Treebanks and Linguistic Theories 2004
Sandra Kübler and Joakim Nivre (eds.)

Annotating and Querying a Treebank of Suboptimal Structures

Stephan Kepser, Ilona Steiner, and Wolfgang Sternefeld

SFB 441, University of Tübingen, Germany

{kepser, steiner}@sfs.uni-tuebingen.de,
wolfgang.sternefeld@uni-tuebingen.de

1 Introduction

Existing treebanks of written language, as e.g., TIGER [2], TüBa-D/Z [11], Penn Treebank [1] etc., usually consist of sentences that can be considered as grammatically well-formed. The SINBAD treebank we present here covers a completely new domain, namely suboptimal syntactic structures, i.e., sentences which are neither fully grammatical nor completely ungrammatical, but merely suboptimal.[1] The treebank consists of a collection of German sentences that are rated suboptimal or ungrammatical in the literature, as well as of sentences drawn from our own experimental work on graded grammaticality judgments. In the literature, these structures are usually compared with grammatical structures which express the same meaning, and for ease of comparison these were sometimes included in the treebank as well. With this data collection we provide access to negative evidence which does not occur in ordinary corpora of written or spoken language.

It is characteristic for suboptimal structures that these data are judged incoherently varying between different speakers and in different contexts. It is therefore important to provide a systematic collection of these judgments in order to allow researchers better access to past judgements on the phenomena they are interested in and thus contribute towards greater consistency, even in tricky cases. Since most work in syntactic theory is based on suboptimal or ungrammatical structures, the treebank aims at providing linguists with a data basis for their research. This requires a rich syntactic annotation with linguistically relevant concepts. The linguistic framework of the annotation is that of generative grammar in the sense that the trees are strictly binary branching and contain traces and empty categories. The

[1] Note that the term *suboptimal* is referred to grammaticality and not to mere processing considerations. Garden-path sentences, for example, are excluded from this domain.

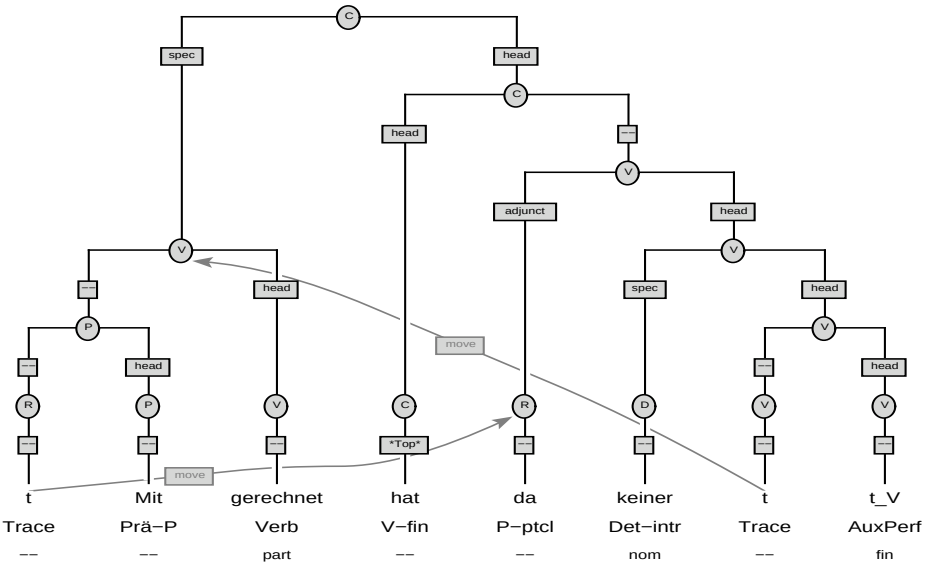

Figure 1: Remnant movement in German [5]

annotation scheme is inspired by the feature grammar which Sternefeld developed for German [9]. To our knowledge this is the first treebank following principles of generative grammar.

The new domain of suboptimal structures and the particular linguistic framework chosen raise additional research questions with respect to annotation schemes as well as querying these structures. In Section 2 we present the design principles chosen for our treebank, in Section 3 we focus on the how these structures can be queried effectively.

2 Syntactic Annotation of Suboptimal Structures

The treebank of suboptimal structures is work in progress and comprises ca. 1060 sentences at the moment. The intended size of the treebank is about 3000 sentences with the target being more a qualitative than a quantitative one. It has been annotated manually by one student assistant using the *Annotate* tool [6]. Fig. 1 shows a sample entry of our treebank: *Mit gerechnet hat da keiner* (lit. “With reckoned has it nobody”, meaning “Nobody expected that”). This sentence is rated suboptimal (‘?’) in the literature (taken from [5]).

The approach was to build up a modest basis of data, and then develop the analytical framework on the basis of this partial data set. The major part of this task has been completed, larger quantities of examples can be added, without the

danger that they need to be recoded in an architectural redesign. To ensure accuracy and consistency of the annotations, the treebank has been checked in several proof-reading sessions. In addition, the query tool fsq (see next section) has been used to eliminate errors in the annotation and to ensure consistency of the data.

2.1 Design Principles

Using a generative framework for the annotation is challenging, because it may well be that "a sentence has as many structures as there are theories" (Haider, [3]). Nonetheless, we tried to find a compromise between (a) naive expectations of a linguistically trained user (b) run of the mill assumptions in generative grammar (c) simplicity of structure, and (d) enhanced parsability. In accordance with these aims we attempted to minimize the number of different syntactic categories, to minimize occurrences of empty categories, to minimize inexplicitness of structure by strictly adhering to binary branching, and to minimize the role of X-bar theory by following minimalistic assumptions. As a result of these requirements, we maximized the analytical importance of structure.

2.2 Annotation Scheme

The treebank is annotated with Part-of-Speech tags (PoS tags), morphological information, syntactic categories (node labels), grammatical functions (edge labels) and additional contextual features (lexical edge labels). In addition, secondary edges are used for the annotation of movement and co-reference. The details of the annotation scheme are described in the SINBAD stylebook [10].

2.2.1 Node Labels

Node labels specify the major syntactic categories of constituents. Due to the richness of syntactic structure it is possible to reduce the number of node labels to a minimum of seven different syntactic categories:

- A the category of adjectives and adverbials
- C the category of complementizers and the position of the finite verb in main clauses
- D the category of determiners, including intransitive determiners like pronouns and proper names
- N the category of common nouns including proper nouns
- P the category of adpositions, i.e., pre- and postpositions
- V the category of verbs
- R a default category for anything that does not fit into the above categories

Categories like AP, CP, DP etc., which are primitives of traditional X-bar theory, are dispensed within our annotation, but can be defined with the help of edge labels, as will be shown further below. Note also that there is no Infl category in our annotation. Following the theory developed by Sternefeld [9], clauses are CPs, and the complement of C is a VP.

2.2.2 Part-of-Speech Tags and Morphological Labels

PoS tags subcategorize the seven node labels according to their morpho-syntactic lexical properties as illustrated in Table 1 below. We opted to develop our own PoS tag set for the following reasons. A considerable amount of information encoded in existing tag sets, such as the STTS [7], is already encoded in our annotation in a different way using edge labels, tree structure or morphological informations. We wanted to avoid the redundancy of restating that. Furthermore existing PoS tag sets do not adequately capture the linguistic intentions of the annotation; they thrive to be theory-neutral while our tag set is derived from the linguistic framework we use.

Subcategories of A		
Ad	adverb, predicative adjective	er fährt/ist *schnell*
A-infl	inflected adjective	ein *schneller* Fahrer
Adv	adverbial	*heute, schon, bald*
W-Pron	wh-pronoun	*wie* geht es dir?
Subcategories of C		
V-fin	the finite verb in C	Fritz *schläft* ein
C-fin	complementizer with finite clause	*dass* er kommt
C-zu	complementizer with infinite clause	*um* zu arbeiten, *anstatt*
Subcategories of D		
W-Pron	wh-pronoun	*wer, wessen, was, welcher*
Rel-Pron	relative pronoun	*dem, dessen*
Poss-Pron	possessive pronoun	*mein, dein, unser*
Refl-Pron	reflexive pronoun	*sich*
Rec-Pron	reciprocal pronoun	*einander*
Pers-Pron	personal pronoun	*ich, du, er,. . . , mich, dich. . . , meiner, mir* etc.
Prop-N	proper name	*Fritz, Anna, Fritzens* Mut, *Annas* Kleid
Det	transitive determiner	*d-er, jed-er, ein, kein*

Det-intr	intransitive determiner	das ist *meins*, da ist *keiner*, *den* kenne ich, *PRO*
Subcategories of N		
CN	common noun	*Haus, Wand, Eis, Gold*
PN	proper noun	der *Hans*, die *Schweiz*
Subcategories of P		
Prae-P	preposition	*in, an, auf, mit, ohne, von*
Post-P	postposition	*wegen, halber*
P-Adv	pronominal adverb	*damit, davon*
P+Det	preposition + determiner	*im, am, ins*
Subcategories of V		
AuxMod	modal auxiliary	*wollen, können, müssen, dürfen, sollen*
AuxPerf	temporal auxiliary	*haben, sein*
AuxFut	temporal auxiliary	*werden*
AuxPass	passive auxiliary	*werden, kriegen, bekommen*
AuxModPass	modal passive auxiliary	*sein*
A.c.I.	exceptional case-marking verb	*lassen, sehen, hören, fühlen*
Rais	raising verb (not one of above)	*scheinen, pflegen, haben + zu*
Cntr	control verb (not one of above)	*wünschen, möchten, versuchen, befehlen*
Verb	main verb (not one of above)	Fritz hat *geschlafen*
Subcategories of R		
Ptcl	particle	*wohl, ja, noch*
P-ptcl	stranded preposition particle	*da* (from *damit, daher*)
V-ptcl	verbal particle	wenn er *weg*läuft
W-ptcl	*was-für*-particle	*was* für Menschen
Neg	negative particle	*nicht*
Category-independent PoS tags		
Trace	trace	t
Conj	conjunction	*und, oder, (so)wie*

Table 1: The SINBAD PoS tagset

Morphological labels are those for case markings on determiners, nouns, and adjectives (nom, acc, dat, gen) and those for inflection on verbs (fin, inf, part (par-

ticiple), to (to-infinitive)). Nouns and adjectives will only be labelled when having an explicit morphological case marking, i.e., a case affix (different from zero affixation). In contrast to this, determiners always bear a morphological label, even if it is a null determiner. Other morphological categories like person, number, and gender were not relevant in the hitherto recorded sentences, but could easily be added in future applications.

2.2.3 Edge Labels

We distinguish between lexical edge labels and syntactic edge labels. Lexical edge labels are the edge labels directly above the lexical layer and encode additional contextual information as *W* (the specifier of C contains a wh-item), *Rel* (the specifier of C contains a relative pronoun) and *TOP* (the specifier of C is a topicalized phrase).

Syntactic edge labels indicate head-complement or head-adjunct relations between two sister nodes. The node labels together with the syntactic edge labels constitute a minimal residue of X-bar theory. These are the syntactic edge labels:

adjunct	immediately dominates an adjunct
head	immediately dominates a head
rel-head	immediately dominates a relativized head
spec	immediately dominates a specifier
– –	immediately dominates a complement

Typical adjuncts are prenominal adjectives, relative clauses and adverbials. Typical specifiers are the SpecC position, prenominal genitive DPs and possessive pronouns, and the subject of a predicate; these will always be immediately dominated by the edge label spec. The head label is employed to encode a residue of X-bar theory. Any node which is not a head is a maximal projection. This way, categories like NP or CP can be dispensed with: A *maximal projection* NP can be defined as an N-node that is not immediately dominated by the edge label head.

2.2.4 Secondary Edge Labels

Secondary edges denote specific relations between nodes, represented as arrows. We identify four types of constructions or grammatical relations:

move	movement relating a trace to its antecedent
co-ind	co-indexing for the purpose of binding theory relating an anaphora to its antecedent

es-ko	*es*-correlative constructions relating the pronoun *es* to a coreferential, extraposed CP
w-w	*was*-*w*-constructions (partial movement) relating a partially moved wh-phrase to *was*

2.2.5 Null Elements

Although to some extend we avoid the use of empty categories, we still formally distinguish five types of empty lexical items:

pro	the subject of subjectless finite clauses
PRO	the empty subject of an infinitival CP
t_V	the trace of a verb-second movement
t	any other trace
0	any other empty category not mentioned above

pro only appears if there is no other way to satisfy some version of the extended projection principle, i.e., there is no nominative that could be argued to be the subject of a finite clause. In general, this is only the case in impersonal passive constructions. PRO is the subject of CPs headed by C-zu. The remaining zero categories represented by "0" are empty determiners, empty *wh*-operators, empty complementizers and empty conjunctions.

Traces are left by every category that has been moved to another position in the tree. Note, however, that we admit the following exception: In verb-second movement, we decided that the PoS tag of the moved verb in C is V-fin, the PoS tag of the trace is not Trace but the original one of the moved verb. The trace of V/2 itself is marked by t_V to distinguish it from other traces which are always connected with the element which has been moved by a secondary edge label. For perspicuity, we tried to reduce the role of movement to a minimum. For example, subjects may be directly generated in SpecC, without moving from within VP; this allows one to distinguish between genuine topicalizations and normal SVO order.

2.2.6 General Considerations

The annotation schema chosen for our treebank is completely different compared to those for existing German treebanks as TIGER, TüBa-D [8], Tüa-D/Z. These annotation schemes do not reflect a commitment to a particular syntactic theory. The syntactic structures are rather flat and simple and do not contain empty categories or traces. See, for example, the 'flat clustering principle' used in TüBa-D and TüBa-D/Z [8, 11] which keeps the number of hierarchy levels in a syntactic structure as small as possible. In the Penn Treebank, empty categories are anno-

tated, but here again a relatively flat context-free notation is used without leaning towards a particular theoretical view.

The advantage of our annotation scheme is that the treebank contains much more information than ordinarily available. Linguistically relevant concepts such as c-command, extraction, pied piping, remnant movement, freezing, and many others are explicitly or implicitly encoded in terms of structure or secondary edges. These concepts are not necessary local and therefore cannot be encoded in other German treebanks; nonetheless they are absolutely crucial for any generative theory of language.

3 Querying Suboptimal Structures

In the treebank presented here, deep syntactic structures are used for the annotation and linguistic information is often encoded implicitly (e.g., the relation c-command). These characteristics pose a specific challenge for query tools and the power of their query languages. We therefore selected the query tool fsq [4] which allows the user to search treebanks for complex syntactic constructions and offers full first-order logic as query language.

3.1 The Query Language of fsq

The properties of a tree in the treebank are expressed as properties of nodes in the tree and relations between nodes. Properties of individual nodes are the annotation labels. That is to say, a nonterminal node has a major category and a grammatical function, which is the syntactic edge label described above. Terminal nodes have part-of-speech labels, lexical edge labels, and tokens and can bear additional morphological information.

Relations between nodes describe (part of) the structure of a tree. Hence, these relations comprise the mother-daughter-relation, also called immediate dominance, the dominance and proper dominance relation, which are the reflexive-transitive and the transitive closure of the mother-daughter-relation. The precedence relations are orthogonal to these, describing the left-to-right orientation in a tree. A node x precedes another node y, if the whole subtree rooted in x is to be found to the left of the whole subtree rooted in y. A node x immediately precedes y, if x precedes y and there is no node in-between, preceeding y and being preceeded by x. There can also be secondary relations between nodes, e.g., a move-relation. And one may express equality or disequality of two nodes.

Most of the above described properties of nodes and relations between nodes can be expressed in many existing treebank search engines. The query language

of fsq is the full first-order logic over these properties and relations as atomic formulae. More explicitely, the properties and relations are formulae of fsq. The negation of a formula, the conjunction, disjunction and implication of formulae are again formulae. And existential or universal quantification of a node variable and a formula is again a formula. It is in particular the arbitrary quantification that provides the high expressive power of the query language. No other off-the-shelf query tool offers a comparable expressive power, which is often needed for the expression of linguistically important relations. A simple, but frequent example is the description of a complex structure in which a certain undesirable feature is absent. This requires universal quantification over all nodes in the complex structure, because *no* node is supposed to bear the feature.

3.2 C-Command and Remnant Movement

Let us explain the use of the query language by means of two examples that have strong linguistic motivations. The first example is that of *c-command*. This notion plays an important role in the binding theory. Roughly, a node c-commands her sister nodes and all the nodes that her sister nodes dominate. Formally, a node x c-commands another node y if there is a third node z that is the mother of x and that dominates y, i.e., $\exists z(z > x \wedge z >+ y) \wedge \neg x >> y$. The second conjunct excludes cases where x dominates y. The situation is actually a little bit more complicated if the node taking command is a terminal node. Due to the annotation scheme of SINBAD, the preterminal level is unary branching. In other words, the mother of a terminal node x is never the mother of any other node than x. To get to a properly branching node we have to go the the grandmother of a terminal node. Formally $(\neg\exists z x > z) \wedge \exists z, w\, w > x \wedge z > w \wedge z >+ y \wedge \neg w >> y$. To consider the terminal and the nonterminal case the disjunction of the two formulae has to be taken. But since the formula for the case of a terminal node has a higher quantifier depth, it should be used only in those circumstances where it is needed. Often linguists consider a c-command relation between nonterminal nodes, and in this situation, the simple formula stated first suffices.

Remnant movement describes the leftward movement of a complex structure out of which a smaller substructure is already moved. Consider Figure 1 as an example. Here, the complex VP [*Damit gerechnet*] is moved into the topic position of the sentence, which is the specifier of the CP. This complex VP contains the trace of the particle *da* which was moved out of the VP before the VP is moved. A necessary precondition for this type of construction is that the landing position of the smaller structure is c-commanded by the landing position of the large structure out of which it was moved. Due to the fact that movement is explicitely annotated in the treebank via a secondary edge with label move, it is simple to search for

instances of remnant movement in the treebank. Remnant movement of node x, where x is the root of the complex structure that is moved, can be expressed by the following formula: $\exists y\,\mathsf{move}(y,x) \wedge \exists w,z\ \mathsf{move}(z,w) \wedge x >> z \wedge x$ c-commands w. The first conjunct expresses the existence of a move-secondary edge that ends in x. The second conjunct expresses the movement of the smaller substructure. It is moved from node z which is dominated by x to a landing position w that is c-commanded by x.

3.3 The Web Interface

The treebank is available on the web under the following URL: `http://barlach.sfb.uni-tuebingen.de/~a3/`. This site gives access to a structural search as well as to a keyword search to be described below.

The tree structure search is realized as a web interface to fsq. Part of fsq is a graphical user interface that systematically supports users in constructing queries. When composing a query most users think in a bottom-up fashion focusing first on the atomic constituents. This approach is supported by the user interface in the following way. An *Atomic* menu lets the user compose atomic formulae. He picks the relation of his choice, say, e.g., the dominance relation. He is successively asked for names of the variables one dominating the other. Thereafter, the syntactically correct formula is added to the list of formulae. The other atomic formulae can be constructed in a similar fashion.

In order to get more complex formulae, the user can choose operations from the *Complex* menu. It contains menu options for the boolean connectives and quantifiers. To compose, e.g., a conjunction, the user first chooses the formulae he wishes to conjoin by clicking on them in the list of formulae. Thereafter he just picks the *Conjunction* menu item and the conjunction of the formulae he chose is added to the list of formulae. In case of an existential or universal quantification, the user selects a formula from the list, and, e.g., the *Existential Quantification* menu item. He will be asked for the name of the variable to quantify over, and the existentially quantified formula is added to the list of formulae.

We transformed the graphical user interface of fsq into an applet and modified it for our purposes. With the help of the applet, queries can be composed, edited and submitted from a standard browser. In addition to the predicates and relations provided by fsq, we offer macros encoding such linguistic constructs as c-command, head relation, extraction, and remnant movement, which can be combined with other relations, thus forming complex queries. These are transmitted from the applet to a cgi-script, which starts the fsq engine and displays the retrieved sentences in HTML format.

Linguists however are not exclusively interested in searching a collection of

suboptimal tree structures, they of course are also interested in additional information as the grammaticality rating of a given sentence, the reference source, etc. Therefore the treebank is embedded in a larger system that also comprises a MySQL database containing these informations. Accordingly, for each sentence retrieved by the tree structure search, the user can request the syntactic annotation as a tree, the source, the set of structurally similar sentences and their ratings as given by the author.

The database also contains an extensive description of each tree in the form of a set of keywords. The keywords are grouped into six areas of linguistic properties of a tree: wh-movement, topicalization, scrambling, binding, extraposition and dislocation, and complementation. For each area, there exists a fine grained list of potential features. As an alternative way to search the treebank we provide a web interface to this keyword database.

Keyword search is simple and may be more appealing to novel users of the treebank. But it provides access only to a proper subset of the structural properties of trees in the treebank. Every keyword search can also be performed by an fsq query. But there are interesting complex queries that cannot be expressed by keyword search.

4 Conclusion

We presented a treebank of suboptimal structures in German. The novelty of the present work is threefold. Our treebank is the first treebank for German that provides analyses of trees within the framework of generative grammar. It is also the first treebank to provide suboptimal sentences together with their grammaticality judgments. It is therefore of high importance for generative linguistics of German. To offer an open access to the treebank we subplanted the treebank with a very powerful query system that is accessible via the web. It is especially this accessibility that makes the treebank so useful for linguists. Future developments include an extension of the size of the treebank and an implementation of techniques to shorten query response times.

Acknowledgments

Kudos goes to Monika Toth, who annotated the trees in the treebank, and to Tanja Kiziak, who is a co-developer of the web interface of fsq. We would also like to thank Sam Featherston and Tanja Kiziak for interesting discussions and helpful comments. This research was funded by a grant from the German Research Council (DFG SFB-441).

References

[1] Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. Bracketing Guidelines for Treebank II style Penn Treebank Project. Technical report, University of Pennsylvania, 1995.

[2] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In Kiril Simov, editor, *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, 2002.

[3] Hubert Haider. *Deutsche Syntax – Generativ*. Narr, Tübingen, 1993.

[4] Stephan Kepser. Finite Structure Query: A Tool for Querying Syntactically Annotated Corpora. In Ann Copestake and Jan Hajič, editors, *Proceedings EACL 2003*, pages 179–186, 2003.

[5] Gereon Müller. *Incomplete Category Fronting. A Derivational Approach to Remnant Movement in German*. Number 42 in Studies in Natural Language and Linguistic Theory. Kluwer, 1998.

[6] Oliver Plaehn and Thorsten Brants. Annotate – An Efficient Interactive Annotation Tool. In *Sixth Conference on Applied Natural Language Processing (ANLP-2000)*, 2000.

[7] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS. Manuscript, Universities of Stuttgart and Tübingen, 1995.

[8] Rosmary Stegmann, Heike Telljohann, and Erhard Hinrichs. Stylebook for the German treebank in VERBMOBIL. Technical Report 239, SfS, University of Tübingen, 2000.

[9] Wolfgang Sternefeld. Syntax. Eine merkmalsbasierte generative Analyse des Deutschen. Book manuscript, 2004.

[10] Wolfgang Sternefeld. The SINBAD Stylebook – **S**ammlung **IN**teressanter **B**eispiele **A**us'm **D**eutschen. Technical report, SFB 441, University of Tübingen, 2004.

[11] Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, 2003.