Treebank Data and Query Tools for Rare Syntactic Constructions

Erhard Hinrichs, Daniël de Kok and Çağrı Çöltekin

Department of Linguistics
University of Tübingen
E-mail: {erhard.hinrichs|daniel.de-kok|cagri.coeltekin}
@uni-tuebingen.de

Abstract

This paper reports on the use of the treebank query tool TüNDRA, which is able to process large treebanks of the size needed for rare syntactic constructions such as the Zwischenstellung of finite auxiliaries in German subordinate clauses. The TüPP-D/Z, an automatically annotated treebank with 11.5M sentences, contains a total of 92 examples of this construction, which need to be hand-filtered from corpus queries that produce a significant number of false positives. The corpus findings about the Zwischenstellung shed new light on the usage of this construction in contemporary German that contradict previous claims put forth in the linguistics literature.

1 Introduction

Treebanks serve a variety of purposes in computational linguistics – as training materials for statistical parsers and other automatic language processing tools – and in theoretical linguistics alike. For linguistic research, they provide authentic language materials for linguistic structure in general and (morpho-)syntax in particular. Authentic language materials present an important data type that can supplement grammaticality judgements of native speakers and that can provide valuable information about the actual usage patterns of linguistic constructions across speakers of a language.

The frequency of a particular grammatical phenomena under consideration determines the amount of corpus/treebank data that are necessary for a meaningful empirical investigation. If the phenomenon is relatively rare, then the amount of annotated data may have to be considerable and may go beyond what can reasonably be offered by treebanks such as the Penn Treebank (4.5 million English words;[10]) and the TüBa-D/Z (95.595 sentences with 1.787.801 German word tokens for Release 10.0 (08/2015);[17]) which were produced entirely by manual annotation. Rather, larger treebanks that were constructed semi-automatically or without any

manual post-editing such as the TüPP-D/Z [15] may need be to be consulted. The critical mass of data for a given grammatical phenomenon has repercussions not only for the method of annotation, but also for search interfaces that can be used to query treebanks. Most query tools currently only support treebanks up to a certain size, due to performance restrictions of the underlying search algorithms. In addition, since the treebank data are generated entirely by automatic means, the resulting data are noisy. This noisiness has to be taken into account when searching the treebank and when interpreting the results.

The purpose of the present paper is to investigate the so-called *Zwischenstellung* of finite auxiliaries in German as a case study of a low-frequency syntactic construction of German that requires large amounts of data and hence a highly performant query tool. The case study highlights: (i) the importance of verifying the claims that have been made in the linguistics literature about this construction by treebank data, and (ii) the processing requirements imposed on a treebank query tool that can accommodate the required amount of data. More specifically, the TüPP-D/Z will be used as the underlying treebank (see Section 3 below), whose annotations were produced by a finite-state chunk parser, and the TüNDRA [12] web application (see Section 4 below) will be used as the query tool of choice.

2 The Data: Placement of Finite Auxiliaries in German Subordinate Clauses

In subordinate clauses of German, finite verbs usually appear in clause-final position, as in (1a). However, when forms of the auxiliary verb *haben* govern a modal verb such *können* or *müssen* in (1b), then the auxiliary appears leftmost in the verbal complex in the so-called *Oberfeld* – in the terminology of Bech [2] – and the modal verbs are realized as so-called *Ersatzinfinitive* ('substitute infinitives'). The ungrammaticality of (1c) and (1d) show that Oberfeld placement and the use of the Ersatzinfinitiv (instead of the ordinary past participles) are obligatory.

(1) a. dass Eike gesungen hat.
that Eike sung has.
'that Eike has sung.'

b. dass Eike hat singen { können / müssen }.
that Eike has sing { can / must }.
'that Eike was able to / had to sing.'

c. * dass Eike singen { können / müssen } hat.
that Eike sing { can / must } has.

d. * dass Eike kommen { gekonnt / gemusst } hat.
that Eike come { can / must } has.

Examples (2) shows that Oberfeld placement is triggered not only by modal verbs, but also by the verb *lassen* ('let'). However, for *lassen*, clause-final placement and Oberfeld placement of the finite auxiliary are both acceptable, as are the

use of the past participle and the Ersatzinfinitiv for *lassen* in the case of clause-final placement of the auxiliary.

(2) dass sie ihn { arbeiten gelassen hat / arbeiten lassen hat / hat arbeiten
that she him { work let has / work let has / has work
lassen }.
let }.
'that she let him work.'

Oberfeld placement of finite auxiliaries is not restricted forms of *haben*, but also occurs with forms of *werden* 'will' in the future tense, as the examples in (3) show.

(3) a. dass sie { arbeiten können wird / wird arbeiten können }.
that she { work be able to will / will work be able to }.
'that she will be able to work.'

b. dass Eike hat singen { können / müssen }.
that Eike has sing { can / must }.
'that Eike was able to / had to sing.'

In examples (1) – (3), the finite auxiliary appears either in initial or final position in the verb cluster. Den Besten and Edmondson [4] have pointed out that there are also cases, where finite auxiliaries appear in the middle of the verbal complex in a so-called *Zwischenstellung* ('intermediate position'), i.e. to the right of the main verb and to the left of the non-finite auxiliary in examples such as (4).

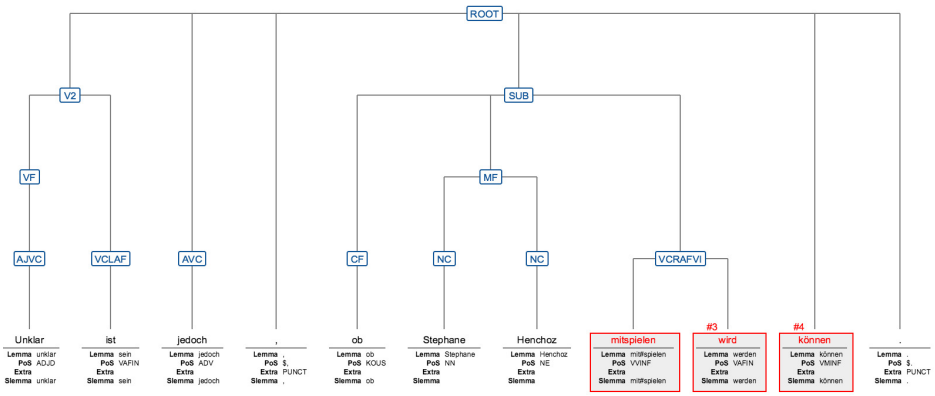
(4) a. dass er arbeiten hat können.
that he work has been able to
'that he has been able to work.'

b. dass er arbeiten wird können.
that he work will be able to
'that he will be able to work.'

c. dass er gewählt hätte werden können.
that he elected had [Passive werden] can.
'that he would have been able to be elected.'

d. dass er abgewählt wird werden können.
that he voted out will [Passive werden] can
'that he will possibly be voted out of office.'

For reasons of space, the data survey of Oberfeld placement of finite auxiliaries is far from complete. It covers only those triggering verbs that are directly relevant for the discussion of the Zwischenstellung in Section 5 below. A more comprehensive account of the Oberfeld is presented, inter alia, in [1], [2], and [5]. The grammaticality judgments on Oberfeld placement reported in this paper or taken



Unklar ist jedoch, ob Stephane Henchoz mitspielen wird können.
unclear is however whether Stephane Henchoz play with will can
'It is unclear, however, whether Stephane Henchoz will be able to play.'

Figure 1: TüPP-D/Z sentence with Zwischenstellung of *wird*

from [5]; however, see [7] for a dissenting view on the acceptability of Oberfeld formation with *werden* and *lassen*, as in (3).

3 The Corpus

The TüPP-D/Z (Tübingen Partially Parsed Corpus of Written German[1] treebank uses as its data source the Scientific Edition of the taz German daily newspaper[2], which includes articles from September 2, 1986 up to May 7, 1999. The corpus consists of 11,512,293 sentences with a total of 204,425,497 tokens. The texts are processed automatically, starting from paragraph, sentence, word form, and token segmentation. All sentences have been automatically annotated with clause structure, topological fields, and chunks, as well as parts of speech and morphological ambiguity classes. Figure 1 shows a sentence from the TüPP-D/Z with Zwischenstellung of the finite auxiliary *wird* ('will') in the verbal complex of a subordinate clause, which is headed by the clause label SUB. The subordinate clause and the main clause form the root clause (ROOT) and are annotated by topological field labels. Main clauses in German place the finite verb in second position; hence the clause label V2. Topological field annotation for main clauses include the Vorfeld (VF) and the left bracket (VCL) with the finite verb. Since the finite verb *ist* in Figure 1 is a finite auxiliary (FA), the left bracket is identified as VCLAF. The topological field structure of a German subordinate clause includes the complementizer field (CF) as the left bracket and the verbal complex as the right bracket (VCR). In Figure 1, the verbal complex is realized as a finite auxil-

[1] www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tuepp-dz.html
[2] www.taz.de

iary *wird* in Zwischenstellung and a non-finite main verb (VI). Accordingly, the verbal complex label is further specified as VCRAFVI. As will be described in Section 5, use to this topological field label in search queries for Zwischenstellung examples for the TüPP-D/Z.[3]

4 Querying Large Treebanks

TüNDRA [12] is a web applications that allows linguists to search and visualize treebanks. It uses the TIGERSearch [8] query language, with support for existential negation. Moreover, it supports both constituency trees and dependency graphs. Recently, the back-end of TüNDRA was rewritten to support large treebanks in the order of hundreds of millions words [3], such as used in this study.

4.1 Query Processing

Treebank search tools use a variety of different query engines and storage methods. Storage methods run the gamut from formats specific to treebank applications to generic graph databases. Specific storage formats provide more opportunity for optimization for the task at hand, whereas using a generic solution allows a treebank tool to leverage existing well-tested storage systems that typically support widely-used standards. We will give an example of both opposites of this gamut.

- INESS-Search [13] uses an on-disk format that is specifically developed for (directed graph) treebanks. It uses inverted indices for the features that are represented in the treebank (such as *word*, *cat*, *parent-edges*, and *child-edges*). The lexicon-part of the indices is stored as a suffix array [9], allowing for quick lookup of sentences and nodes using regular expressions. INESS-Search uses an extended version of the TIGERSearch query language. Queries are parsed to an internal representation that is similar to the logical form of the query. The inverted indices and relation/predicate signatures are used to restrict the set of candidate nodes. As Meurer [13] points out, the use of task-specific storage eliminates overhead, such as the use of transactions and locking, which is typically present in more generic databases.

- Dact [20] follows the exact opposite approach – it stores Alpino dependency structures as-is in a Berkeley DB XML database. Although the use of an XML database incurs some overhead, it makes the data queryable (XPath and XQuery) and processable (XSLT and XQuery) using W3C web standards. As a result, Dact can leverage XML technology extensively. It uses

[3] A more in-depth description of the linguistic annotation can be found in the TüPP-D/Z stylebook [15], and information about the actual XML encoding of linguistic annotation can be found in the TüPP-D/Z markup guide [18].

XPath (with support for macros) as its query language and heavily on XSLT for rendering and data export.

TüNDRA takes the latter route and uses BaseX [6] as its database backend. BaseX is a light-weight XML database that uses XQuery as its query language. To execute a query, TüNDRA's query processor first parses a TIGERSearch query into an intermediate representation, TIQR, that is amendable for query optimization [11]. The TIQR representation is then used to write the XQuery program that is executed by BaseX.

4.2 Motivation for Improving Scalability

TüNDRA relied on a couple of different techniques to make query processing performant. The BaseX database performs indexed queries on attributes of the elements that represent syntax tree nodes to restrict the set of nodes to be analyzed. Moreover, TIQR graph is processed such that attribute values that are infrequent are selected in XQuery before frequent attributes. Despite such optimizations, searching a treebank in TüNDRA could be slow. For example, consider the following query to select nodes (*d*) that dominate an *NX* immediately followed by *PX*:

(5) `#nx:[cat="NX"] . #px:[cat="PX"] & #d > #nx & #d > #px`

Processing such a query is relatively slow, because the (indexed) attributes select for a substantial number of trees (e.g. 75.3% of the sentences in TüBa-D/Z). Moreover, since two categories are considered to be adjacent in the TIGERSearch query language when their lexical nodes are adjacent, the query requires more structural matching than it may seem on the surface. While such a query takes tens of seconds on a treebank such as TüBa-D/Z (95000 sentences), under the assumption of linear scaling it would take hours to process on the TüPP-D/Z (11.5M sentences).

Even if long query processing times are a given, optimization of the user experience alleviates most of that problem. In our redesign of the TüNDRA backend, two principles guided this optimization: (1) the user should see the first query results within seconds. This is motivated by the observation that query formulation is typically an iterative process — a query is refined until it reflects exactly the phenomenon that a user is interested. If the time to the first result is to long, it interferes with this iterative refinement. (2) The user should be able to get intermediate statistics when the query is running. For many queries, one can already get a rough idea of the distribution of results when a fraction of a large treebank is processed. This allows the user to see if there are any interesting trends.

4.3 Architecture

As discussed in the previous sections, attribute indices form the backbone of speedy query processing. For this reason, XML databases generally load or map the in-

dexes into memory in order to process the query. It turns out that for large treebanks, this is the largest impediment to return results as early as possible [3], since the indices get paged out regularly. We solve this problem in TüNDRA by splitting the treebank in chunks that are small enough to make this loading time negligible for each chunk. To present the treebank as one single unit, each chunk is programmatically wrapped in a *multi-treebank*. This multi-treebank does the necessary translations to make it appear as a single treebank, such as: presenting iterators over results in all chunks, rewriting tree identifiers to monotonously increase, extracting/caching treebank metadata, and assuring that each chunk is of the same treebank type. Since our current implementation of the multi-treebank processes chunks sequentially, the mean time to the first match is roughly $E = \frac{t_c}{p_q}$ where t_c is the time to process a chunk and p_q the probability that a hit is found in a chunk for query q.

Another way TüNDRA provides immediate feedback to the user is by providing live query statistics. For instance, if the user executes the query of the previous section, they can view the distribution of the values that occurred for a particular attribute (for instance, *cat*). The statistics window is updated by executing the query asynchronously and updating the statistics window every n seconds. Unfortunately, we found that gathering statistics on large treebanks often resulted in copious memory use, since some queries can result in many distinct values.

For queries that result in an extremely large number of hits, we switch to reservoir sampling [19]. Reservoir sampling is an algorithm that is strongly related to Fisher-Yates shuffling for choosing k out of n items uniformly, where n is unknown beforehand. At each moment, the sample should be representative of query hits *thus far*, assuming that query match values are uniformly distributed across the corpus.[4] The statistics are updated when a match is replaced in the reservoir — the count for the replacee is decreased and that of the replacement increased.

5 Corpus Results on the Zwischenstellung

Table 1 summarizes the corpus results for the Zwischenstellung found in the TüPP-D/Z treebank. With a total of 92 occurrences in a corpus of 11,512,293 sentences, this phenomenon is, indeed, rare and hence requires large corpus resources of the kind used in the present study. The VVINF, VVPP, and VMINF part-of-speech tags in Table 1 are taken from the STTS tagset [16] for German and stand for main verb infinitive, main verb past participle, and modal auxiliary verb infinitive, respectively. While most of the corpus examples involve *können* and *müssen*, they also appear in the TüPP-D/Z with *sollen*, *wollen*, *dürfen*, and *mögen* the other four modal verbs subsumed under the part-of-speech tag VMINF.

The Zwischenstellung is often characterized as dialectal, especially attributed to southern varieties of German, and sometimes as archaic. Interestingly, the cor-

[4] This is a weakness in our current implementation. One possible solution is to shuffle the sentences before use.

pus findings in Table 1 do not confirm either of these claims. With more than 90 occurrences, the Zwischenstellung is well-attested in the TüPP-D/Z treebank. The regional character of the Zwischenstellung is also not confirmed by the TüPP-D/Z. The taz newspaper used for the TüPP-D/Z treebank is published in Berlin, and the particular local taz issue used for the treebank is the Bremen taz edition. While it is not a foregone conclusion that the journalists are from this northern area only, it is highly unlikely that they are all speakers of southern varieties of German.

Linguistic Pattern	Avg. occurrences per 1 million tokens	Raw Corpus frequencies
VVINF ***haben*** VMINF	0.07	15
VVINF ***werden*** VMINF	0.15	30
VVINF ***haben*** VMINF	0.02	4
VVINF ***werden*** VMINF	0.15	26
VVINF ***haben*** *lassen*	0.05	11
VVINF ***werden*** *lassen*	0.01	1
VVPP ***haben*** werden VMINF	0.03	5

Table 1: Zwischenstellung of *haben* and *werden*

The examples in (6) are taken from the TüPP-D/Z treebank. They illustrate each of the seven linguistic pattern listed in Table 1.

(6) a. daß er von Wahlfälschungen nichts wissen habe können.
that he of election fraud nothing known has been able to
'that could know anything about election fraud.'

b. wegen dem der Strauß 62 gehen hat müssen.
because of which the Strauß 62 leave has had to
'because of which Mr. Strauss had to leave in 1962.'

c. ob die sich in der neuen Hochblüte des Kapitalismus
whether they self in the new hayday of capitalism
halten werden können.
keep will be able to
'whether they will be able to persist in this new hayday of capitalism'

d. daß sie so lange Haftstrafen absitzen werden müssen.
that they such long prison terms serve will have to.
'that they will have to serve such long prion terms.'

e. die man laufen hat lassen.
which one walk has let
'which one has let go.'

f. die ... uns lange vor unserer Hybris erzittern werden
which ... us long due to our arrogance tremble will
lassen.
let
'which will let us tremble on account of our arrogance.'

g. deren Zustimmung eingeholt hätte werden müssen.
whose consent sought had [Passive werden] have to
'whose consent would have to have been sought.'

Interestingly, the Zwischenstellung occurs also among the 4-element verbal clusters. One possible language-processing explanation for this finding may be that the Zwischenstellung offers an effective way to separate the full verb from the other (auxiliary) verb members of the verb cluster. For the 4-element verbal clusters, with three auxiliaries following the main verb, this clear separation may well facilitate language comprehension and production.

The TüNDRA search queries used to extract instances of the Zwischenstellung from the TüPP-D/Z require reference to the syntactic annotation of the treebank, in particular to the layer of topological field annotation, and reference to the layer of morpho-syntactic part-of-speech annotation.

(7) a. [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN"
& lemma = /haben|werden/] & #2 . #3:[pos="VMINF"
& lemma=/müssen|können|dürfen|wollen|sollen|mögen/]

b. [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN"
& lemma = /haben|werden/] & #2 . #3:[pos="VVINF"
& lemma=/lassen/]

The first terms in the two TüNDRA search queries in (7) use the topological field label VCRAFVI (short for: right-bracket verbal complex (VCR_) with finite auxiliary (_AF) and infinite verb (_VI) and, thus, suitably restrict the search to subordinate clauses. The > operator stands for immediate dominance, and the dot operator (.) for immediate precedence.

Simpler queries that search for sequences of part-of-speech labels and lemmas and that do not include topological field information, as in (8), lack the required accuracy.

(8) #1:[pos="VVINF"] . #2:[lemma=/haben|werden/]
& #2 . #3:[lemma=/müssen|können|dürfen|wollen|sollen|mögen/lassen]

They retrieve as false positives sentences as in (9), where the sequence of matching lexical tokens for query (8) are identified in (9) by corresponding numerical subscripts. The lexical tokens #1 and #2 matching the query do not belong to a single topological field, as is required by query (7a), but straddle the left bracket (VCL) of a main clause and the Vorfeld (VF) and left bracket of a main clause in (9a); hence they do not constitute examples of the Zwischenstellung of the finite

auxiliary *wird*. (9b) is not admitted by query (7a), since the auxiliary *kann* does not match the part-of-speech tag VMINF.

(9) a. Einziehen$_1$ wird$_2$ dürfen$_3$, wer dringend ein Dach
move in will be allowed who urgently a roof over
über dem Kopf braucht.
the head needs.
'They will be allowed to move who urgently need a place to stay.'

b. Welche Schlüsse Milosevic daraus ziehen$_1$ wird$_2$ kann$_3$
which conclusions Milosevic from that draw will can
noch niemand voraussagen.
so far nobody predict.
'Which conclusions Milosevic will draw from that so far nobody can predict.'

While TüNDRA search query (7a), which includes topological field information, is highly accurate, it does not succeed in retrieving all cases of the Zwischenstellung construction contained in the TüPP-D/Z treebank. This is due to annotation errors in the treebank data that arise from automatic annotation of the data source. Such annotation errors often orginate at the level of part-of-tagging. For the construction at hand, auxiliaries such as *haben* and *können*, where the finite and non-finite forms coincide, are often mistagged. In order to retrieve examples of the Zwischenstellung for which finite auxiliaries have been mistagged as non-finite (VAINF), the queries in (10) are necessary.

(10) a.
```
[cat="VCRVI"] > #1:[ pos="VVINF"] &
#1 . #2:[pos="VAINF" & lemma = /werden|haben/] &
#2 . #3:[lemma=/müssen|können|dürfen|wollen|sollen|mögen/]
```

b.
```
[cat="VCRVI"] > #1:[ pos="VVINF"] &
#1 . #2:[pos="VAINF" & word=/haben|werden/] &
#2 . #3:[lemma=/lassen/]
```

These queries refer to the same topological field of a right-bracket verbal complex (VCR) that is also included in queries (7). But the queries (10) use a different suffix (_VI) for this topological field since the field contains only non-finite verbs.

While queries (10) lead to the required recall for examples of the Zwischenstellung with mistagged POS tags, they lack precision since they admit a number of false positives. The subscripts in (11) match the hash tags in query (10).

(11) Es wird bereits eine Denkpause gefordert, wie mit den
it werden passive already a moratorium demanded, how with the
Unterlagen der Staatssicherheit weiter verfahren$_1$ werden$_2$
documents of the secret police further proceeded werden passive
soll$_3$.
should .

'A moratorium has already been demanded how best to proceed with the documents of the secret police.'

Such false positives include examples such as (11), where the participial and infinitival forms of main verbs coincide, as is the case for the verb *verfahren*. (11) is actually an example of an impersonal passive with *werden* as a passive marker, rather than an instance of *werden* in Zwischenstellung.

In sum, the TüNDRA queries used to extract instances of the Zwischenstellung from the TüPP-D/Z treebank need to be hand-filtered since they are inevitably noisy. This noisiness is due to two main factors: (i) annotation errors in the treebank data that arise from automatic annotation of the data source, and (ii) the imprecision of the queries themselves, which also yield instances of other syntactic constructions, in particular of passive sentences. The manual filtering of such false positives is greatly facilitated by the incremental presentation of query results in TüNDRA described in Section 4 above.

6 Conclusion and Outlook

This paper has reported on the use of the treebank query tool TüNDRA, which is able to process large treebanks of the size needed for rare syntactic constructions such as the Zwischenstellung of finite auxiliaries in German subordinate clauses. The corpus findings about the Zwischenstellung shed new light on the usage of this construction in contemporary German that contradict previous claims contained in the linguistics literature.

The TüPP-D/Z, an automatically annotated treebank with 11.5M sentences, contains a total of 92 examples of this construction, which need to be handfiltered from corpus queries that produce a significant number of false positives. The noisiness of automatically annotated data, incidentally looking at the Zwischenstellung as one of the syntactic constructions under consideration, is addressed also in some detail in [14], whose observations and conclusions are largely confirmed by the present corpus study.

At present, the burden is on the users of the TüNDRA tool to overcome noisiness of annotation by refining their queries in the appropriate way. In future work, it would be interesting to explore to what extent such query refinements can be guided by the tool itself.

7 Acknowlegdements

The authors gratefully acknowledge the financial support of their research by the German Ministry for Education and Research (BMBF) as part of the CLARIN-D research infrastructure grant given to the University of Tübingen.

References

[1] John Ole Askedal. "Ersatzinfinitiv/Partizipialsatz" und Verwandtes: Zum Aufbau des verbalen Schlussfeldes in der modernen deutschen Standardsprache. *Zeitschrift für germanistische Linguistik*, 19, 1–23, 1991.

[2] Gunnar Bech. *Studien über das deutsche verbum infinitum*. Dan. Hist. Filol. Medd. Bind 35, no. 2 (1955) & Bind 36, no. 6 (1957). Det Kongeliege Danske Videnskabers Selskab, 1955/57.

[3] Daniël de Kok, Dörte de Kok, and Marie Hinrichs. Build your own treebank. In *Proceedings of the CLARIN Annual Conference*, Volume 2014, 2014.

[4] Hans den Besten and J. Edmondson. The verbal complex in continental West Germanic. In Werner Abraham, editor, *On the Formal Syntax of the Westgermania. Papers from the '3rd Groningen Grammar Talks'*. John Benjamins Publishing Company, Amsterdam/Philadelphia, 155–216, 1983.

[5] Peter Eisenberg, G. Smith, and O. Teuber. Ersatzinfinitiv und Oberfeld – ein großes Rätsel der deutschen Syntax. *Deutsche Sprache*, 29(1):242–260, 2001.

[6] Christian Grün, Alexander Holupirek, and Marc H Scholl. Visually exploring and querying XML with BaseX. In *BTW*, 103, 629–632, 2007.

[7] John Evert Härd. *Studien zur Struktur mehrgliedriger deutscher Nebensatzprädikate. Diachronie und Synchronie*. Number 21 in Göteborger Germanistische Forschungen. Göteborg, 1981.

[8] Wolfgang Lezius. TIGERSearch ein Suchwerkzeug für Baumbanken. *Tagungsband zur Konvens*, 2002.

[9] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[10] Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313–330, 1993.

[11] Scott Martens. Tündra: Tigersearch-style treebank querying as an xquery-based web service. In *Proceedings of the joint CLARIN-D/DARIAH Workshop "Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts"(DH 2012)*, 41–50, 2012.

[12] Scott Martens. TüNDRA: A web application for treebank search and visualization. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, 133–144, 2013.

[13] Paul Meurer, Miriam Butt, and Tracy Holloway King. Iness-search: A search system for LFG (and other) treebanks. In *Proceedings of the LFG'12 Conference*, 404–421, 2012.

[14] Walt Detmar Meurers. On the use of electronic corpora for theoretical linguistics. Case studies from the syntax of German. *Lingua*, 115 (11), 1619–1639, 2005.

[15] Frank Henrik Müller. Stylebook for the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2004.

[16] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset). Technical report, Universität Stuttgart, Universität Tübingen, Tübingen, Germany, 1999.

[17] Heike Telljohann, Erhard Hinrichs, Heike Zinsmeister, and Kathrin Beck. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2015.

[18] Tylman Ule. Markup Manual for the Tübingen Partially Parsed Corpus of Written German (tüpp-d/z). Technical report, University of Tübingen, Seminar für Sprachwissenschaft, 2004.

[19] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[20] Gertjan Van Noord, Gosse Bouma, Frank Van Eynde, Daniel De Kok, Jelmer Van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. Large scale syntactic annotation of written Dutch: Lassy. In *Essential Speech and Language Technology for Dutch*, 147–164. Springer, 2013.