactic, semantic and pragmatic levels of description. RRG is intended to serve as an explanatory theory of grammar as well as a descriptive framework for field researchers. A key assumption of the RRG approach to syntactic analysis is a *layered structure* of the clause: The *core* layer consists of the *nucleus*, which specifies the (verbal) predicate, and its arguments. The *clause* layer contains the core plus extracted arguments, and each of the layers can have a *periphery* for attaching adjuncts. Another important feature of RRG is the separate representation of *operators*, which are closed-class morphosyntactic elements for encoding tense, modality, aspect, etc., which are attached to specific constituent layers depending on their type. The ordering among the operators systematically correlates with the scope given by their attachment site at the layered structure. The surface order of the operators relative to arguments and adjuncts, however, often requires crossing branches (see Figure 1).

RRGbank. Providing a treebank resource to the RRG community is interesting for several reasons: (i) it will be a valuable resource for corpus-based investigations in the context of linguistic modeling using RRG and in the context of formalizing RRG (which is needed for a precise understanding of the theory and for using it in NLP contexts). Efforts towards a formalization of RRG as a tree-rewriting grammar have already been made recently [6, 4]. (ii) In the context of implementing precision grammars, at least for English, an RRG treebank is useful for testing the grammar and evaluating its coverage. (iii) It will enable supervised data-driven approaches to RRG parsing (grammar induction and probabilistic parsing). (iv) Finally, and more immediate, the specification of the treebank transformation yields valuable new insights into RRG analyses of English syntax.

Since manual annotation is very time-consuming, we decided to (semi-)automatically derive RRGbank from the Penn Treebank (PTB). The PTB has been used in the past, among others, for deriving CCGbank, a corpus of Combinatory Categorial Grammar derivations [3]. A somewhat different route is taken by the LinGO and ParGram approaches to dynamic treebanking for HPSG and LFG, respectively [2]. These projects made use of manually developed grammars and parsers for the grammar formalisms in question, and then manually checked and corrected the parse results. This is not an option for developing RRGbank at the moment, but dynamic treebanking might be worthwhile after a first wide-coverage RRG grammar has been extracted from RRGbank.

Differences between PTB and RRG. Although syntactic representation of the trees in PTB differs from the layered structures of the RRG, the information needed for converting a PTB tree to a derivation in RRG is implicitly provided by the node labels and functional marks in PTB. Node labels in PTB also store information about sentential operators (such as tense, aspect, modality etc.), which RRG represents through operator projections attached at different clausal layers. In our work, we do not use a separate operator projection structure [7], but mark this information with additional features on the nodes (e.g. *[+OP]*), as shown in Figure 1 [5]. While PTB uses *traces* as placeholders for the cases of the argument shifting (for example, by *wh-extraction*), the notion of trace is absent in RRG. Instead, the traces in PTB have to be treated in different ways depending on the kind of the argument movement.

General methodology. After converting a small set of sentences to RRG by hand, we defined automatic transformation rules for particular constituents and constructions. We will proceed with this process starting from the most frequent constituents, with the aim of covering the whole treebank. The goal is to be able to automatically transform the whole PTB with rules, although in case of particular rare constructions or

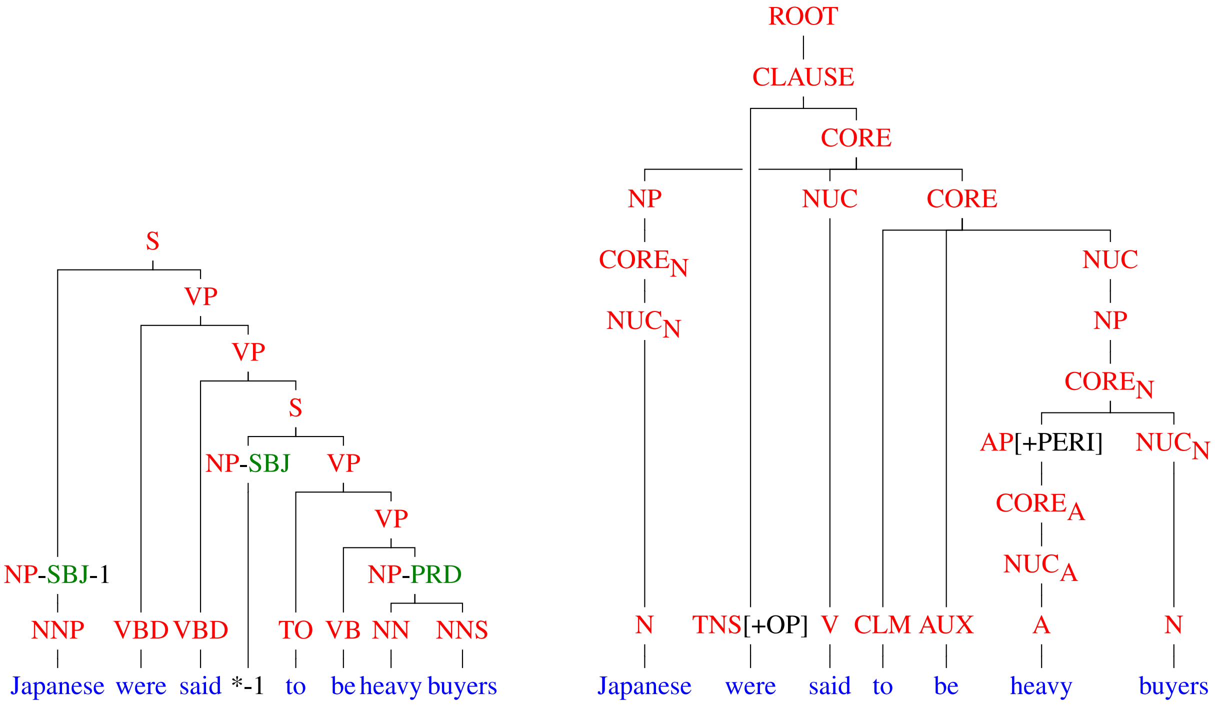

Figure 1: An example of a sentence from PTB (left tree) converted to RRG (right tree).

annotation errors [1], highly specific rules may need to be introduced for individual sentences; this avoids the need for a separate manual annotation and correction step.

Evaluation and future work. We measure how much of a PTB tree is already covered by our transformation rules using EVALB bracketing scores. The end goal is zero common bracketings between the PTB and converted RRG trees, which implies full coverage. Correctness is monitored by maintaining a set of manually-corrected RRG trees used as regression tests to confirm that changes to the rules do not introduce errors. In the future we plan to use the RRGbank for implementation of an RRG-based syntax-semantics tool for grammar development and parsing within the TreeGraSP project (https://treegrasp.phil.hhu.de/).

References

[1] Don Blaheta. Handling noisy training and testing data. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 111–116. Association for Computational Linguistics, 2002.

[2] Dan Flickinger, Valia Kordoni, and Yi Zhang. Deepbank: A dynamically annotated treebank of the wall street journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, Lisbon, Portugal, 2012.

[3] Julia Hockenmaier and Mark Steedman. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3), 2007. URL http://www.aclweb.org/anthology/J07-3004.

[4] Laura Kallmeyer and Rainer Osswald. Combining predicate-argument structure and operator projection: Clause structure in role and reference grammar. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 61–70, Umeå, Sweden, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-6207.

[5] Laura Kallmeyer and Rainer Osswald. Towards a formalization of role and reference grammar. In R. Kailuweit, L. Künkel, and E. Staudinger, editors, *Applying and Expanding Role and Reference Grammar*, to appear.

[6] Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. Tree wrapping for Role and Reference Grammar. In G. Morrill and M.-J. Nederhof, editors, *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer, 2013.

[7] Robert D. Van Valin, Jr. *Exploring the Syntax-Semantics Interface*. Cambridge University Press, 2005.

[8] Robert D. Van Valin, Jr. and Randy LaPolla. *Syntax: Structure, meaning and function*. Cambridge University Press, 1997.