Automatic Extraction of Tree-Wrapping Grammars for Multiple Languages

Tatiana Bladier, Laura Kallmeyer, Rainer Osswald, Jakub Waszczuk
Heinrich Heine University Düsseldorf, Germany
{bladier,kallmeyer,osswald,jakub.waszczuk}@phil.hhu.de

Abstract

We present an algorithm for extracting Tree-Wrapping Grammars (TWGs) for multiple languages from constituency treebanks. The TWG formalism, which is inspired by Tree Adjoining Grammar (TAG), has been developed for the formalization of Role and Reference Grammar (RRG). We describe the extraction of TWGs for English, German, French and Russian from the multilingual RRG corpus RRGparbank. A special focus is given to how non-local dependencies are treated by the extraction algorithm. In TWGs, non-local dependencies are considered as arising from local dependencies in elementary trees by the operation of 'wrapping substitution'. The extracted grammars are validated by using them in a subsequent parsing step.

1 Background: Tree-Wrapping Grammars

The Tree Wrapping Grammar (TWG) formalism (Kallmeyer et al., 2013; Kallmeyer, 2016; Osswald and Kallmeyer, 2018) is a tree-rewriting formalism much in the spirit of Tree Adjoining Grammar (TAG) (Joshi and Schabes, 1997) that has been developed for the formalization of Role and Reference Grammar (RRG) (Van Valin, 2005; Van Valin, 2010), a theory of grammar with a strong emphasis on typological concerns. A TWG consists of a finite set of elementary trees which can be combined by the following three operations: a) *(simple) substitution* (replacing a leaf by a new tree), b) *sister adjunction* (adding a new tree as a subtree to an internal node), and c) *wrapping substitution* (splitting the new tree at a d(ominance)-edge, filling a substitution node with the lower part and adding the upper part to the root of the target tree). As in (lexicalized) TAG, the elementary trees of a TWG are assumed to encode the argument projection of their lexical anchors. Figure 1 shows an application of wrapping substitution for generating the German sentence in (1) (the dashed line indicates a d-edge).[1]

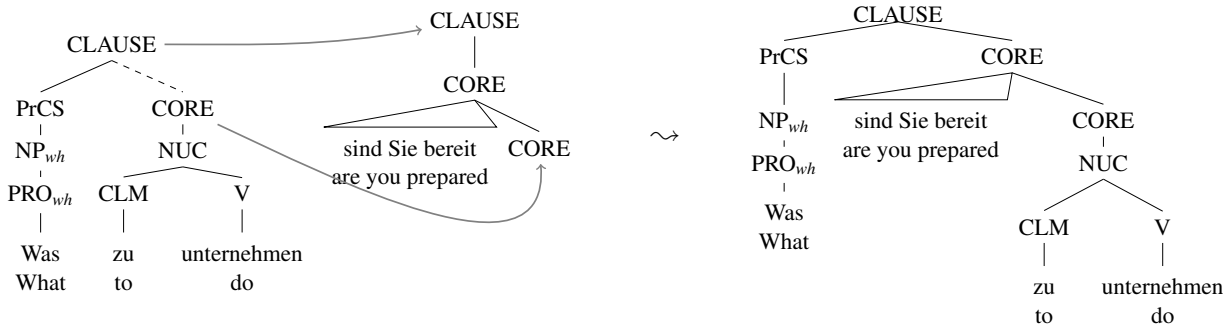

Figure 1: Wrapping substitution for the construction in (1).

(1) *Was sind Sie bereit zu unternehmen ?*
What are you prepared to do ?

[1] Abbreviations: NUC = Nucleus, PrCS = Precore slot. All examples are taken from George Orwell's novel '1984' or its published translations.

The example illustrates how non-local dependencies, here a wh-extraction across a control construction, can be generated by wrapping substitution from local dependencies in elementary trees.

TWG are more powerful than TAG (Kallmeyer, 2016). The reason is that a) TWG allows for more than one wrapping substitution stretching across specific nodes in the derived tree and b) the two target nodes of a wrapping substitution (the substitution node and the root node) need not come from the same elementary tree, which makes wrapping non-local compared to adjunction in TAG. The latter property is in particular important for modeling extraposed relative clauses (see example (3) for a deeper embedded antecedent NP, which requires a non-local wrapping substitution).

In this paper, we adopt a slightly generalized version of wrapping substitution which allows the upper part of the split tree, provided that the upper node of the d-edge is the root, to attach at an inner node of the target tree. For instance, in Figure 1 an additional SENTENCE node above the CLAUSE node in the tree of *bereit* ('prepared') would be possible. A further example for this generalized wrapping will be discussed in Figure 2 below.

By using TWG as a formalization of RRG and applying it to multilingual RRG treebanks, we aim at extracting corpus-based RRG grammars for different languages, thereby obtaining in particular a cross-linguistically valid "core" RRG grammar and, furthermore, providing a cross-lingual proof of concept for TWG in general with respect to its ability to model non-local dependencies. The work presented in this paper is a first step towards these goals.

2 Non-local dependencies in RRGparbank

RRGparbank is part of an ongoing project to create annotated treebanks for RRG (Bladier et al., 2018; Bladier et al., 2019).[2] RRGparbank provides parallel RRG treebanks for multiple languages. At present, RRGparbank contains George Orwell's novel '1984' and its translations in several languages.[3]

RRGparbank provides annotations of non-local dependencies (NLDs) including those given by long-distance wh-extraction (2a), relativization (2b), topicalization (2c), and extraposed relative clauses (2d).

(2) a. *What do you think you remember?*
b. *[. . .] two great problems, which the Party is concerned to solve.*
c. *By such methods it was found possible to bring about an enormous diminution of vocabulary.*
d. *Nothing has happened that you did not foresee.*

In the present context, 'non-local' means that the dependency is not represented within a single elementary tree. We refer to non-local wh-extraction, relativization and topicalization as long-distance dependencies (LDDs).

In RRGparbank, LDDs are annotated in the following way: The fronted phrase node carries a feature PRED-ID whose (numerical) value coincides with the value of the feature NUC-ID of the NUCLEUS the fronted phrase semantically belongs to. For instance, in the annotation of sentence (1), the NP_{wh} node in the tree shown on the right of Figure 1 is marked by [PRED-ID 1] while the NUC node above *unternehmen* is marked by [NUC-ID 1]. See Figure 3 for another example of the annotation convention. In the case of extraposed relative clauses, the relative pronoun and the NP modified by the relative clause both carry the feature REF with identical values (cf. Figure 4).

3 Deriving non-local dependencies by wrapping substitution

Similar to TAG, (simple) substitution in TWG represents the mode of tree composition for expanding argument nodes by the syntactic representations of specific argument realizations, while sister adjunction is mainly used for adding peripheral structures (i.e., modifiers) to syntactic representations. Wrapping substitution, on the other hand, is used for linguistic phenomena in which an argument is displaced from its canonical position and which cannot be handled by simple substitution or sister adjunction

[2] https://rrgparbank.phil.hhu.de/

[3] The data are partly taken from the MULTEXT-East resource (Erjavec, 2012).

(Kallmeyer et al., 2013; Osswald and Kallmeyer, 2018). This holds in particular for the cases of non-local dependencies (NLDs) listed in Section 2. The TWG derivation of LDDs by means of wrapping substitution follows basically the pattern illustrated by the example in Figure 1.

Extraposed relative clauses (ERCs), as in (2d), represent a different type of NLD, namely the extraction of a modifier (the relative clause), typically to a position to the right of the CORE, which leads to a non-local coreference link between the relative clause and its antecedent NP. Example (2d) can be analyzed using wrapping as shown in Figure 2. The extraposed relative clause is associated with a tree that contributes a periphery CLAUSE below a CLAUSE node while requiring that an NP node (which serves to locate the antecedent NP) is substituted into an NP node somewhere below the CLAUSE, modeled by a d-edge between the upper CLAUSE node and a single NP node without daughters. This NP is a substitution node that gets filled with the actual antecedent NP tree. Put differently, the antecedent NP merges with this single NP node, which establishes the link to its modifying relative clause.

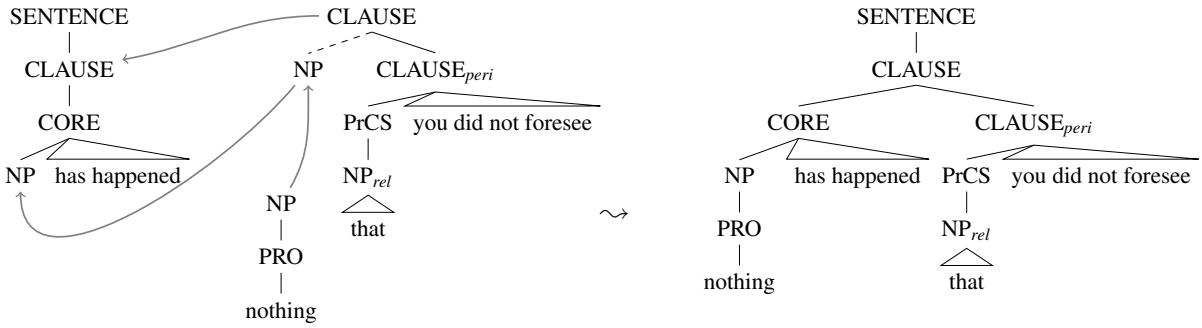

Figure 2: Wrapping substitution for the extraposed relative clause from (2d)

In RRGparbank, we encountered cases where the antecedent NP is further embedded and also cases with more than one relative clause modifying the same antecedent. (3) is an example where we have both: The antecedent NP *Menschen* ('people') is embedded in the direct object NP, and we have two extraposed relative clauses, both modifying the same antecedent.

(3) *Unzählige Male hatte sie [...] [die Hinrichtung von Menschen]$_{NP}$ gefordert , [deren Namen sie nie*
Numerous times had she [...] the execution of people demanded , whose names she never
zu vor gehört hatte] [und an deren angebliche Verbrechen sie nicht im entferntesten glaubte] .
before heard had and in whose alleged crimes she not in the least believed .
'On numerous occasions, she had [...] demanded the execution of people whose names she had never heard before and in whose alleged crimes she did not even remotely believe.'

Another interesting phenomenon is illustrated by the Russian example in (4), which shows both wh-extraction (*čto*) and topicalization (*ja*).

(4) *Ja vot čto xoču skazat'.*
I here what want to.say
'What I'm trying to say is this.'

The current annotation in RRGparbank presumes a scrambling analysis of this topicalization, which gives rise to an RRG tree with crossing branches not generated by sister adjunction. This case is not yet covered by the extraction algorithm presented in Section 4.

4 TWG Extraction

To extract TWGs from treebanks, we adapt the top-down algorithm from (Xia, 1999) for TAG. While substituting and sister-adjoining trees can be extracted following the procedure described in (Xia, 1999), we developed a new algorithm to extract d-edge trees which we describe in more detail below.[4] Since TWGs do not allow for trees to have crossing branches, but the RRG trees often contain them, such edges need to

[4]Additional information on the extraction algorithm can be found in (Bladier et al., 2020).

be removed following a rule-based algorithm for re-attaching certain subtrees in the original tree in a pre-processing step. The process of decrossing tree branches concerns only local re-attaching of peripheral constituents and operator projections and can be reverted applying a rule-based back-transformation algorithm after the parsing step. We extract lexically unanchored elementary tree templates (i.e. *supertags*) for the TWGs. The lexical anchoring happens in the subsequent parsing step.

1. **Decross tree branches.** First, for local discontinuous constituents (for instance NUCs consisting of a verb and a particle in German), we split the constituent into two components (e.g., NUC1 and NUC2), both attached to the mother of the original discontinuous node.
   Second, if a tree τ still has crossing branches, the tree is traversed top-down from left to right and among its subtrees those trees are identified whose root labels contain one of the following strings: OP-, -PERI, -TNS, CDP, or VOC. For each such subtree γ in question with r being its root, we choose the highest node v below the next left[5] sibling of r such that the rightmost leaf dominated by v immediately precedes the leftmost leaf dominated by r. If r and v are not yet siblings, γ is reattached to the parent of v. If the subtree in question has no left siblings, it is reattached to the right in a corresponding way. After this step, it should be checked if the tree τ still contains crossing branches. If yes, the process of decrossing branches is continued by applying the steps above to the next subtree in question.

2. **Extract NLDs.** Then we traverse each tree τ in a top-down left-to-right fashion and check for each subtree of τ whether it contains the following special markings for NLDs in its root label: PREDID=, NUCID= or REF=. The indexes identify the parts of the NLD which belong together. In case of an LDD, the parts of the minimal subtree which contain both parts of the LDD are extracted within a single tree with a *d-edge* (see the multicomponent NUC and CORE in Figure 3). The substitution site and the mother node are added to the remaining subtree in order to mark the nodes on which the wrapping substitution takes place (see Figure 3). A similar process is applied to extract ERCs.

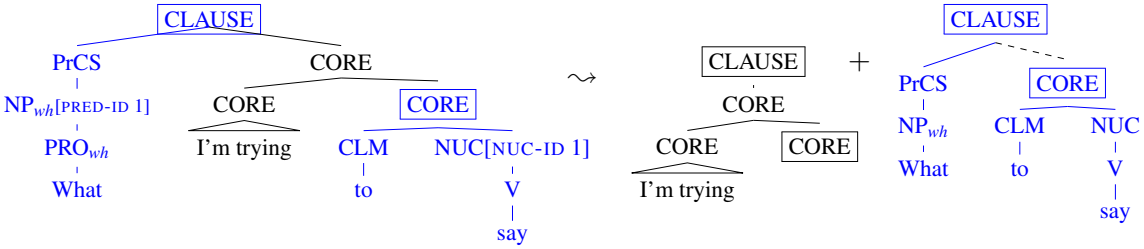

Figure 3: Extraction of tree with a d-edge for an LDD

   The antecedent and the following relative clause (marked with feature REF) are extracted to form a single d-edge tree. The antecedent of the extraposed relative clause is then removed from this d-edge tree and replaced by a substitution slot, as represented in Figure 4.
   After this step, an empty agenda is created and the extracted tree chunks and the pruned tree τ with the remaining nodes are placed into the agenda.

3. **Extract initial and sister-adjoining trees.** If no agenda with tree chunks was created in the previous step, an empty agenda is created in this step and the entire tree τ is placed into it. Each tree chunk in the agenda is traversed and the percolation tables are used to decide for each subtree $\tau_1 \ldots \tau_n$ in the tree chunk whether it is a head, a complement or a modifier with respect to its parent. Initial trees for identified complements and sister-adjoining trees for identified modifiers are extracted recursively in the top-down fashion until each elementary tree has exactly one anchor site.

5 Evaluation of extracted TWGs

We extracted four TWGs for English, German, French, and Russian from the subcorpora of RRGparbank. We used silver and gold annotated data for our experiments, which means that each sentence was

[5] A node v_1 is left to another node v_2 if the leftmost leaf dominated by v_1 is left of the leftmost leaf dominated by v_2.

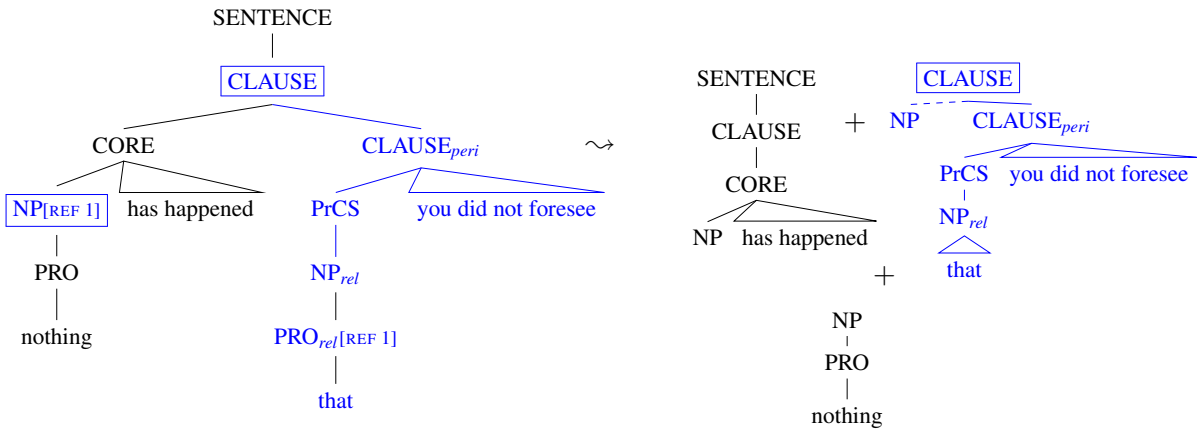

Figure 4: Extraction of a tree with a d-edge for an ERC

annotated and verified manually by at least one linguist. Table 1 provides statistics on the used annotated subcorpora from RRGparbank[6] and the occurrences of non-local dependencies (LDDs and ERCs) in subcorpora. NLDs are generally a relatively rare linguistic phenomenon (Candito and Seddah, 2012; Bouma, 2018). Compared to the other three languages, German shows a fairly large number of ERCs due to its dominant verb-final word order which does not allow putting heavy NPs at the end of the sentence.

Parameters	**English TWG**	**German TWG**	**French TWG**	**Russian TWG**
# word tokens	76893	41324	10550	35975
# word types	7193	7372	2571	9996
Avg. sentence length	14.12	13.5	12.4	10.03
# sentences	5445	3062	851	3586
# LDDs	58	13	36	27
# ERCs	8	110	4	0

Table 1: Statistics on annotated subcorpora in RRGparbank.

The extracted TWGs show a relatively large amount of supertags, more than a half of which occur only once in the corpus. Table 2 shows some statistics on the extracted grammars. The number of supertags with d-edges (which are used for wrapping substitution) is relatively low since the cases of NLDs are not frequent in the data.

Parameters	**English TWG**	**German TWG**	**French TWG**	**Russian TWG**
# supertags	3340	2591	947	2272
# supertags occuring once	1994	1689	584	1503
# initial trees	1727	1490	483	1350
# sister-adjoining trees	1571	1031	431	898
# d-edge trees	42	70	33	22
# nominal supertags	366	299	99	290
# verbal supertags	1382	1164	395	957

Table 2: Statistics on extracted TWG grammars.

We measured the similarity of the extracted TWGs for each language pair. In Table 3 we show the proportions of supertags in one grammar contained in the other grammar[7] (for example, the cell with the row name 'English TWG' and the column name 'German TWG' shows how many supertags from the German TWG are contained in the English grammar). The numbers show that the extracted grammars

[6]The annotation process of the subcorpora in RRGparbank is still in progress and the coverage of annotated sentences differs across the languages. Currently, around 81% of English data, 47% of German, 12% of French, 54% of Russian, and 15% of Farsi sentences are annotated.

[7]Please note that the annotation for different languages in RRGparbank is still in progress, and the proportion of common supertags can change in future.

tend to have a large number of supertags in common. For example, the smallest grammar French TWG (947 supertags) has around 55% supertags in common with the largest grammar for English (3340 supertags). There are 263 supertags common to all four grammars. In future work, we plan to explore the extent to which common supertags in grammars of different languages can be beneficial for multilingual parsing.

Common supertags	**English TWG**	**German TWG**	**French TWG**	**Russian TWG**
English TWG	–	24.97 (834)	15.45 (516)	21.8 (728)
German TWG	32.19 (834)	–	15.51 (402)	24.9 (645)
French TWG	54.49 (516)	42.45 (402)	–	37.80 (358)
Russian TWG	32.04 (728)	28.4 (645)	15.76 (358)	–

Table 3: Ratio of common supertags across language pairs in percents and in numbers (in brackets).

We used the TWG parser ParTAGe (Waszczuk, 2017; Bladier et al., 2020) in a symbolic way in order to validate our grammars and to check that the elementary trees in the extracted TWGs can be combined to produce the original trees.[8] While the majority of sentences could be processed by the parser (see Table 4), some complex sentences which contain an ERC resulting from the free-order placement of predicate arguments as in (4) above could not be parsed. We address these cases in our future work.

	English TWG	**German TWG**	**French TWG**	**Russian TWG**
% exactly matching parses	81	79.07	78.86	80.68
# not parsed sentences	13	8	5	10

Table 4: Validation of extracted TWGs on symbolic parsing with TWG parser ParTAGe.

6 Summary and future work

We presented work in progress on the extraction of TWGs for several languages from the multilingual treebank corpus RRGparbank. TWG is a tree-rewriting system developed for the formalization of Role and Reference Grammar (RRG). TWG is related to TAG and allows, among others, the adequate representation of non-local dependencies (NLDs) in sentences using the operation of wrapping substitution. We showed how wrapping substitution can be used to model various cases of NLDs, including long-distance relativization, long-distance wh-movement, long-distance topicalization, and extraposed relative clauses. We noticed cross-linguistic differences concerning the frequency of NLDs and the corresponding applications of wrapping substitution. At the same time, we observed a considerable overlap of supertags in the TWG grammars extracted for different languages. We validated the extracted grammars using a revised version of the TWG parser ParTAGe.

In future work, we plan to extract larger grammars from the RRG corpora (as the annotation of these projects progresses) and to use them in probabilistic parsing experiments. We also intend to include other languages from RRGparbank into parsing experiments, for example Hungarian and Farsi, depending on the availability of annotated data. Moreover, we will explore how wrapping substitution can be applied to model further linguistic phenomena, such as the variable placement of predicate arguments in languages with a relatively free word order. Finally, we plan to perform multilingual TWG parsing experiments, hopefully benefiting from the considerable number of common supertags across the extracted grammars.

Acknowledgements

We would like to thank three anonymous reviewers for their valuable comments. The work presented in this paper has been partially funded by the European Research Council, within the ERC grant TreeGraSP.

References

Tatiana Bladier, Andreas van Cranenburgh, Kilian Evang, Laura Kallmeyer, Robin Möllemann, and Rainer Osswald. 2018. RRGbank: a Role and Reference Grammar Corpus of Syntactic Structures Extracted from the

[8]For statistical TWG-parsing for English see (Bladier et al., 2020).

Penn Treebank. In *Proceedings of International Workshop on Treebanks and Linguistic Theory TLT17*, Oslo, December.

Tatiana Bladier, Kilian Evang, Laura Kallmeyer, Robin Möllemann, and Rainer Osswald. 2019. Creating RRG treebanks through semi-automatic conversion of annotated corpora. Abstract presented at the International Conference on Role and Reference Grammar 2019, Buffalo, USA, August 19-21, 2019.

Tatiana Bladier, Jakub Waszczuk, and Laura Kallmeyer. 2020. Statistical Parsing of Tree Wrapping Grammars. In *Proceedings of COLING*, December. To appear.

Gosse Bouma. 2018. Corpus-evidence for true long-distance dependencies in Dutch. *Grammar and Corpora*, pages 337–356.

Marie Candito and Djamé Seddah. 2012. Effectively long-distance dependencies in French: Annotation and parsing evaluation.

Tomaž Erjavec. 2012. MULTEXT-East: Morphosyntactic resources for central and eastern european languages. *Language Resources and Evaluation*, 46(1):131–142.

Aravind K Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer.

Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. 2013. 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.

Laura Kallmeyer. 2016. On the mild context-sensitivity of k-Tree Wrapping Grammar. In Annie Foret, Glyn Morrill, Reinhard Muskens, Rainer Osswald, and Sylvain Pogodalla, editors, *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*, number 9804 in Lecture Notes in Computer Science, pages 77–93, Berlin. Springer.

Rainer Osswald and Laura Kallmeyer. 2018. Towards a formalization of Role and Reference Grammar. In Rolf Kailuweit, Lisann Künkel, and Eva Staudinger, editors, *Applying and Expanding Role and Reference Grammar.*, pages 355–378. Albert-Ludwigs-Universität, Universitätsbibliothek. [NIHIN studies], Freiburg.

Robert D. Van Valin, Jr. 2005. *Exploring the syntax-semantics interface*. Cambridge University Press.

Robert D. Van Valin, Jr. 2010. Role and Reference Grammar as a framework for linguistic analysis. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, pages 703–738. Oxford University Press, Oxford.

Jakub Waszczuk. 2017. *Leveraging MWEs in practical TAG parsing: towards the best of the two worlds*. Ph.D. thesis.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, pages 398–403.