Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank

Martin Forst[†], Núria Bertomeu[*], Berthold Crysmann[*], Frederik Fouvry[*], Silvia Hansen-Schirra[*], Valia Kordoni[*]

[*]Computational Linguistics & Phonetics, Saarland University
Postfach 15 11 50, 66041 Saarbrücken, Germany
{bertomeu|crysmann|fouvry|hansen|kordoni}@coli.uni-sb.de

[†]Institute for Natural Language Processing, University of Stuttgart
Azenbergstr. 12, 70174 Stuttgart, Germany
forst@ims.uni-stuttgart.de

Abstract

In this paper we discuss the construction, features and intended uses of the TiGer DB. The TiGer DB is a dependency bank derived from the TiGer Treebank containing predicate-argument relations and several grammatical features which can be considered as semantically meaningful. It is produced semi-automatically by the conversion of the TiGer treebank into an LFG f-structure bank, which then in turn is converted into the TiGer DB. This allows for a relatively rapid construction. The grammatical relations and features encoded in the TiGer DB are chosen in order to keep the mapping from parser output, e.g. LFG f-structures or HPSG feature structures, to dependency triples simple. Hence, the TiGer DB can be used as a gold standard for the evaluation of German parsers.

1 Introduction

The TiGer Dependency Bank (henceforth TiGer DB) consists of a part of the TiGer Treebank, currently about 40,000 syntactically annotated sentences of German newspaper texts (Brants et al., in press). It is annotated with so-called dependency triples, i.e. a functor representing a grammatical relation or feature and two arguments representing the head and the value of this feature respectively. The format of these dependency triples is the same as in the PARC 700 Dependency Bank (King et al., 2003), which makes it possible to use the tools for displaying and pruning structures that are available together with this English dependency bank. The grammatical relations encoded in the TiGer DB are to a fair extent identical to the edge labels used in the TiGer Treebank; in order to make it more suitable as a basis for the evaluation of German parsers, additional distinctions have to be made in the set of grammatical relations, which leads to an enlarged set of features compared to the TiGer Treebank.[1]

The TiGer DB is created because the graphs of the TiGer Treebank themselves are difficult to use as a gold standard for German parsers for a number of reasons: The constituency annotation in the TiGer graphs has the advantage of being fairly theory-neutral, but (i) since it includes discontinuous constituents and secondary edges, it cannot be mimicked by any of the German parsers we are aware of. Besides, (ii) the tokenization (and lemmatization) of certain multiword expressions, compounds etc. differs from the analyses most parsers obtain. The functional annotation in the TiGer graphs is more suitable for the evaluation and comparison of parsers across theoretical frameworks, but (iii) it is intimately tied to the constituency annotation, including edge labels such as ADC[2] (multi-token adjective component), which only exist due to the lemmatization decisions mentioned above, and (iv) it does not encode all information and distinctions that (deep) parsers are supposed to obtain.

Similar problems arise with other syntactically annotated corpora of German text and speech, such as the Negra Corpus (Skut et al., 1998; Brants et al., 1999), the Verbmobil Corpus (Wahlster, 2000) and the Tübingen Treebank of Written German (Telljohann et al., 2003),

[1]It is planned to distribute the TiGer DB alongside the TiGer Treebank, which is freely available for research and evaluation, and to include detailed documentation in the release. For more information, please refer to http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/.

[2]Upper case labels will be used for the functional annotation encoded in the TiGer Treebank and lower case labels for the dependencies encoded in the TiGer DB.

since they all encode constituency and dependency information in one structure, the latter being biased by the former. Grammar developers, however, are interested in pure dependency representations, which allow for a much more meaningful evaluation than the bracketing of constituents, and are therefore clearly moving away from treebanks to dependency banks (Carroll et al., 1999; Carroll et al., 2003).

In this paper we present the strategy that has been adopted in the TiGer project for semi-automatically obtaining a dependency bank that will be of use for a wide variety of applications, in particular parser evaluation for various formalisms.

2 Constructing the TiGer DB

In this section, the method for producing the TiGer DB is presented. It is a combination of automatic and manual techniques in order to achieve the most accurate and consistent results in a reasonable amount of time. The basic process is as follows:

1. Convert the TiGer graph into an f-structure chart (packed representation of one or several f-structures).
2. Match the resulting f-structure chart against the output of a broad-coverage LFG for German and bank the compatible reading(s).
3. For all sentences for which there are either several or no compatible readings, select the correct/best analysis manually.
4. Fully automatically convert the selected f-structure into dependency triples.
5. Manually check/correct each structure using the pretty-printing and validation tools that are distributed with the PARC 700 Dependency Bank.

2.1 Automatic derivation of f-structure charts from the TiGer Treebank

The conversion of TiGer graphs into f-structure charts is described in Forst (2003a) and Forst (2003b). It takes the TiGer graphs encoded in TiGer XML as input and produces f-structure charts, i.e. packed representations of one or several f-structures. The ambiguity in the mapping from TiGer graphs to f-structures is due to information lacking in the TiGer Treebank, such as information concerning the decomposition of compounds, the nature of phrases labeled as *MO*s (modifiers), the subjects of infinite verb forms etc. In the conversion process it can be dealt with by means of optional rules, but for its use as a gold standard the output of the conversion has to be disambiguated, of course. How this can be done is discussed in 2.2 and 2.3.

Apart from changes due to the shift from one representation to another, we have decided to perform some changes to the analyses chosen by the TiGer Treebank annotators as well. They are motivated by the fact that the treatment given to these phenomena by all German parsers we are aware of differs from the analysis in the TiGer Treebank in a systematic way. One of these changes concerns PPs that are extracted from NPs, such as *statt dessen* in *Statt dessen gestand ihnen die Regierung eine Entschädigung zu:* (TiGer Corpus sentence #14839). In the TiGer Treebank, this PP is attached as an *MNR* (noun modifier) to the NP *eine Entschädigung*. Current German parsers, however, would attach this PP to the verb, since the attempt of attaching it to the NP would result in a massive increase of ambiguity. For a gold standard for German parsers, we consider it reasonable to encode the latter attachment rather than one that no parser would be able to achieve.

2.2 Automatic disambiguation of TiGer-derived f-structure charts

As a first step towards disambiguating the f-structure charts resulting from the fully automatic treebank conversion, these are matched against the output of a broad-coverage LFG for German (Dipper, 2003) and the compatible reading(s) are saved. This is done with the help of a Perl script that has been specifically developed for this purpose. Of course, this mapping can only be performed for sentences that are assigned a full parse by the broad-coverage LFG, and although the information both in the TiGer graphs and in the LFG parses is relatively detailed it can be impossible to fully disambiguate. Typical remaining ambiguities are due to the decomposition of compounds and to person and number ambiguities of possessive determiners and pronouns, these informations not being included in the TiGer Treebank.

Moreover, it has to be kept in mind that the mapping against the LFG output does not always retain the correct analysis, although most ambiguities can be resolved correctly in this way. This is particularly true for the *mo* (modifier) vs. *op* (prepositional object) distinction; an incomplete lexicon entry in the LFG can lead to the selection of the *mo* reading, even if the *op* reading is more adequate.

Nevertheless, the matching of the TiGer-derived f-structure charts is extremely useful. Not only does it help to disambiguate the ambiguous representations, but it also helps to increase consistency in the gold standard, since every time a match between the TiGer-derived structure and the grammar output is expected, but cannot be achieved, the human annotators can pay special attention to the phenomenon that caused the match to fail.

2.3 Manual disambiguation of TiGer-derived f-structure charts

All ambiguous TiGer-derived f-structure charts that cannot be fully disambiguated in the previous step have to be disambiguated manually. This is performed by visualizing the structures in the grammar development tool XLE (Maxwell III and Kaplan, 1993). It displays the packed representation of all f-structures encoded, the currently selected f-structure and an additional window, where the alternatives with the information differing among them are visualized. This allows human annotators to choose and save the correct reading in a relatively comfortable way. When none of the readings can be considered correct, the best analysis is selected and the annotator puts the sentence number on a list of structures to be reconsidered in the validation step.

2.4 Conversion into dependency triples and validation

The conversion from f-structures to dependency triples is fully automatic and unambiguous. It is carried out in basically the same way as it was done for the PARC 700 Dependency Bank (King et al., 2003). It mainly involves a certain amount of "flattening", i.e. articulate f-structures without a PRED have to be restructured, but this can be done without any loss of information. In addition to the flattening, a certain amount of renaming and reorganizing has to be carried out in order to make the structures meet the annotation principles outlined in Section 3.

In a final (and very important) step, each TiGer DB structure is manually evaluated by two people. If the structure is not correct, changes are made in the text-based representation of the structure. These changes are expected to be minor and to concern mainly the attachment of certain phrases in cases where the TiGer DB annotator disagrees with the annotation in the TiGer Treebank.

3 Grammatical relations and features encoded in the TiGer DB

The choice of the format and the dependencies encoded in the TiGer DB is crucial for its possible uses. Therefore the contents of the TiGer DB structures themselves are discussed in this section. Most of this information will be included in greater detail in the documentation distributed with the TiGer DB.

First we discuss indices, reentrancies and lemmatization. We then present the grammatical relations we have decided to encode in the TiGer DB and finally the atomic features chosen.

3.1 Indices, Reentrancies and Lemmatization

Just as in the PARC 700 Dependency Bank, all predicates in a given TiGer DB structure are assigned a unique index. For displaying reasons, the matrix predicate is always assigned the index 0. All other predicates are assigned the index corresponding to the ID of the terminal node in the TiGer Treebank that it relates to. Predicates which do not clearly relate to a terminal node in the TiGer treebank are given a "new" index. (This is the case for the compound non-head *privat∼1001* in Figure 2, for example.)

The use of indices has a number of advantages: First, they help to distinguish two instances of the same word. Second, they allow to express reentrant structures, i.e. structures in which a single item is related to more than one predicate. This occurs with controlled infinitives and with predicative constructions.

Consider the sentence *Privatmuseum muß weichen (Private museum must leave).* Its TiGer graph representation (in TiGer XML) and its representation as dependency triples are shown in Figures 1 and 2 respectively.

The matrix predicate of the sentence is the verb *müssen*; this is thus assigned the index 0. All other predicates are assigned the indices corresponding to the terminal nodes they relate to, which are 1 for *(Privat)Museum* and 3 for *weichen*. The "new" predicate *privat*, whose existence is due to the decomposition of compounds in the TiGer DB, is assigned a new unique index calculated on the basis of the index of its head and the position of the compound

```
<s id="s8595">
 <graph root="s8595_500">
  <terminals>
   <t id="s8595_1"
     word="Privatmuseum"
     pos="NN" morph="Nom.Sg.Neut"/>
   <t id="s8595_2" word="muß"
     pos="VMFIN" morph="3.Sg.Pres.Ind"/>
   <t id="s8595_3" word="weichen"
     pos="VVINF" morph="--" />
  </terminals>
  <nonterminals>
   <nt id="s8595_500" cat="S">
    <edge label="SB" idref="s8595_1"/>
    <edge label="HD" idref="s8595_2"/>
    <edge label="OC" idref="s8595_3"/>
   </nt>
  </nonterminals>
 </graph>
</s>
```

Figure 1: TiGer XML representation of TiGer Corpus sentence #8595

```
case(Museum~1, nom),
cmpd_form(Museum~1, Privatmuseum),
gend(Museum~1, neut),
mod(Museum~1, privat~1001),
mood(müssen~0, indicative),
num(Museum~1, sg),
oc_inf(müssen~0, weichen~3),
pers(Museum~1, 3),
sb(müssen~0, Museum~1),
sb(weichen~3, Museum~1),
tense(müssen~0, pres)
```

Figure 2: Dependency triple representation of TiGer Corpus sentence #8595

non-head within the compound, which turns out to be 1001 in the present example. The fact that the subject of the embedded verb *weichen* shares its structure with the subject of the top verb *müssen* is expressed by the two triples *sb(müssen∼0, Museum∼1)* and *sb(weichen∼3, Museum∼1)*.

The above example also demonstrates the lemmatization applied in the TiGer DB. Verb forms are lemmatized to the infinitive, nominal forms to the nominative singular etc. Compounds are split up into their components, of which the head is used in the predicate name and the others are *mod* dependents of this head. In order to keep track of the original compound form, the lemma of the compound is encoded as the value of the feature *cmpd_form*, as can be seen in Figure 2.

3.2 Grammatical relations

The most difficult decisions in creating the TiGer DB involve choosing the grammatical relations to be encoded. Which dependencies are needed in the final application differs from framework to framework, and the names they are given vary from grammar to grammar. As a guideline, it has been decided to stick to the functional annotation in the TiGer Treebank, i.e. the edge labels in the TiGer graphs. Additional distinctions were introduced where the TiGer Treebank annotations seemed not to make all the distinctions current deep parsers of German make. The most striking example of a TiGer Treebank edge label which is treated in a number of different ways by German parsers is *MO*, which can be a truly optional modifier (still labeled as *mo* in the TiGer DB), but also a predicative argument (labeled as *pd* in the TiGer DB) or a (more or less) obligatory directional or locative argument (labeled as *op_dir* and *op_loc* respectively). Finally, there were also some distinctions in the functional annotation of the TiGer Treebank which are generally not made by German parsers, so that some rather special functional labels have been abandoned in favor of more general ones. An example of such a relation is *AMS* (measure argument/adjunct of adjective) which is treated as a *mo* in the TiGer DB.

The grammatical relations that are encoded identically in the TiGer Treebank (or in a former version of it) and in the TiGer DB are:

- *sb* – subjects
- *oa* – direct objects in the accusative
- *oa2* – secondary objects in the accusative
- *da* – objects in the dative
- *og* – objects in the genitive
- *op* – prepositional objects
- *sbp* – logical subjects of verbs in the passive
- *cj* – conjunct of a coordination
- *gl* – genitive attribute on the left of its head noun
- *gr* – genitive attribute on the right of its head noun

- *pg* – *von*-PPs considered as pseudo-genitives
- *rc* – relative clauses
- *vo* – vocatives

Grammatical relations that can also be found in the TiGer Treebank, but whose definition diverges from it, are:

- *pd* – all predicative arguments, not only those of *bleiben*, *sein* and *werden*.
- *mo* – optional modifiers; in contrast to *MO* in the TiGer Treebank, *mo* in the TiGer DB no longer comprises (more or less) obligatory directional and locative arguments; the definition is enlarged with respect to the TiGer Treebank in that it now includes *AMS*s and certain *CC*s.
- *app* – close appositions, opposed to wide appositions in the TiGer Treebank; the latter are shifted to *mo*.
- *cc* – comparative (and equative) complements; in contrast to *CC* in the TiGer Treebank, this no longer comprises *wie*-PPs that are not triggered by an equative context; not being subcategorized, these are treated as *mo*s in the TiGer DB.
- *rs* – reported speech, in constructions such as *Technisch sei dies machbar, widersprach Starzacher den Skeptikern in der Verwaltung.*, where the *RS* clause is not regularly subcategorized for by the matrix verb; note that all other *RS* constructions of the TiGer Treebank are reinterpreted as *oc_fin*s.

Finally, there are a number of "new" grammatical functions in the TiGer DB, which arise from the more fine-grained distinctions that are made in the dependency bank with respect to the TiGer Treebank edge labels.

- *oc_fin* – finite clausal objects (*dass*/*ob*-clauses, indirect questions)
- *oc_inf* – infinite clausal objects
- *op_dir* – directional oblique arguments
- *op_loc* – locative oblique arguments
- *app_cl* – appositive clauses, occurring with *es* and pronominal adverbs
- *det* – articles, demonstrative and interrogative determiners
- *measured* – measured entity in constructions such as *zwei Flaschen Wein*
- *mod* – non-head components of compounds
- *number* – numbers in specifier position
- *poss* – possessive specifiers
- *quant* – quantifying specifiers

In general, determining these grammatical relations is relatively straight-forward. There are exceptions to this rule, however, such as the distinction between *mo*s and the different *op*s, as well as certain constructions where a given PP could be analyzed either as a *mo* or a *pd*.

3.3 Atomic features

The atomic features included in the TiGer DB correspond mostly to the morphological information encoded in the TiGer Treebank, but also to information from the part-of-speech tags. Generally these features further specify the predicates that relate to the terminal nodes in the TiGer Treebank where the information is encoded. An exception to this rule is the person/number agreement information given for finite verb forms, which ends up in the features *num* and *pers* of the subject of the verb in consideration, as well as agreement information provided by determiners and inflected adjectives, which is attached to their head noun. The purpose of this is to avoid the doubling of information.

It has been decided to encode the following atomic features in the TiGer DB: *case*, *cmpd_form* (cf. 3.1), *comp_form* (complementizer form), *coord_form* (form of coordinating conjunction), *degree*, *det_type* (type of determiner), *mood*, *num*, *pass_asp* (dynamic vs. stative passive), *pers*, *precoord_form* (first part of composite coordinating conjunction), *pron_type* (type of pronoun), *tense*.

4 Applications

The TiGer DB is designed as a gold standard for the dependency-based evaluation of German parsing systems. Since it uses the fairly theory-independent dependency structures (even though they are labelled), we expect them to be of use for a number of linguistic theories and, which is at least equally important, to allow cross-framework comparisons. For the moment, such a gold standard does not exist. It is thus of interest to parser and grammar developers belonging to different communities to have a common set of evaluation sentences for internal and perhaps even external comparison. Within both the HPSG and the LFG communities, tools and data sets for regression testing are already in common use for English and, to a lesser extent, for other languages (Oepen and Carroll, 2000).

Concerning the concrete possibilities of matching parser output against the TiGer DB dependency triples, we have considered this task both for an LFG and an HPSG parser.

Given the resemblance between the TiGer DB representations and f-structures, the first mapping (i.e. from LFG f-structures to dependency triples) turns out to be straight-forward. It involves some renaming and reorganizing of the structures, but since this is basically the same as is used in the final step of the construction of the TiGer DB (cf. 2.4), we do not expect major difficulties.

The mapping from HPSG feature structures to the TiGer DB is less trivial, since the representations differ more. Nevertheless, the TiGer DB is constructed in close collaboration with the HPSG developers at Saarland University, so that its usability by the HPSG community is ensured. Moreover, the mapping from HPSG feature structures to dependency triples will be part of the investigations carried out within the TiGer Project.

Concretely, the mapping from an HPSG analysis to a dependency structure requires the following steps: (i) the relations between the different semantic predicates in the sentence are maintained in the dependency structure, and (ii) the feature information encoded in TiGer DB is extracted from the HPSG analysis. As for (ii), the collection of information like case, mood, tense and so on is straight-forward as it is directly encoded in the grammar. As for the different arguments that verbs can take, we expect, if not a direct mapping between the argument labels in MRS and the ones in the dependency structure, a mapping with optional rules, defining the different constellations in which arguments can appear, to be possible.

For the use within the HPSG community, we hope to be able to use the dependency information that is encoded in the semantic representation of sentences (Minimal Recursion Semantics; MRS) (Copestake et al., 2001). It is a smaller structure than the full analysis, and can be used as a comparison between different HPSG grammars.

Depending on how large a share of the TiGer Corpus can be annotated with dependency triples, the TiGer DB can also be used for the construction of framework-specific training corpora for the probabilization of existing grammars.[3] This will be performed by matching the full analyses of these grammars against the dependency triples and determining the readings that are compatible with them. Given the linguistic detail of the TiGer DB, it is hoped that it allows to achieve so-called fully labeled data (opposed to partly labeled data that can be obtained with the help of less detailed annotation such as bracketing).

5 Conclusions

The TiGer Dependency Bank is a dependency bank containing both grammatical relations between predicates and arguments and a number of other grammatical features. In this it is closely related to the PARC 700 Dependency Bank (King et al., 2003). It is produced semi-automatically on the basis of the TiGer Treebank annotations, partly cross-validated by means of a broad-coverage LFG for German and finally validated by human annotators. The automation and cross-validation allow to build the TiGer Dependency Bank in a relatively short time and, more importantly, they help to obtain a degree of consistency of analysis otherwise impossible to achieve. Although the process still necessitates human intervention and validation, we hope to convert a large part (>2000 sentences) of the TiGer Treebank into a dependency bank by the time of the workshop.

The TiGer DB is mainly intended to be used for the evaluation of German parsers, both for regression tests during a specific grammar development effort and for parser comparison across frameworks. In this sense, it opens up new possibilities in the development of hand-crafted deep grammars. Moreover, since most statistical parsing systems for German (cf. Beil et al. (1999) and Frank (2002), for example) are based either on the TiGer Treebank or on the closely related NeGra Treebank, the TiGer DB can directly be used for a dependency-based evaluation of such parsing systems.

Furthermore, it is hoped that the TiGer DB will help to establish training corpora for the probabilization of grammars that to date are

[3]The difference with the work in the Redwoods project (HPSG) (Oepen et al., 2002) is that here it is intended to use the corpus annotations.

purely symbolic. Concretely, this is planned for the broad-coverage LFG developed at the University of Stuttgart along the lines of Riezler et al. (2002) and for the broad-coverage HPSG developed at Saarland University.

References

Anne Abeillé, editor. 2003. *Building and using syntactically annotated corpora.* Kluwer, Dordrecht.

Franz Beil, Glenn Carroll, Detlef Prescher, Stefan Riezler, and Mats Rooth. 1999. Inside-outside estimation of a lexicalised PCFG for German. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, Maryland.

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. 1999. Syntactic annotation of a German newspaper corpus. In *Proceedings of the ATALA Treebank Workshop*, pages 69–76, Paris, France.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. in press. TIGER: Linguistic Interpretation of a German Corpus. *Journal of Language and Computation.*

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC '99)*, Bergen.

John Carroll, Guido Minnen, and Edward Briscoe. 2003. Parser evaluation: using a grammatical relation annotation scheme. In Abeillé (Abeillé, 2003), pages 299–316.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse, France.

Stefanie Dipper. 2003. *Implementing and Documenting Large-scale Grammars – German LFG.* Ph.D. thesis, IMS, University of Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), Volume 9, Number 1.

Martin Forst. 2003a. Treebank Conversion – Creating a German f-structure bank from the TIGER Corpus. In *Proceedings of the LFG03 Conference*, Saratoga Springs. CSLI Publications.

Martin Forst. 2003b. Treebank Conversion – Establishing a testsuite for a broad-coverage LFG from the the TIGER Treebank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC '03)*, Budapest.

Anette Frank. 2002. Treebank Conversion. Converting the NEGRA treebank to an LTAG grammar. In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis*, Iasi.

Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ron M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC '03)*, Budapest.

John T. Maxwell III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571 – 589.

Stephan Oepen and John Carroll. 2000. Parser engineering and performance profiling. *Natural Language Engineering*, 6(1):81–97.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Chris Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank. Motivation and Preliminary Applications. In *Proceedings of 19th International Conference on Computational Linguistics, Coling 2002*, Taipei, Taiwan.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics 2002*, Philadelphia.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In *Proceedings of the Conference on Language Resources and Evaluation LREC-98*, pages 705–711, Granada, Spain.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2003. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, University of Tübingen, Tübingen, Germany.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, Heidelberg.