Querying topological fields in the TIGER scheme with TIGERSearch

Stefanie Dipper

Institute of Linguistics
Ruhr-University Bochum
E-mail: dipper@linguistics.rub.de

Abstract

The TIGER corpus is a German treebank with a hybrid dependency-constituency annotation. In this paper, I address the question how well topological fields (e.g. Vorfeld, Verb second) can be searched in this treebank, using the search tool TIGERSearch. For most queries, a version without crossing branches is used. It turns out that queries can be formulated that result in quite good F-scores for the Vorfeld and left and right brackets. Mittelfeld and Nachfeld are hard to query. This is due partly to properties of the language, partly to design decisions in the TIGER scheme, and partly to restrictions imposed by the search tool.

1 Introduction

This paper deals with the TIGER scheme [1], one of the two main annotation schemes for German syntax. The other main scheme is the TüBa-D/Z scheme [18]. The two schemes implement quite different design principles. The TIGER scheme is famous for its extensive use of *crossing branches* for encoding non-local dependencies. The TüBa-D/Z scheme is special in that it puts a layer with *topological fields* on top of the constituency structure.[1]

Topological fields are widely acknowledged as a useful concept by modern syntactic theories of German. Hence, linguists using German treebanks often would like to refer to these notions in formulating a query expression. The TIGER corpus [5] was created to serve both as training data for automatic applications and as a source for linguistic investigations. The question, addressed in this paper, is then whether linguist users are able to query topological fields not only in the TüBa-D/Z treebank, where they are explicitly encoded, but also in the TIGER treebank. The TIGER corpus comes with its own search tool, TIGERSearch [10], which is also used in this paper for searching the corpus.

[1] For a comparison of the two schemes, see [6].

VF	LK	MF			RK	NF		
Hans	*hat*	*heute*	*Maria*		*getroffen*	*die*	*einkaufen*	*war*
H.	has	today	M.		met	who	shopping	was
Hans	*traf*	*heute*	*Maria*			*die*	*einkaufen*	*war*
H.	met	today	M.			who	shopping	was
	dass	*Hans*	*heute*	*Maria*	*getroffen hat*	*die*	*einkaufen*	*war*
	that	H.	today	M.	met has	who	shopping	was

Figure 1: Topological field analysis of different sentences (‘(that) Hans met Maria today, who was shopping’)

The paper first gives a short introduction to German syntax (Sec. 2). Sec. 3 introduces the TIGER annotation scheme, and Sec. 4 presents the evaluation, followed by the conclusion (Sec. 5). The appendix contains sample templates.

2 German syntax: topological fields

German has a relatively free constituent order. Following a long tradition, German sentences are usually analyzed and split into different *topological fields* [9]. The element that functions as the separator between these fields is the verb or verbal parts (in most cases). The verb can be located in two different positions, either the second (“verb second”) or the final position (“verb final”) of the clause. Fig. 1 shows three sentences with their field analyses. “LK” and “RK” (“Linke/Rechte Klammer”, ‘left/right bracket’) indicate the two verbal positions (LK can also be occupied by subordinating conjunctions). The brackets divide the sentences into “VF” (“Vorfeld”, ‘prefield’), containing exactly one constituent, “MF” (“Mittelfeld”, ‘middle field’) with multiple constituents, and “NF” (“Nachfeld”, ‘postfield’), which often contains clausal constituents (which can be assigned a separate layer with topological fields). The brackets and the fields can also stay empty.

If the sentence contains only a simple verb form, one of the brackets remains empty, possibly resulting in ambiguous structures, see Fig. 2: (ia/b) and (iia/b) contain identical strings each, which can be analyzed by different brackets and fields, though. To (manually) disambiguate such structures, the simple verb form is replaced by some complex verb form, e.g. a particle verb or a combination of an auxiliary or modal plus verb. In (i’) the simple verb *ging* ‘went’ has been replaced by the particle verb *ging weg/wegging* ‘went away’; in (ii’) the simple form of the preterite *traf* ‘met’ has been replaced by perfect tense *hat getroffen* ‘has met’. The test paraphrases in (i’) reveal that (i) can be a verb-second (a) or verb-final (b) clause. The two options in (ii’) are stylistic variants, and it is sometimes hard to tell which is “the right” one. The TüBa-D/Z scheme defines a default rule for such cases [18, p. 93]: *Unless there is strong evidence for a position in MF, the relative clause is located in NF.* In the TIGER scheme, which does not annotate topological fields, the variants result in the same analysis.

The different topological slots — fields and brackets — are highly relevant

	VF	LK	MF	RK	NF
(i) a	***wer***	***ging***			
(i) b	***wer***			***ging***	
	who	went		went	
(i') a	*wer*	*ging*			*weg*
	who	went			away
(i') b	*wer*			*wegging*	
	who			away-went	
(ii) a	*Hans*	*traf*	***Leute, die*** ...		
(ii) b	*Hans*	*traf*	***Leute,***		***die*** ...
	H.	met	people who		who
(ii') a	*Hans*	*hat*	*Leute, die* ...	*getroffen*	
(ii') b	*Hans*	*hat*	*Leute*	*getroffen,*	*die* ...
	H.	has	people who	met	who

Figure 2: (Fragments of) syntactically-ambiguous (i/ii) and non-ambiguous (i'/ii') sentences ('who went (away)?'; 'Hans met people who ...')

for research in German syntax. E.g. the Vorfeld often serves as a test position for constituency because it usually contains exactly one constituent — there are exceptions, though (see e.g. [11]). The Vorfeld is also interesting from an information-structural point of view because it seems to be the prime position for sentence topics — it often contains constituents with other information-structural functions, though (e.g. [16, 7]). Constituent order ("scrambling") within the Mittelfeld has been investigated extensively (e.g. [2]), as well as the question which constituents can occur *extraposed*, i.e. in the Nachfeld slot (e.g. [17]). Finally, the relative order of verbal elements in the Rechte Klammer has been researched a lot (e.g. [8]).

3 The TIGER annotation scheme

The TIGER scheme implements a hybrid approach to syntactic structure, combining features from constituency and dependency structures. On the one heand, it uses virtual nodes like "NP" and "VP" for constituents. On the other hand, non-local dependents are connected by crossing branches, directly linking the head and its dependent; edges are labeled by grammatical functions such as "SB" (subject) or "MO" (modifier).

The TIGER scheme omits "redundant" nodes, assuming that these nodes can be recovered automatically by combining information from the POS tags and/or the functional labels. This concerns two types of nodes: *unary* nodes, i.e. non-branching nodes like NP nodes that dominate one terminal node only; and NP nodes dominated by PPs.

This design principle — omitting redundant nodes — poses obvious problems for treebank users. If users are interested e.g. in VPs with an NP daughter, they have

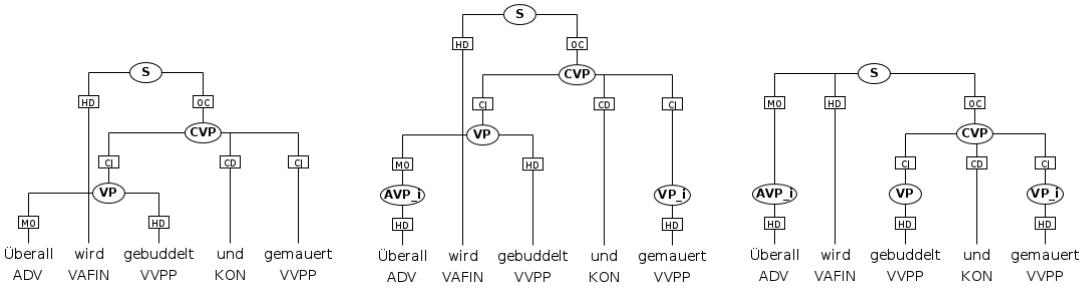

Figure 3: TIGER sentence no. 291 in original version (left), enriched version ("ENR", center), and endriched context-free version ("CF", right) ('Everywhere people are digging and constructing')

to make additional efforts to retrieve all actual NPs. The search tool that comes with the TIGER corpus, TIGERSearch [10], allows the user to define *templates*, which facilitates such queries enormously. (1a) defines a sample template for pronouns. The first conjunct exploits the fact that all pronominal POS tags start with "P" (e.g. "PPER" for personal pronouns, "PPOSAT" for attributive possessive pronouns, etc. [15]). Other tags that start with "P" (pronominal and interrogative adverbs and particles) are excluded by the second conjunct. (1b) shows how to use the template in a query to constrain the otherwise unspecified node variable "#a" to pronouns. The query searches for VPs that directly dominate some pronoun.

(1) a. `PRON(#x) <- #x: [pos=/P.*/ & pos!=/PROAV|PWAV|PTK.*/];`

b. `[cat="VP"] > #a:[] & PRON(#a)`

In a similary way, a template for NPs in general could be defined. An alternative way is to apply a script that expands TIGER's minimalistic structures and inserts such redundant nodes, thus creating an enriched, user-friendly version of the treebank, as has been suggested e.g. by [14]. Fig. 3 illustrates both formats. The figure shows a TIGER structure in the original version (left) and in the enriched version (center), with two inserted nodes: AVP_i and VP_i.[2]

Non-local dependencies are encoded by crossing branches in the TIGER scheme. Such structures are difficult to process automatically, so scripts have been created to re-attach these branches in a way to avoid crossings. I call the resulting structures "context-free" because they could have been created by a context-free grammar. The rightmost structure shown in Fig. 3 is such a context-free structure.[3] It attaches the AVP_i node higher up, eliminating the crossing branch. The evaluation

[2]The enriched version of the corpus has been created by the tool *TIGER Tree Enricher* [13]. The marker "_i" for inserted nodes is optional, and is used here to highlight inserted nodes. All sentence numbers in this paper refer to the TIGER corpus, release 2.2; URL: `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`.

[3]The context-free version of the corpus has been created by the program *treetools* by Wolfgang Maier, URL: `https://github.com/wmaier/treetools`.

Field	Template description
VF	– In main clauses: leftmost constituent of a sentence, preceding a finite verb; coordinating conjunctions may precede – In subord. clauses: leftmost constituent of a sentence, dominating a relative or interrogative word
LK	– In main clauses: the finite verb following VF – In subord. clauses: the subordinating conjunction
MF	The part between LK and RK (i.e. both brackets must be filled); the template marks the beginning (MFB) and end (MFE) of MF
RK	– Single-element RKs (RKS): the finite verb in a subordinate clause, or a verb particle, infinitive or participle – Multi-element RKs: a cluster of several verbs; the template marks the beginning (RKB) and end (RKE) of complex RKs
NF	The part following a verb particle, infinitive, participle or verb cluster; the template only marks the beginning (NFB) of NF

Table 1: Description of the topological templates

will show that querying the topological fields is rather difficult if not impossible if crossing branches may occur.[4]

4 Querying the scheme: an evaluation

The TIGER corpus has been used successfully to search for elements in specific topological fields. For instance, [12] investigates (a subset of) extraposed clauses, i.e. clauses in the Nachfeld. However, [12] only considers clauses that are dependent from a noun (object or relative clauses), which facilitates querying enormously. As we see below, querying for Nachfeld constituents *in general* is actually very hard.

For the evalution, a student of linguistics annotated the first 600 sentences of the TIGER corpus with topological fields.[5] Sentences 1–100 were used in the development of the query templates, sentences 101–600 were reserved for the evaluation.

Of course, the results of the evaluation heavily depend on the quality of the templates. Table 1 summarizes the most relevant properties of the individual templates. In the definitions of the templates, I tried to avoid exploiting linguistic knowledge about the topological fields, such as "sentential constituents often are located in the NF" (because this is a statement that one might want to verify). However, I did use information such as "relative clauses are verb-final clauses".

I evaluate the scheme by applying the templates to the TIGER corpus in an

[4]This observation can be transferred to treebanks annotated with pure dependency structures.

[5]The annotation tool was WebAnno [19]. The fields were annotated mainly according to the TüBa-D/Z scheme. However, the student located interrogative and relative pronouns in VF, and subordinating conjunctions in LK (rather than C).

enriched version with redundant nodes and in context-free format ("CF"). Querying the version with crossing branches ("ENR") results in highly complex (and inefficient) queries, so I defined only the template for VF in ENR.[6]

The appendix displays the template definitions for the Vorfeld position and the precedence relation in the CF version.[7,8] For efficiency reasons, the VF template is split in two parts: VF in main (VFmain) and subordinate (VFsub) clauses. VFmain also covers the verb-second (V2) position in the LK slot, since VF and V2 depend on each other. A query using the VF template is shown in (2).

(2) `#vf:[] & #v2:[] & VFmain_cf(#vf,#v2)`

The templates are designed to result in high precision rather than high recall. For instance, only VF instances are covered where the sentence either directly starts (i) with the VF or (ii) with a coordinating conjunction that directly precedes the VF. Other sentence-initial elements or elements following the VF are not allowed to maintain the constraint that there is exactly one constituent preceding the finite verb. This constraint, e.g., excludes VF in sentences with preposed material (3a) or with parentheticals intervening between VF and the finite verb (3b).[9]

(3) a. [$_{AVP}$ Gewiß] — [$_{NP}$ <u>die wirtschaftliche Liberalisierung und Öffnung des Landes</u>$_{VF}$] schreiten voran . (s62)

b. [$_{CAP}$ <u>Früher oder später</u>$_{VF}$] , [$_{S}$ da sind sich alle einig] , muß Perot Farbe bekennen und Konzepte vorlegen . (s47)

4.1 Qualitative results

Qualitative results from the development process show that there are certain types of constructions that cannot be handled properly by the templates. The problems can be traced back to (i) difficult constructions, (ii) systematic ambiguities of the language, (iii) constraints of the search tool, and (iv) the design of the annotation scheme, in particular (v) crossing branches.

[6] Using the enriched versions facilitates querying since we do not have to care about omitted NP nodes etc. The vast majority of the conversion steps of the enrich-script are trivial so they do not affect the evaluation, cf. [13]. Creating the context-free version involves more complex operations, see Fn. 4. Still, the conversion does not seem to introduce problematic structures.

[7] All template definitions used in this paper can be found at `http://www.linguistics.ruhr-uni-bochum.de/~dipper/tiger-templates.html`.

[8] TIGERSearch uses a purely left corner-based definition of precedence, which is not sufficient in most cases (a node #n1 is said to precede another node #n2 if the left corner of #n1 precedes the left corner of #n2 [10, p. 80]; according to this definition, a node consisting of two or more words does not precede its following sibling). In addition, the precedence template allows for intervening quotes (via the template "prec_quotes"; and similarly with "prec_comma"). The VF template further refers to a template "hasLeftChild", which defines left-corner dominance. This template extends the corresponding TIGER relation to one that holds between terminal or non-terminal nodes.

[9] The parenthetical sentence in (3b) contains a VF, which is correctly found by the VF template. Here and in the following examples, the underlined, labeled part indicates the "target" slot, as annotated in the gold data, and the part in boldface indicates the string matched by the template (if any).

(i) Difficult constructions In general, parentheticals, non-constituent coordination and elliptical constructions are difficult to handle by templates, so a large number of these are not covered. (4) shows instances of coordinated elliptical sentences: In (4a), the VF is missing in the second conjunct so that the verb in second position (LK) cannot be recognized. In (4b), the second conjunct consists of the VF only, and the predicate is missing.

(4) a. Er **tritt**$_{LK}$ in die GM-Verwaltung ein und wird$_{LK}$ Großaktionär des Autokonzerns . (s25)

b. " **Geschäftemachen**$_{VF}$ ist seine Welt und nicht die Politik$_{VF}$. (s44)

(ii) Systematic linguistic ambiguities First, sentences with empty Mittelfeld and simple finite verbs are systematically ambiguous, as shown in Fig. 2.[10] The finite verb in such sentences would be (possibly incorrectly) matched by the VF template. A pertinent example from our development corpus is the verb of the relative clause *die meinen* 'who think' in (5).

(5) Allerdings gibt es dem Magazin zufolge in kleinen und mittleren Firmen viele Unternehmer , die meinen$_{RK}$, Perot sei einer von ihnen , und die den Texaner unterstützen . (s18)

A similar ambiguity arises whenever the right bracket is not filled. In such cases, it is hard to tell (automatically) where to draw the boundary between MF and NF, as in (6a). One option would be to use the syntactic category (S, VP, NP, PP, etc.) as an indicator of the position: usually, S and (most) VPs are located in NF, NPs in MF, and PPs can be in MF or NF. However, one aim of annotating (and querying) corpora is exactly to verify such common wisdom.

The MF and NF templates both require that the right bracket be filled, to minimized incorrect matches that result from an unclear position of the right bracket. This excludes a lot of instances (false negatives), such as (6a). At the same time, the (very) simple heuristics applied in the template also yields false positives (6b) (the beginning of the (incorrect) NF matches are marked in boldface).

(6) a. " Ich glaube kaum , daß mit seinem , naja , etwas undiplomatischen Stil im Weißen Haus dem Land ein Gefallen getan wäre $_{NF}$. (s24)

b. So will der politische Außenseiter beispielsweise das Steuersystem vereinfachen , **das** Bildungssystem verbessern , **das** gigantische Haushaltsdefizit abbauen , **Einfuhren** aus Japan drosseln **und** die geplante Freihandelszone der USA mit Mexiko verhindern . (s37)

[10] Such cases do occur: in the TüBa-D/Z treebank, there are 720 (0.84%) MF-less instances of the form VF-LK(-NF), and 125 (0.15%) of the form VF-RK (in the TüBa-D/Z scheme, the VF constituent is placed under a C node in the second type of constructions, cf. Fn. 5).

(iii) Constraints of the search tool The query language of TIGERSearch supports searches for linguistic relations such as precedence and dominance relations. It is not a programming language, though. Hence, certain query constraints cannot be formulated (or would require complex constraints). This includes cases where mother and daughter constituents match the query but only the highest, maximal one is correct. This happened, e.g., with the first version of the Nachfeld (NF) template that searched for a (i.e. *some*) constituent following the right bracket (RK), see (7): the S node occupies the Nachfeld but both the S and NP nodes matched the NF query. (The current NF template only matches the first word of the NF.)

(7) " Es ist wirklich schwer $[_{RK}$ zu sagen] , $[_{S}$ $[_{NP}$ **welche Positionen**] **er einnimmt**$_{NF}$] **,** da er sich noch nicht konkret geäußert hat " , beklagen Volkswirte . (s36)

Another example are cases where a topological field does not correspond to a single TIGER constituents. Variables in TIGERSearch queries always correspond to single constituents. Hence, for complex fields like the Mittelfeld (MF), which can consist of multiple constituents, two variables have to be used, one marking the beginning of the MF (MFB), one marking the end (MFE). Similarly, complex verb clusters in the right bracket (RK) and multiple Nachfeld constituents cannot be matched by a single variable.

(iv) Design of the the annotation scheme The crossing edges of the TIGER scheme are hard to query in general (see below). Certain sentences contain edges that encode dependencies rather than constituents, without resulting in crossing branches, though. This concern different types of left dislocation with resumptive elements, as in (8). In such cases, constraints on the number of constituents (e.g. in VF) cannot be applied sensibly.

(8) $[_{PP}$ $[_{S}$ **Daß Perot ein Unternehmen erfolgreich leiten kann**] , **davon**$_{VF}$] sind selbst seine Kritiker überzeugt . (s6)

The last example shows that the queries would have to provide exceptions for individual cases. Such an approach is not desirable in general because it uses queries to *encode* a lot of information rather than to simply *extract* information from the treebank.

(v) Crossing branches Turning now to the enriched (ENR) scheme with crossing branches, it is obvious that extra efforts have to be made to correctly treat discontinuous constituents. Fig. 4 shows an example sentence (left) that would not be matched by the VFmain template in the appendix because the right corner of the NP node does not precede the finite verb. In contrast, the VFsub template incorrectly matches the phrase *was [. . .] eigentlich machen* of the other example sentence in Fig. 4 (right) because the right corner of the VP node is adjacent to the finite verb.

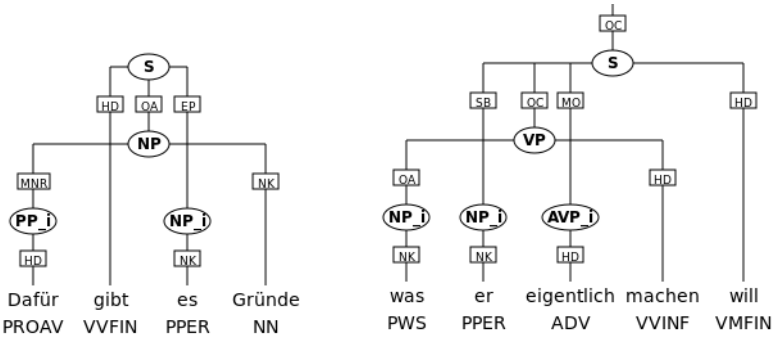

Figure 4: TIGER sentences no. 86 (left) and 48 (fragment; right) in ENR version with crossing branches ('There are reasons thereof' (left) and '... what he actually wants to do' (right))

What we need here is a way to address the daugther nodes that are continuous. TIGERSearch provides an operator "discontinuous()" which can be used here. Relevant parts of the template are shown in the appendix.[11] This way, (many of the) missed discontinuous cases (i.e. false negatives like in Fig. 4, left) can be matched. What is rather unclear, however, is how false positives (Fig. 4, right) can be excluded.

4.2 Quantitative results

I evaluted the templates by comparing the query results of the sentences 101–600 with the manually-annotated fields. Fields and brackets of the queries must span strings of words that are identical with the gold fields for counting as a match, i.e. overlapping spans were considered errors.

Table 2 shows the results from the evaluation set (TIGER sentences s101–s600). Due to performance issues and out-of-memory errors, not all templates could be run successfully by TIGERSearch, and had to be modified accordingly.[12] For multi-constituent fields, the gold annotations were transformed into boundary markers.

The table shows that the F-scores for VF, LK and RK are near or above 90% and quite good. In contrast, the MF templates have a bad recall, and the (simple) NF template yields the lowest F-score.[13]

[11] [4] investigates Vorfeld positions in the CGN corpus of spoken Dutch, which is annotated according to the TIGER scheme. [4] deals with discontinuous VF constituents roughly by assuming that either the entire constituent or at least its head must precede the finite verb to qualify as the VF constituent [4, p. 76]. It seems to me that this definition would fail to correctly determine the VF in the first example in Fig. 4 because the head occurs sentence-final.

[12] E.g. the original MF template referred to both the VF/LK template and the RK template, which made it computationally too expensive.

[13] [3] present a topological parser for German. The parser was trained on a version of the Negra corpus (which is annotated similar to the TIGER corpus) that has been automatically enriched with

Field		F1	Prec	Rec	#Gold	#System
VF		87.26	88.43	86.11	648	631
LK		93.01	97.11	89.23	678	623
MF:	MFB	58.27	96.74	41.69	854	368
	MFE	56.78	87.96	41.92	854	407
RK:	RKS	89.06	87.41	90.77	390	405
	RKB	83.42	78.67	88.77	187	211
	RKE	88.22	82.63	94.62	186	213
NF:	NFB	45.70	56.71	38.27	243	164

Table 2: Results of TIGERSearch template-based queries for topological fields: F-Score, Precision, Recall (all in %), number of instances in the gold and system data (i.e. query results)

5 Conclusion

To sum up the findings of this paper: The dependency-oriented TIGER annotation scheme (in its original form) does not really seem suitable for syntactic investigations at the level of topological fields. In particular, crossing branches that result from long-distance dependencies are difficult to handle, and especially excluding false positives is difficult.

Hence, converting the treebank to a context-free format is a good idea in general and facilitates further (automatic and manual) processing to a great extent. However, searching for topological fields in this format still requires complex templates and a considerable amount of processing time. What we actually need is a version of the TIGER corpus enriched with topological-field annotations. For some of the fields (VF, LK, RK), automatically adding topological fields seems feasible (especially if a powerful programming language is used). Other fields (MF, NF) would require manual work.

Approaches like the one taken by the TüBa-D/Z scheme seem favorable, by explicitly annotating topological fields from the beginning. So why not just stick to the TüBa-D/Z corpus? I think there are two main reasons why it is favorable to be able to use both treebank, TüBa-D/Z and TIGER. First, both are only medium-sized (TüBa-D/Z, release 9: around 85,000 sentences; TIGER: around 50,000 sentences). Second, while both cosist of texts from newspapers from the 1990s, the style differs to some extent: TIGER contains texts from *Frankfurter Rundschau*, TüBa-D/Z from *taz*, which is a rather progressive newspaper. So in an ideal world, users would probably like to exploit both treebanks.

topological field annotations. They report 93.0% precision and 93.7% recall for the enrichment script. Unfortunately the script is no longer available.

Acknowledgements

I would like to thank the anonymous reviewers for helpful comments. Many thanks to Ronja Laarmann-Quante for creating the gold topological annotation of 600 TIGER sentences, and to Adam Roussel for the evaluation script.

References

[1] Stefanie Albert et al. TIGER Annotationsschema, 2003. Technical Report, Universität des Saarlandes, Universität Stuttgart, Universität Potsdam, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/annotation/tiger_scheme-syntax.pdf`.

[2] Markus Bader and Jana Häussler. Word order in German. a corpus study. exploring the left periphery. *Lingua*, 120(3):717–762, 2010.

[3] Markus Becker and Anette Frank. A stochastic topological parser for German. In *Proceedings of COLING-2002*, Taipei, Taiwan, 2002.

[4] Gerlof Bouma. *Starting a sentence in Dutch. A corpus study of subject- and object-fronting*. PhD thesis, University of Groningen, 2008.

[5] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation, Special Issue*, 2(4):597–620, 2004.

[6] Stefanie Dipper and Sandra Kübler. German Treebanks: TIGER and TüBa-D/Z. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*. Springer, Berlin, forthcoming.

[7] Werner Frey. A medial topic position for German. *Linguistische Berichte*, 198:53–190, 2004.

[8] Erhard Hinrichs and Tsuneko Nakazawa. Linearizing AUXs in German verbal complexes. In John Nerbonne, Carl Pollard, and Klaus Netter, editors, *German in Head-Driven Phrase Structure Grammar*, pages 11–38. CSLI, Stanford, 1994.

[9] Tilman Höhle. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany, 1986.

[10] Wolfgang Lezius. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. PhD thesis, Universität Stuttgart, 2002. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), volume 8, number 4.

[11] Stefan Müller. Zur Analyse der scheinbar mehrfachen Vorfeldbesetzung. *Linguistische Berichte*, pages 29–62, 2005.

[12] Stefan Müller. Qualitative Korpusanalyse für die Grammatiktheorie: Introspektion vs. Korpora. In Gisela Zifonun and Werner Kallmeyer, editors, *Sprachkorpora — Datenmengen und Erkenntnisfortschritt*, IDS-Jahrbuch 2006. de Gruyter, Berlin, New York, 2007.

[13] Adam Roussel. Documentation of the tool TIGER Tree Enricher. `http://www.linguistics.ruhr-uni-bochum.de/resources/software/tte`, 2014.

[14] Yvonne Samuelsson and Martin Volk. Automatic node insertion for treebank deepening. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT)*, pages 127–136, Tübingen, 2004.

[15] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset), 1999. Technical report, Universitäten Stuttgart und Tübingen, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-1999.pdf`.

[16] Augustin Speyer. Die Bedeutung der Centering Theory für Fragen der Vorfeldbesetzung im Deutschen. *Zeitschrift für Sprachwissenschaft*, 26:83–115, 2007.

[17] Jan Strunk and Neal Snider. Subclausal locality constraints on relative clause extraposition. In Heike Walker, Gert Webelhuth, and Manfred Sailer, editors, *Rightward Movement from a Cross-linguistic Perspective*, pages 99–143, Amsterdam, 2013. John Benjamins.

[18] Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany, 2012.

[19] Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. WebAnno: A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 1–6, 2013.

Appendix: TIGERSearch templates

- Templates for the Vorfeld position in main and subordinate clauses in context-free format (VFmain_cf and VFsub_cf)
- Extension of the VFmain_cf template for discontinuous constituents (VF-main_enr)
- Definition of the precedence relation

```
// VF (and V2) in main clauses
VFmain_cf(#vf,#v2) <-
  #s: [cat="S"]
& #v2: [pos=/V.FIN/]    // #v2: Verb in second position
& #s > #vf              // #vf: Vorfeld constituent
& #s >HD #v2

// VF is first constituent
& ( // 1. VF is very first element in the sentence
    hasLeftChild(#s,#vf)   // #vf is left-most child
  | // 2. Or some coordinating conjunction precedes VF
    #s >@l #conj
  & [] >JU #conj
  & prec(#conj,#vf)
  )

// VF precedes VFIN
& ( // 1. VF directly precedes V2
    prec(#vf,#v2)
  | // 2. A comma may intervene after clausal or appositive VF
    ( #vf: [cat=("S"|"VP")]           // either VF itself precedes comma
    & prec_comma(#vf,#v2)
    | #vf >* #clause_app              // or some embedded constituent
    & ( #clause_app: [cat=("S"|"VP")]
      | [] >APP #clause_app
      )
    & prec_comma(#clause_app,#v2)
    )
   );

// VF in subordinate clauses
VFsub_cf(#vf) <-
  #s: [cat="S"]
& #s > #vf               // #vf: Vorfeld constituent
& // VF is very first element in the sentence
  hasLeftChild(#s,#vf)   // #vf is left-most child
& #vf >* [pos=/.*(REL|W).*/];  // relative or interrogative elements

// Discontinuous VF
VFmain_enr(#vf,#v2) <-
// VF contains discontinuous element -> take daughter node
  #s: [cat="S"]
& #v2: [pos=/V.FIN/]
```

```
& #s > #vfin
& #s >* #vf_disc // #vf_disc: disontinuous mother of VF constituent
& discontinuous(#vf_disc)
& #vf_disc > #vf
& ...

// Precedence relation
prec(#x,#y) <-
( // 1. #x is a terminal node
  #x: [word=/.*/]
& #x . #y
| // 2. #x is non-terminal
  #x: [cat=/.*/]
& #x >@r #xchildR
& #xchildR . #y
| // 3. quotes may intervene (everywhere)
  prec_quote(#x,#y)
);
```