6

Treebanks and Mild Context-Sensitivity

WOLFGANG MAIER AND ANDERS SØGAARD †

Abstract

Some treebanks, such as German TIGER/NeGra, represent discontinuous elements directly, i.e. trees contain crossing edges, but the context-free grammars that are extracted from them, fail to make any use of this information. In this paper, we present a method for extracting mildly context-sensitive grammars, i.e. simple range concatenation grammars (RCGs), from such treebanks. A measure for the degree of a treebank's mild context-sensitivity is presented and compared to similar measures used in non-projective dependency parsing. Our work is also compared to discontinuous phrase structure grammar (DPSG).

Keywords TREEBANKS, ANNOTATION, DISCONTINUOUS CONSTITUENTS, MILD CONTEXT-SENSITIVITY

6.1 Introduction

Discontinuous constituents (Huck and Ojeda, 1987) are common across natural languages, and they occur particularly frequently in languages with relatively free word order, such as German. In the following example, the discontinuity is caused by topicalization:

(1) Drei Papiere will ich heute noch schreiben
three papers want I today still write
'I still want to write three papers today.'

However, discontinuous constituents are also found in languages with relatively fixed word order, such as Chinese:

†Thanks to Laura Kallmeyer and the three anonymous reviewers for helpful comments and suggestions.

FG-2008.
Philippe de Groote (Ed.).

(2) shu_1, wo zhi mai pian-yi-de t_1.
book_1, I only buy cheap t_1.
'As for books, I only buy cheap ones.'

The constituent annotation schemata used in treebanks typically include some mechanism for treating discontinuous constituents. One of the most common ways is simply to use special labels. Consider, for instance, the following case of right node raising in the Penn Treebank (Marcus et al., 1994):

```
(S But
   (NP-SBJ-2 our outlook)
   (VP  (VP has
            (VP been
                (ADJP *RNR*-1)))
        ,
        and
        (VP continues
            (S (NP-SBJ *-2)
               (VP to
                   (VP be
                       (ADJP *RNR*-1)))))
        ,
        (ADJP-1 defensive)))
```

FIGURE 1 A PTB tree

A reference is established between the raised constituent and its original sites by the special label `*RNR*` and by the coindexation (`-1`). The German Tübingen Treebank of Written German (TüBa-D/Z) (Telljohann et al., 2006), as another example, uses edge labels to establish the reference between parts of discontinuous constituents. This mechanism is supported by an additional level of topological field annotation. Figure 2 shows the annotation of (3).

(3) Schillen wies dies gestern zurück
Schillen rejected that yesterday VPART
'Schillen rejected that yesterday.'

Here, the edge label `V-MOD` on the adverb (*gestern*) establishes a link to its referent, the verb *wies*. Similar conventions are found in the Spanish 3LB treebank (Civit and Martí Antònín, 2002), for example. The German NeGra (Skut et al., 1997) and TIGER (Brants et al., 2002) treebanks take a different approach. Both depart from annotation backbones based on context-free grammar and represent discontinuous constituents directly. Figure 3 shows the NeGra annotation for (4).

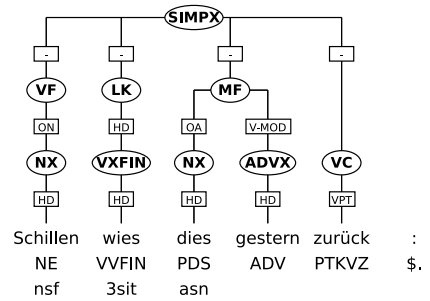

FIGURE 2 A TüBa-D/Z tree

(4) Darüber muß nachgedacht werden
Thereof must thought be
'Thereof must be thought.'

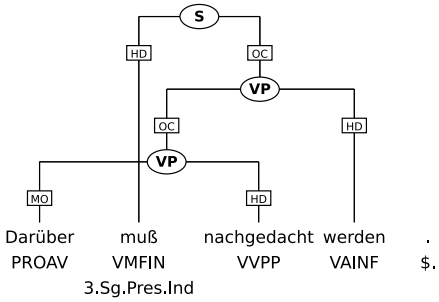

FIGURE 3 A NeGra tree

The verb phrase *darüber nachgedacht* in (4) is a discontinuous constituent. The discontinuity is represented in NeGra by crossing edges. TIGER also uses crossing edges to represent discontinuities.

Most grammars extracted from TIGER/NeGra, if not all, have nevertheless been context-free. Conversions into context-free representations, however, introduce inconsistencies (Kübler et al., 2006, Boyd, 2007). Müller (2004) also shows in an experiment with two head-driven phrase structure grammars (HPSGs) for German that, in addition, the grammar that did not use discontinuous constituents led to around twice as many passive edges in parsing.

Simple RCGs are equivalent to linear context-free rewriting systems (LCFRSs) (Weir, 1988), as also shown in Boullier (1998). The derivation

structures of simple range concatenation grammars (RCGs) (Boullier, 1998) can be used as a unified approach to formally describe trees with discontinuous elements. So can the derivation structures of other similar grammar formalisms (see Sect. 6.4). Our choice of formalism is mainly motivated by the authors' recent work on using RCGs for parsing multicomponent tree-adjoining grammars (Lichte, 2007, Kallmeyer et al., 2008) and on using RCGs in syntax-based machine translation (Søgaard, 2008). The extraction of such derivation structures from treebanks is also a first step toward probabilistic RCGs.

In Sect. 6.2, RCGs and their derivation structures are introduced in some detail. Sect. 6.3 shows how to interpret TIGER/NeGra trees as RCG derivations, and how to extract the underlying simple RCGs. This enables the treebanks to serve as resources for probabilistic RCG parsing. A measure of mild context-sensitivity is then defined and compared to a similar notion from non-projective dependency parsing. In Sect. 6.4, ties to other grammar formalisms are discussed. Future work is also outlined.

6.2 Range concatenation grammars

In RCGs (Boullier, 1998), predicates can be negated (for complementation). If RCGs contain no negated predicates they are called *positive* RCGs. Since simple RCGs are included in the positive RCGs, negated predicates are ignored in the following.

Definition 1 [Positive RCGs] A positive RCG is a 5-tuple $G = \langle N, T, V, P, S \rangle$. N is a finite set of predicate names with an arity function ρ: $N \rightarrow \mathbb{N}^*$, T and V are finite sets of terminal and non-terminal symbols. P is a finite set of clauses of the form

$$\psi_0 \rightarrow \psi_1 \ldots \psi_m$$

where and each of the $\psi_i, 0 \leq i \leq m$, is a predicate of the form $A(\alpha_1, \ldots, \alpha_{\rho(A)})$. Each $\alpha_j \in (T \cup V)^*, 1 \leq j \leq \rho(A)$, is an argument. $S \in N$ is the start predicate name with $\rho(S) = 1$.

Note that the order of RHS predicates in a clause is of no importance. Two subclasses of RCGs are introduced for further reference:

- An RCG $G = \langle N, T, V, P, S \rangle$ is *simple* iff for all $c \in P$, it holds that no variable X occurs more than once in the LHS of c, and if X occurs in the LHS then it occurs exactly once in the RHS, and each argument in the RHS of c contains exactly one variable.
- An RCG $G = \langle N, T, V, P, S \rangle$ is a *k-RCG* iff for all $A \in N, \rho(A) \leq k$.

The language of RCGs is based on the notion of *range*. For a string $w_1 \ldots w_n$ a range is a pair of indices $\langle i, j \rangle$ with $0 \leq i \leq j \leq n$, i.e. a string span, which denotes a substring $w_{i+1} \ldots w_j$ in the source string or a substring

$v_{i+1} \ldots v_j$ in the target string. Only consequtive ranges can be concatenated into new ranges. Terminals, variables and arguments in a clause are bound to ranges by a substitution mechanism. An *instantiated* clause is a clause in which variables and arguments are consistently replaced by ranges; its components are *instantiated predicates*. For example $A(\langle g \ldots h \rangle) \rightarrow B(\langle g+1 \ldots h \rangle)$ is an instantiation of the clause $A(aX_1) \rightarrow B(X_1)$ if the target string is such that $w_{g+1} = a$. A *derive* relation $\Longrightarrow$ is defined on strings of instantiated predicates. If an instantiated predicate is the LHS of some instantiated clause, it can be replaced by the RHS of that instantiated clause. The language of an RCG $G = \langle N, T, V, P, S \rangle$ is the set $L(G) = \{w_1 \ldots w_n \mid S(\langle 0, n \rangle) \stackrel{*}{\Longrightarrow} \varepsilon\}$, i.e. an input string $w_1 \ldots w_n$ is recognized if and only if the empty string can be derived from $S(\langle 0, n \rangle)$.

Example 1 Let $G = \langle \{S, A\}, \{a, b\}, \{X, Y\}, P, S \rangle$ be a simple 2-RCG with $P = \{S(XY) \rightarrow A(X,Y), A(aX, aY) \rightarrow A(X,Y), A(bX, bY) \rightarrow A(X,Y), A(\varepsilon, \varepsilon) \rightarrow \varepsilon\}$. It is easy to see that $L(G) = \{ww \mid w \in \{a,b\}^*\}$ (the copy language). Consider, for instance, a derivation of the string *abab* in G:

$$\begin{array}{ll} & S(\langle 0,4 \rangle) \\ \Longrightarrow & A(\langle 0,2 \rangle, \langle 2,4 \rangle) \\ \Longrightarrow & A(\langle 1,2 \rangle, \langle 3,4 \rangle) \\ \Longrightarrow & A(\langle \varepsilon \rangle, \langle \varepsilon \rangle) \\ \Longrightarrow & \varepsilon \end{array}$$

6.2.1 RCG derivation structures

All possible parses of a string w with respect to some RCG G can be resprepresented as a context-free grammar G_D (Bertsch and Nederhof, 2001). Intuitively, this is achieved by introducing a context-free production for each possible instantiation of every clause in G with ranges of w, interpreting the instantiated predicates as non-terminal symbols of the resulting CFG. It allows for a packed representation of all parses, i.e. a shared forest or AND-OR graph (Billot and Lang, 1989). The derivation in Example 1 is, for instance, represented as:

$$\begin{array}{c} S(\langle 0,4 \rangle) \\ | \\ A(\langle 0,2 \rangle, \langle 2,4 \rangle) \\ | \\ A(\langle 1,2 \rangle, \langle 3,4 \rangle) \\ | \\ A(\langle \varepsilon \rangle, \langle \varepsilon \rangle) \\ | \\ \varepsilon \end{array}$$

6.3 RCG derivation structure treebanks

6.3.1 Reconstructing clauses from derivation structures

The trees of both NeGra and TIGER can be interpreted as RCG derivations; in other words, these treebanks can be considered a resource for estimating probabilistic RCGs for German. Since the estimated RCGs are guaranteed to be simple, standard estimation procedures can be adopted, e.g. Kato et al. (2006) (Sect. 6.4.2). A method is presented below for extracting RCGs from treebanks with crossing edges.

Our goal is thus to interpret the treebank trees as RCG derivations. In order to do that in a meaningful way, we first have to identify the clauses the parse tree is composed of. To achieve that, different arguments of RCG clauses have to be identified. For each tree over some sentence $w_1 \dots w_n$, we extract a set of clauses. All clauses are counted and collected into a single grammar. The clauses for a single tree are extracted as follows. For each nonterminal node N with n daughters $N'_1, \dots N'_m$, where N is not a preterminal, introduce a clause with an LHS predicate named N and m RHS predicates named N'_1 to N'_m. For each w_i, $1 \leq i \leq n$, we introduce a variable X_i. Then for the predicate N, the following conditions must hold:

- The arguments of N must contain no terminals,
- the concatenation of all variables in all arguments of N must be the concatenation of all $X \in \{X_i \mid N \text{ dominates } w_i\}$ such that X_i precedes X_j if $i < j$,
- a variable X_i with $1 \leq i < n$, is the right boundary of an argument of the predicate N iff $X_{i+1} \notin \{X_i \mid N \text{ dominates } w_i\}$, i.e., an argument boundary is introduced at each discontinuity.

The arguments of the RHS predicates are determined in the same way. Range variables which are adjacent on the LHS and the RHS are collapsed into single variables, which assures that the RHS predicates of the resulting clauses have a single variable per argument. For each preterminal node N dominating some terminal node w_i, we introduce a clause $N(w_i) \rightarrow \varepsilon$, called a *lexical clause*. This procedure yields the following set of clauses for the tree in Figure 3:

$$
\begin{aligned}
\text{PROAV(Darüber)} &\rightarrow \varepsilon \\
\text{VMFIN(muß)} &\rightarrow \varepsilon \\
\text{VVPP(nachgedacht)} &\rightarrow \varepsilon \\
\text{VAINF(werden)} &\rightarrow \varepsilon \\
\text{S}(X_1X_2X_3) &\rightarrow \text{VP}(X_1,X_3)\ \text{VMFIN}(X_2) \\
\text{VP}(X_1,X_2X_3) &\rightarrow \text{VP}(X_1,X_2)\ \text{VAINF}(X_3) \\
\text{VP}(X_1,X_2) &\rightarrow \text{PROAV}(X_1)\ \text{VVPP}(X_2)
\end{aligned}
$$

We can now reconstruct the the RCG derivation, as in Figure 4. Note that

$\langle 0,1\rangle$ is *Darüber*, $\langle 1,2\rangle$ is *muß*, $\langle 2,3\rangle$ is *nachgedacht* and $\langle 3,4\rangle$ is *werden*.

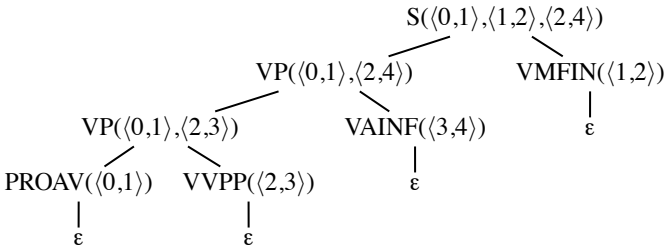

FIGURE 4 Interpretation of a NeGra tree as RCG derivation

It is easy to see that the RCGs extracted this way are all *simple* RCGs. This is a result of the fact that no string range can be part of more than one constituent in TIGER/NeGra. What differentiates the TIGER/NeGra annotation from a context-free annotation is merely the possibility to group all parts of discontinuous constituents under the same node disregarding the possible intervening material.

6.3.2 Extracting range concatenation grammars

The algorithm described above was applied to TIGER/NeGra to extract simple RCGs. The dimensions of the treebanks are shown in table 1.

	sent	cross	av slen	av nt
NeGra	20602	5853 (28.40%)	17.24	6.07
TIGER	50474	14114 (27.96%)	17.60	6.44

TABLE 1 Properties of NeGra and TIGER

TIGER is roughly 2.5 times as big as NeGra. The ratio of sentences with crossing edges (*cross*), the average sentence length (*av slen*) and the average number of nonterminals per sentence (*av nt*) are all comparable, however, which confirms a consistent application of the annotation guidelines. Figure 2 shows some dimensions of the extracted grammars G_N, i.e. the simple RCG extracted from NeGra, and G_T, i.e. the simple RCG extracted from TIGER.

The most frequent clauses in the extracted grammars that involve discontinuities are listed below, i.e. the most frequent clauses with predicates of arity ≥ 2. Note how similar the lists are: the first most frequent clauses in the two grammars are identical; the second most frequent clause in G_N is the third most frequent clause in G_T; the third most frequent clause in G_N is the second most frequent clause in G_T; 4–6 are again identical; and finally, 7–9 in

	G_N (NeGra)	G_T (TIGER)
total # of clauses	468,607	1,192,807
total # of different clauses	71,868	127,154
lexical clauses	52,747	92,731
non-lexical clauses	19,121	34,423

TABLE 2 Dimensions of extracted RCGs

G_N are identical to, resp., 8, 10 and 7 in G_T. Only 10 in G_N is not in the top ten list in G_T, and 9 in G_T is not in the top ten list in G_N.

1	733	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VAFIN(X_2)NP(X_3)$
2	271	$VP(X_1,X_2) \rightarrow PP(X_1)VVPP(X_2)$
3	268	$S(X_1X_2X_3X_4X_5) \rightarrow VP(X_1,X_3,X_5)VAFIN(X_2)NP(X_4)$
4	236	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VAFIN(X_2)PPER(X_3)$
5	193	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_3)NP(X_2)VAFIN(X_4)$
6	149	$NP(X_1X_2,X_3) \rightarrow ART(X_1)NN(X_2)S(X_3)$
7	148	$NP(X_1,X_2) \rightarrow PPER(X_1)VP(X_2)$
8	142	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VMFIN(X_2)NP(X_3)$
9	130	$VP(X_1,X_2X_3) \rightarrow VP(X_1,X_2)VAINF(X_3)$
10	127	$NP(X_1,X_2) \rightarrow PPER(X_1)S(X_2)$

TABLE 3 Most frequent clauses with predicates of arity ≥ 2 in G_N (NeGra)

1	1996	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VAFIN(X_2)NP(X_3)$
2	790	$S(X_1X_2X_3X_4X_5) \rightarrow VP(X_1,X_3,X_5)VAFIN(X_2)NP(X_4)$
3	645	$VP(X_1,X_2) \rightarrow PP(X_1)VVPP(X_2)$
4	526	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VAFIN(X_2)PPER(X_3)$
5	454	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_3)NP(X_2)VAFIN(X_4)$
6	401	$NP(X_1X_2,X_3) \rightarrow ART(X_1)NN(X_2)S(X_3)$
7	378	$VP(X_1,X_2X_3) \rightarrow VP(X_1,X_2)VAINF(X_3)$
8	364	$NP(X_1,X_2) \rightarrow PPER(X_1)VP(X_2)$
9	351	$PP(X_1,X_2) \rightarrow PROAV(X_1)S(X_2)$
10	325	$S(X_1X_2X_3X_4) \rightarrow VP(X_1,X_4)VMFIN(X_2)NP(X_3)$

TABLE 4 Most frequent clauses with predicates of arity ≥ 2 in G_T (TIGER)

6.3.3 Degree of mild context-sensitivity

The extracted RCGs give an intuitive picture of the constituents contained in the treebank trees: It is easy to see which subtree corresponds to a clause such as VP(X_1,X_2X_3) $\rightarrow$ VP(X_1,X_2) VAINF(X_3), and especially that two VPs with one interruption per yield each are involved. How can we classify the degree

of discontinuity (context-sensitivity respectively) of our RCGs in a precise way?

Simple RCGs are, as already said, equivalent to LCFRSs. It follows that the languages $L(G_T)$ and $L(G_N)$ of the extracted grammars G_T and G_N are mildly context-sensitive. In order to make a more fine-grained statement about the degree of mild context-sensitivity of the treebanks, the notion of *gap degree* used in non-projective dependency parsing (Nivre, 2006, Kuhlmann and Nivre, 2006) is useful. In short, the gap degree of a dependency graph corresponds to the maximal number of interruptions in the projection of a node.

Definition 2 [Dependency graph] D is a *dependency graph* for some sentence $s = w_1, \ldots, w_n$ iff $D = \langle V, E, L \rangle$ is a labeled directed graph with $V = \{0, \ldots, n\}$ with a bijection $f : \{w_1, \ldots, w_n\} \rightarrow V \setminus \{0\}$ such that $f(w_i) = i$, $E : V \times V \setminus \{0\}$, and $L : E \rightarrow R$ is a labeling function from the set of edges to some set R of dependency types. We introduce the relation $\rightarrow$ (*dominates*). We write $i \rightarrow j$ if $(i, j) \in E$. $\overset{*}{\rightarrow}$ is the reflexive transitive closure of $\rightarrow$. The set of nodes dominated by i is called the *yield* of i. We use π_i to refer to the yield of i, arranged in ascending order. π_i is called the *projection* of i. A dependency graph D is well-formed iff it is acyclic and connected, and the in-degree of all vertices is at most 1.

Definition 3 [Gap degree] Let $D = \langle V, E, L \rangle$ be a dependency graph. Let π_i be the projection of some node $i \in V$.

1. For some $i \in V$, a *gap* is a pair (j_k, j_{k+1}) of nodes adjacent in π_i such that $j_{k+1} - j_k > 1$, i.e., a gap is a discontinuity in the projection of a node. The *gap degree* d of a node i in a dependency graph is the number of of gaps in π_i,
2. The gap degree d of a dependency graph D is the maximal gap degree of any of its nodes.

A dependency graph D is called *projective* if its gap degree is 0.

	PDT	DDT
$d = 0$	76.85%	84.95%
$d = 1$	22.72%	14.89%
$d \geq 2$	0.43%	0.16%

TABLE 5 Gap degree of graphs in two dependency treebanks

Table 5 (from Kuhlmann and Nivre (2006)) shows the gap degree figures of the Prague Dependency Treebank (PDT) and the Danish Dependency Treebank (DDT). How can we transfer the notion of projectivity to constituent

structures?

Our extracted RCGs gives us easy access to the sentences in which terminal sequences dominated by some nonterminal node N are interrupted by material not dominated by N. Whenever there is discontinuous constituency, predicates with multiple arguments are extracted. Moreover, the number of arguments in the predicates in question reflect the minimum number of connected subtrees that span the intervening substring.

Call the number of arguments of a predicate in the RHS of a clause generated for some nonterminal node N in a treebank tree minus one the *constituent gap degree c* of the clause. The constituent gap degree of a treebank tree is the maximal constituent gap degree of one its extracted clauses. Table 6 shows the constituent gap degree figures for NeGra and TIGER.

	NeGra	TIGER
$c = 0$	14,924 (72,44%)	36,573 (72,46%)
$c = 1$	4,991 (24,23%)	12,302 (24,37%)
$c = 2$	679 (3,30%)	1,585 (3,14%)
$c = 3$	8 (0,04%)	14 (0,03%)

TABLE 6 Constituent gap degree of TIGER/NeGra trees

If it is assumed that the linguistic phenomena that give rise to non-projective in dependency structures in dependency treebanks are similar to the phenomena that are described in terms of discontinuous constituents in constituent-based treebanks, the figures for all four treebanks can be compared. The fact that the figures for NeGra and TIGER are closer to each other than to the gap degree figures for the two dependency treebanks may be due to differences between annotation guidelines or it may reflect structucal differences between the languages in question. The difference between the gap degree figures in the two dependency treebanks suggests the latter. This hypothesis remains to be confirmed by an analysis of dependency versions of the TIGER/NeGra treebanks (Daum et al., 2004).

If the two measures are assumed to reflect exactly the same linguistic phenomena, our results indicate that discontinuous constituents are more frequent (*modulo* text types) in German than in Czech or Danish, and more frequent in Czech than in Danish. This result is consistent with the literature, e.g. Kübler et al. (2006). The difference between the frequency of discontinuous constituents in languages like Danish and German can also be shown by the ratio of translation units and discontinuous translation units in hand-aligned parallel corpora, e.g. in the Danish–English parallel corpus used in Buch-Kromann (2007) and the English–German parallel corpus used in Padó and Lapata (2006). It should be noted that these two parallel corpora differ

considerably in size, i.e. the Danish–English parallel corpus contains 4,729 sentences, whereas the English–German one only contains 650 sentences. It should also be noted that a discontinuous translation unit need not be a discontinuous constituent, and a discontinuous constituent need not always be treated as a discontinuous translation unit in parallel corpora, e.g. if it translates into a structurally similar translation unit. Nevertheless, the numbers in Table 7 are comparable to our results. The results are from the two parallel corpora just mentioned.

	TUs/DTUs
Danish–English	1.63%
English–German	7,36%

TABLE 7 Ratio of translation units (TUs) and discontinuous translation units (DTUs) in two parallel corpora

6.4 Related work

6.4.1 Discontinuous phrase structure grammar

Discontinuous phrase structure grammar (DPSG) warrants a separate comparison, since it is explicitly motivated by discontinuous constituency and has been used in practical applications. DPSG was introduced in Bunt et al. (1987) as an extension of context-free grammar that enables direct representation of discontinuous elements. Plaehn (1999, 2004) presents applications to treebank-based parsing.

The notion of discontinuous trees or *discotrees* is central to DPSG. See Figure 5 for an example of a discotree (a) and two of its subtrees (b) and (c).

Essentially, DPSG represents discontinuity in some subtree T rooted at some node r by specifying material not dominated by r alongside with rs daughters. The DPSG productions that correspond to (b) and (c) are $P \rightarrow a[b]c$ and $Q \rightarrow b[c]d$. DPSG rules extracted from a tree without discontinuous elements are simply context-free rules.

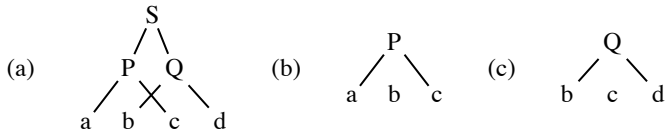

FIGURE 5 Discotree (a) and two of its subtrees (b), (c)

Certain DPSG productions seem somewhat unintuitive from a linguistic

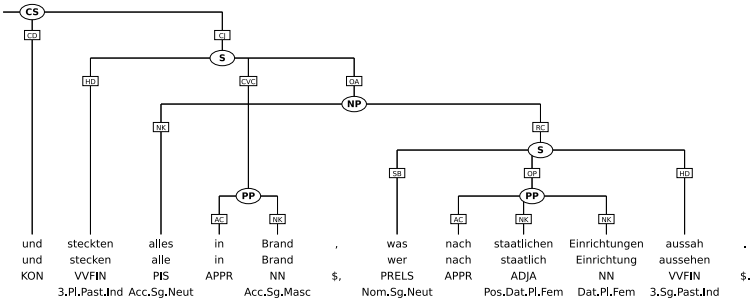

FIGURE 6 An extracted relative clause in TIGER

point of view. In the case of extracted relative clauses, for instance, the relative clause is not in any way influenced by the material that may occur between itself and the modified noun. Nevertheless a DPSG rule would in this case include all intervening material. A corresponding RCG clause though would simply separate the noun and the relative clause into two different arguments, allowing for intervening material, but not specifying it. Consider, for instance, the NeGra annotation for (5) in Figure 6.

(5) …und steckten alles in Brand,
…and set everything alight
was nach staatlichen Einrichtungen aussah.
what after state facilities looked
'…and set everything alight what looked like state facilities.'

A DPSG rule describing the relative clause would be NP → PIS [PP] S. An RCG clause describing the same datum would simply be NP(X,Y) → PIS(X)S(Y), with the immediate advantages that (i) already from the LHS of the clause, we know that we are dealing with a constituent with a single discontinuity and that (ii) in the RHS, we do not have to specify exactly what it is that separates the relative clause from its dependent.

Note also that RCG has better worst-case complexity than DPSG. The complexity of the parsing algorithm in Plaehn (2004) is exponential, while there exists polynomial time parsing algorithms for RCG (Boullier, 2000, Villemonte de la Clergerie, 2002, Parmentier et al., 2008).

6.4.2 Linear context-free rewriting systems and multiple context-free grammars

Simple RCGs and LCFRSs are also equivalent to multiple context-free grammars (MCFGs) (Seki et al., 1991). For all three theories available parsers exist.

LCFRS	Burden and Ljunglöf (2005)
MCFG	Kato et al. (2006), Kanazawa (2008)
RCG	Parmentier et al. (2008)

Kato et al. (2006) even present algorithms for parsing and estimation of probabilistic MCFGs.

The three theories are merely notational variants if MCFGs are assumed to be non-erasing, i.e. a variable of a function f must be used exactly once in the RHS of f. It is thus possible to compare these parsers, something left for future work for now.

6.5 Conclusion

In this paper motivation has been provided for the interpretation of two treebanks using annotation schemata with crossing edges, namely TIGER and NeGra, as collections of simple RCG derivation structures. Such interpretations also give us ready-to-use resources for extraction of probabilistic RCGs. The degree of mild context-sensitivity of the RCGs extracted from the two treebanks was measured, and it was shown that our results are comparable to related results from dependency treebanks. Nivre (2006) remarks that not much work has been done on parsing discontinuous structures directly. Our current research involves using the extracted simple RCGs for probabilistic parsing.

Estimation and probabilistic parsing of simple RCGs is relatively simple if the spans in the complex labels are ignored and can be done with the techniques for MCFGs described in Kato et al. (2006). Estimation and probabilistic parsing of positive RCGs *in general*, however, is more complicated because of the copying of substrings that occurs when there are multiple occurences of the same variable in a clause's RHS. What is needed, it seems, is to unravel the underlying simple derivation structures and estimate the probabilities of the unravelled trees seperately. If the propability of a derivation structure in which substrings are copied n times is said to be p_0 with $p_0^n = p_1 \times \ldots \times p_n$, tightness follows immediately.

References

Bertsch, Eberhard and Mark-Jan Nederhof. 2001. On the complexity of some extensions of RCG parsing. In *Proceedings of the 7th International Workshop on Parsing Technologies*, pages 66–77. Beijing, China.

Billot, Sylvie and Bernard Lang. 1989. The structure of shared forests in ambiguous parsing. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 143–151. Vancouver, Canada.

Boullier, Pierre. 1998. Proposal for a natural language processing syntactic backbone. Rapport de Recherche RR-3342, Institut National de Recherche en Informatique et en Automatique, Le Chesnay, France.

Boullier, Pierre. 2000. Range concatenation grammars. In *Proceedings of the 6th International Workshop on Parsing Technologies*, pages 53–64. Trento, Italy.

Boyd, Adriane. 2007. Discontinuity revisited: An improved conversion to context-free representations. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, the Linguistic Annotation Workshop*, pages 41–44. Prague, Czech Republic.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, pages 24–42. Sozopol, Bulgaria.

Buch-Kromann, Matthias. 2007. Computing translation units and quantifying parallelism in parallel dependency treebanks. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, the Linguistic Annotation Workshop*, pages 69–76.

Bunt, Harry, Jan Thesingh, and Ko van der Sloot. 1987. Discontinuous constituents in trees, rules and parsing. In *Third Conference of the European Chapter of the Association of Computational Linguistics*, pages 203–210. Copenhagen, Denmark.

Burden, Håkan and Peter Ljunglöf. 2005. Parsing linear context-free rewriting systems. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 11–17. Vancouver, British Columbia.

Civit, Montserrat and M. Antònia Martí Antònín. 2002. Design principles for a Spanish treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*. Sozopol, Bulgaria.

Daum, Michael, Kilian Foth, and Wolfgang Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. Lisbon, Portugal.

Huck, Geoffrey and Almerindo Ojeda, eds. 1987. *Discontinuous constituency*. New York, New York: Academic Press.

Kallmeyer, Laura, Timm Lichte, Wolfgang Maier, Yannick Parmentier, and Johannes Dellert. 2008. Developing an MCTAG for German with an RCG-based parser. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. Marrakech, Morocco. To appear.

Kanazawa, Makoto. 2008. A prefix-correct earley recognizer for multiple context-free grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, pages 49–56. Tübingen, Germany.

Kato, Yuki, Hiroyuki Seki, and Tadao Kasami. 2006. RNA pseudoknotted structure prediction using stochastic multiple context-free grammar. *IPSJ Digital Courier* 2:655–664.

Kübler, Sandra, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 111–119. Sydney, Australia.

Kuhlmann, Marco and Joakim Nivre. 2006. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514. Sydney, Australia: Association for Computational Linguistics.

Lichte, Timm. 2007. An MCTAG with tuples for coherent constructions in German. In *Proceedings of the 12th Conference on Formal Grammar*. Dublin, Ireland.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

Müller, Stefan. 2004. Continuous or discontinuous constituents? *Research on Language & Computation* 2(2):209–257.

Nivre, Joakim. 2006. Constraints on non-projective dependency parsing. In *11th Annual Meeting of the European Chapter of the Association for Computational Linguistics*, pages 73–80. Trento, Italy.

Padó, Sebastian and Mirella Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of the 21st International Conference on Computational Lingustics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 1161–1168.

Parmentier, Yannick, Laura Kallmeyer, Wolfgang Maier, Timm Lichte, and Johannes Dellert. 2008. TuLiPA: A syntax-semantics parsing environment for mildly context-sensitive formalisms. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*. Tübingen, Germany.

Plaehn, Oliver. 1999. *Probabilistic parsing with discontinuous phrase structure grammar*. Diploma thesis, Dpt. of Computational Linguistics, Saarland University, Saarbrücken, Germany.

Plaehn, Oliver. 2004. Computing the most probable parse for a discontinuous phrase-structure grammar. In H. Bunt, J. Carroll, and G. Satta, eds., *New Developments in Parsing Technology*, pages 91–106. Kluwer Academic Publishers.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88:191–229.

Skut, Wojciech, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the 5th Applied Natural Language Processing Conference*, pages 88–95. Washington, District of Columbia.

Søgaard, Anders. 2008. Range concatenation grammars for translation. In *Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, England. To appear.

Telljohann, Heike, Erhard Hinrichs, Sandra Kübler, and Heike Zinsmeister. 2006. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technischer Bericht, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen. Revidierte Fassung.

Villemonte de la Clergerie, Eric. 2002. Parsing mildly context-sensitive languages with thread automata. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7. Taipei, Taiwan.

Weir, David J. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.