German and English Treebanks and Lexica for Tree-Adjoining Grammars

Miriam Kaeshammer, Vera Demberg

Institut für Sprache und Information, Department of Computational Linguistics
University of Düsseldorf, Saarland University
kaeshammer@phil.uni-duesseldorf.de, vera@coli.uni-sb.de

Abstract

We present a treebank and lexicon for German and English, which have been developed for PLTAG parsing. PLTAG is a psycholinguistically motivated, incremental version of tree-adjoining grammar (TAG). The resources are however also applicable to parsing with other variants of TAG. The German PLTAG resources are based on the TIGER corpus and, to the best of our knowledge, constitute the first scalable German TAG grammar. The English PLTAG resources go beyond existing resources in that they include the NP annotation by (Vadas and Curran, 2007), and include the prediction lexicon necessary for PLTAG.

Keywords: grammar extraction, tree-adjoining grammar, incremental processing

1. Introduction

Grammars play a key role in natural language modelling and processing. While they have been created by hand traditionally, the availability of annotated resources such as treebanks has rendered it possible to automatically derive wide-coverage grammars for various formalisms, for example CFG (Charniak, 1996), LTAG (Xia et al., 2000) and LFG (Cahill, 2004) to name just a few approaches. Treebank grammars furthermore have the crucial advantage of holding statistical information that is necessary to train the parameters of stochastic parsers.

The treebanks and lexica presented in this paper were developed for a recent version of TAG, called "PsychoLinguistically motivated Tree-Adjoining Grammar" (PLTAG, Demberg and Keller (2008)). Recent psycholinguistic research suggests that humans process sentences in a strictly incremental fashion (Tanenhaus et al., 1995; Konieczny, 2000), integrating incoming words eagerly with the incremental analysis (Sturt and Lombardo, 2005), and make predictions about upcoming structure and lexemes (Kamide et al., 2003; Staub and Clifton, 2006; van Berkum et al., 1999).

PLTAG and standard LTAG generate the same derived trees, and the PLTAG grammar is a superset of a standard LTAG grammar: in addition to the standard initial trees and auxiliary trees of an LTAG grammar, it includes unlexicalized so-called prediction trees, which are necessary in order to make explicit predictions about upcoming material in a sentence. The German and English PLTAG resources, which we present in this paper[1], are hence also useful for other variants of TAG. We induce them from the TIGER Treebank (Brants et al., 2002) and the Penn Treebank (PTB, Marcus et al. (1993)) respectively. The two main steps are (1) to convert the specific treebank format to (P)LTAG format and (2) to extract canonical elementary trees as well as prediction trees.

The created (P)LTAG resources are of interest to several fields for various reasons: First of all, to the best of our knowledge, no broad-coverage lexicalized tree-adjoining grammar (and treebank) for German is currently available. The induced lexicon together with the converted treebank will close this gap. Furthermore, PLTAG can be combined with a sentence processing theory (Demberg-Winterfors, 2010) that models human processing difficulties. The theory has already been validated for English (Demberg-Winterfors, 2010) with a PLTAG parser based on the lexicon and the treebank we are presenting here, but, from a psycholinguistic view, evaluation for other languages should follow. Finally, the resources can also play a role in language technology applications (e.g. dialogue systems) in the future, in which a PLTAG parser can be used to determine processing difficulties, for example in order to optimize machine-generated text.

2. The Formalism

Tree-adjoining grammar is a linguistically inspired tree-rewriting formalism introduced by (Joshi et al., 1975).

2.1. LTAG

The primitive elements of a lexicalized TAG (LTAG) are elementary trees which have at least one lexical anchor. They are divided into initial trees and recursive auxiliary trees with a unique foot node (marked with $*$) that has the same label as the root node. The trees (a)-(c) in Figure 1 are examples. Elementary trees are combined via *substitution* and *adjunction* operations to build derived trees. Initial trees can substitute into substitution nodes (marked with $\downarrow$), while auxiliary trees adjoin to a node of the partially derived tree. In doing so, the daughter nodes of the adjunction site become daughters of the foot node. Both operations are constrained by the label of the nodes.

A non-standard TAG operation is *sister-adjunction*, introduced by (Chiang, 2000). In sister-adjunction, the root node of an inital tree is added as a new daughter to any other node.

[1] The PLTAG treebank and lexicon for English and German can be downloaded from `http://www.coli.uni-saarland.de/~vera/`, if you have a license for the original PTB and TIGER corpora.

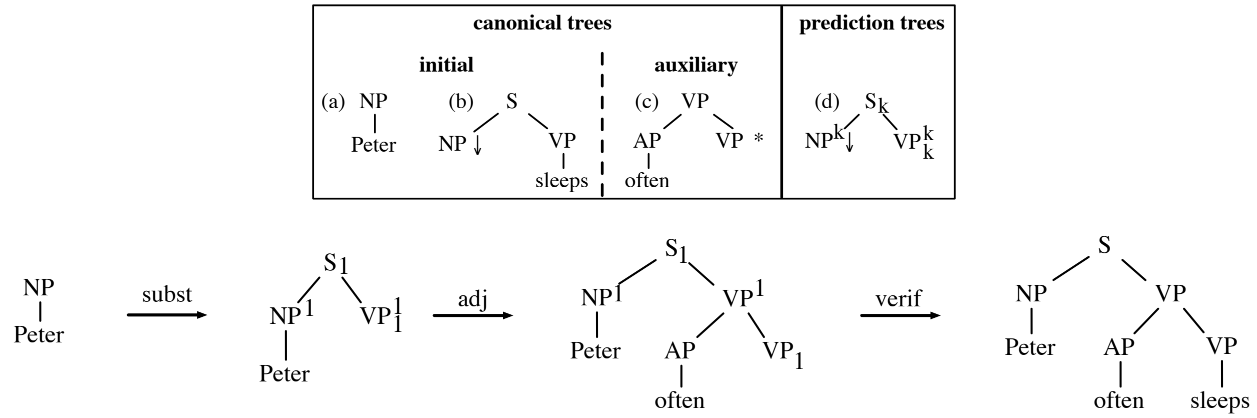


Figure 1: PLTAG lexicon and incremental derivation.

2.2. PLTAG

LTAG with its standard linguistically motivated elementary trees (see for example, the XTAG grammar (XTAG Research Group, 2001)) is not guaranteed to be able to derive a sentence strictly incrementally. For instance, when deriving *Peter often sleeps* with the trees (a)-(c) in Figure 1, the elementary trees for *Peter* and *often* cannot be combined because the VP node necessary for adjunction of (c) has not been derived yet. PLTAG therefore extends LTAG in that it specifies not only a lexicon of canonical lexicalized initial and auxiliary trees, but also a predictive lexicon which contains potentially unlexicalized *prediction trees*. In order to distinguish predicted nodes from canonical nodes, all nodes of a prediction tree have *markers*, see the superscript and/or subscript on the nodes of tree (d) in Figure 1 as an example. Similarly to the features in feature-based TAG, substitution nodes and foot nodes only have superscripts, and root nodes only have subscripts, while internal nodes have both. This is because root, foot and substitution nodes are conceptualized as incomplete (upper / lower halves of) nodes which will be completed when elementary trees combine through adjunction or substitution. Consider for example the substitution and adjunction operations in Figure 1. Substituting the elementary tree for *Peter* into the predicted substitution node NP^1 leads to a complete node for which we have partial evidence (i.e. we have observed the lower half but only predicted the upper half). Adjoining into a node that carries markers (like the VP^1_1 node) pushes the two markers apart. The upper marker becomes the upper marker of the root of the auxiliary tree, whereas the lower marker becomes the lower marker of the foot node (see the second step of Fig. 1). Note that if a prediction tree is adjoined into a node that already carries markers, this may create nodes that have an upper and lower marker with different values.

Markers are eliminated from a partial derived tree through a new operation called *verification*. Recall that markers indicate nodes that were only predicted during the derivation, without having been introduced by a word that was actually observed so far. The verification operation removes these markers by matching them with the nodes of the canonical elementary tree for a word in the sentence. Consider the last derivation step in Figure 1. This is a verification step for the marker 1, using the canonical tree for *sleeps* as the *verification tree*. The verification tree has to match the predicted tree in shape (i.e. the verification tree must contain all nodes with the same prediction marker, and in the same order; additional nodes in the verification tree can only be at the bottom of its spine[2] or to the right of its spine – otherwise incrementality would be violated). A valid PLTAG derived tree may not contain nodes with prediction markers.

PLTAG otherwise allows the same operations (adjunction and substitution) as standard LTAG, with the difference that they can also be applied to prediction trees. Since the verification does not introduce new tree configurations that would not be allowed in standard LTAG, and no prediction markers are allowed in the final derived tree, PLTAG generates the same derived trees as a corresponding LTAG. Even though the PLTAG formalism does not impose constraints on the shape of the prediction trees, it only makes sense to include those prediction trees which are the same or smaller than some canonical elementary tree due to the verification operation. The optimal granularity of predictions and the desired level of generalisation is an open research question. Demberg-Winterfors (2010) decides for minimal predictions that only predict upcoming structures as far as needed for full connectivity or subcategorization. Prediction trees are therefore defined as having the same shape as the canonical elementary trees, except that they do not have nodes to the right of the spine and that unary nodes at the bottom of the spine, including the lexical item, are removed. For auxiliary trees, the foot node is also included in the prediction tree.

PLTAG furthermore emphasises the use of multi-anchored elementary trees as a means for prediction at the lexical level in strong collocations such as "either..or" or idioms, for which there is psycholinguistic evidence (Staub and Clifton, 2006; Tabossi et al., 2005).

3. Treebank Conversion

In this section we describe the steps that are taken to convert the specific treebank formats to (P)LTAG derived trees. Crucially, this also includes introducing linguistic generalisations that are missing in the original treebank. Our over-

[2] The spine is the path from the root to the lexical anchor, which usually is the linguistic head.

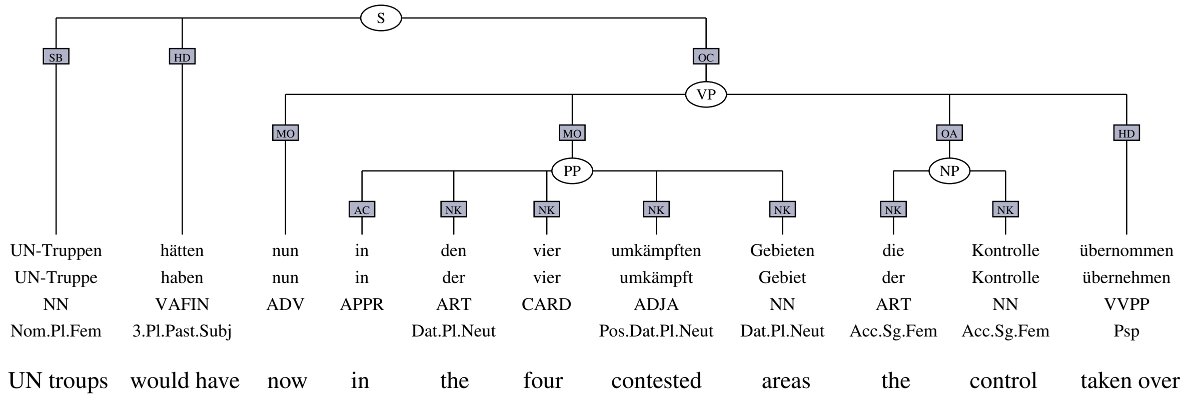

S
SB
HD
OC
VP
MO
MO
OA
HD
PP
NP
AC
NK
NK
NK
NK
NK
NK
UN-Truppen UN-Truppe NN Nom.Pl.Fem
hätten haben VAFIN 3.Pl.Past.Subj
nun nun ADV
in in APPR
den der ART Dat.Pl.Neut
vier vier CARD
umkämpften umkämpft ADJA Pos.Dat.Pl.Neut
Gebieten Gebiet NN Dat.Pl.Neut
die der ART Acc.Sg.Fem
Kontrolle Kontrolle NN Acc.Sg.Fem
übernommen übernehmen VVPP Psp
UN troups
would have
now
in
the
four
contested
areas
the
control
taken over

Figure 2: TIGER sentence: *UN troops have now taken control of the four contested areas.*

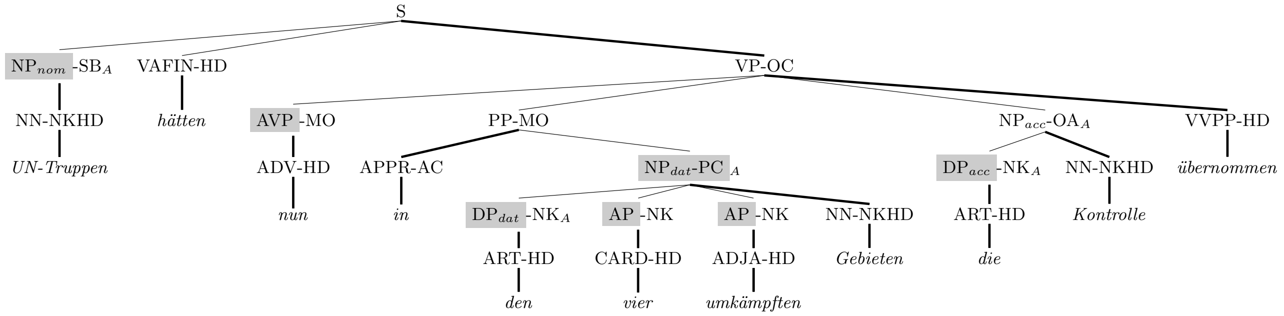


Figure 3: TIGER sentence after conversion (with head paths and argument nodes being marked)

all goal is to be able to induce a linguistically sound grammar with maximal generalisation capacity with respect to unseen data. For more details, see Demberg-Winterfors (2010, PTB) and Kaeshammer (2012, TIGER).

3.1. English Penn Treebank

The necessity to convert the PTB arises from the fact that its relatively flat structures often do not allow for modifiers to be adjoined into the trees with the standard TAG adjunction operation. The first step of the conversion algorithm is to add noun phrase annotation created by (Vadas and Curran, 2007) to the PTB. We then remove quotation marks, brackets, sentence-final punctuation and some of the traces from the PTB (for more details see Demberg-Winterfors (2010)). Next, additional structure is heuristically inserted to disambiguate the flat quantifier phrases. We also insert explicit right branching and additional nodes wherever adjunction is possible. Furthermore, auxiliaries are assigned a special POS tag in order to enable the lexicon induction algorithm (Section 4.) to extract them as auxiliary trees. Deviating from the XTAG analysis, copula verbs are treated in the English PLTAG as subcategorizing for two NPs and are therefore marked during treebank conversion.

3.2. German TIGER Treebank

Apart from phrase structure information (circled labels in Fig. 2), the graphs in the TIGER Treebank also express syntactic functions e.g. head (HD), subject (SB), accusative object (OA), modifier (MO) (grey boxes in Fig. 2). The leaf nodes additionally carry morphological information and lemmata.

Nominal Case Marking The TIGER Treebank uses flat syntactic representations as well, but most strikingly in comparison to the PTB, the subject and the finite verb are always immediate daughters of the sentence node S, see Figure 2. Only non-finite verbs project to VP. In this way, the annotation scheme accounts for the relatively free word order of German. Since argument roles are determined by morphological case rather than by syntactic position, the German PLTAG treebank includes case annotation for NPs. During treebank conversion, information about case is obtained bottom-up from the morphological layer of TIGER, or, if underspecified there, top-down from the syntactic functions. Since determiners often disambiguate the case of a noun phrase, we also mark DPs for case.[3] The PLTAG treebank also contains the lemma information for each lexical anchor.

Linguistic Generalisations Other peculiarities of the TIGER annotation scheme are that prepositional phrases do not embed an NP (see the PP in Fig. 2), and that there are almost no unary productions. Categories do not have a maximal projection unless they have their own dependents (cf. the NN of *UN-Truppen* in Fig. 2 which does not project to an NP while *Kontrolle* does).

Restructuring is necessary in order to obtain a modular grammar with good coverage on unseen data: We introduce an NP complement for adpositions, and complete the

[3] Attributive adjectives also agree in case with the noun they modify. The conversion procedure can be easily extended to provide them with case annotation as well. However, as modifiers, their role is less important, and we anticipate data-sparsity issues.

annotation with phrasal projections for all nouns, determiners, adjectives, adverbs and verbs (see the shaded nodes in Fig. 3). The overall result are more uniform tree structures. Since the head of noun phrases is not explicitly marked in the annotation (see the NPs in Fig. 2), we identify it with the help of the part-of-speech labels and provide a corresponding syntactic function NKHD, which will be used by the subsequent extraction procedure.

Sister-Adjunction Because of the relatively free word order in German where modifiers and arguments can occur in almost any order, we decided to use sister-adjunction (Chiang, 2000) and retain the flat TIGER annotation for the German PLTAG.

In contrast to Chiang (2000), we constrain sister-adjunction by the category of the node to which a tree can sister-adjoin. Besides initial, auxiliary and prediction trees, the German PLTAG therefore comes with an additional set of *modifier trees* M whose root node is required to have exactly one daughter. We mark them with an asterisk on the root node. When some $\gamma \in M$ sister-adjoins at position i to a node n, the root of γ and n must have the same label. The only daughter of γ's root is then added as a daughter of n in the same way as defined by (Chiang, 2000).

Besides modification, finite auxiliary and modal verbs are also encoded as sister-adjoining modifier trees. This enables direct modelling of the flat sentence structure without introducing additional levels.

Predicative Auxiliary Trees The annotation of raising and control verbs in the TIGER treebank (see *hofften* in Fig. 4) requires the introduction of a new S node in order to encode raising and control verbs as auxiliary trees (see node S_3 in Fig. 5(a)). Non-finite auxiliary and modal verbs can be analysed as VP auxiliary trees without further conversion.

Discontinuous Constituents To express long-distance relations, such as extraposed relative clauses, appositions, topicalized objects and repeated elements, the TIGER Treebank uses crossing branches, which occur in almost 30% of all sentences. Figure 4 shows an example. Since neither CFG nor TAG can directly encode crossing branches, the standard procedure in data-driven parsing[4] is to convert the graphs with crossing branches to trees (with indexed traces) by re-attaching non-head daughters of the discontinuous constituents higher.[5] The consequence of this reattachment is that the dependency information contained in the phrase-structure tree is changed, and that a trace and its filler can only be correctly interpreted if they co-occur in the final derived tree.

In contrast to CFG which corresponds to trees of depth 1, TAG with its extended domain of locality can localize some of the trace-filler pairs within one elementary tree. Given our analysis of modal and auxiliary verbs, this is for example the case for arguments of the (non-finite) full verb in compound tenses. To be able to generally describe scrambling of one argument of an embedded verb into the matrix clause[6], an additional S level has to be inserted to which the argument filler is re-attached. This is illustrated in Figure 5: The fronted object of the embedded VP is attached to S_2 with the standard procedure to resolve crossing branches, i.e. the substitution node corresponding to the filler would end up in the elementary tree of *hofften*, which violates the co-occurrence principle that is generally accepted for natural language TAG. However, given our analysis of control verbs as auxiliary trees and the additional S_1 node, the trace-filler dependency is localized in the elementary tree of *erfahren*. Ca. 80% of the argument trace-filler pairs can be captured by our lexicon.

However, about 90% of the discontinuous constituents in the TIGER Treebank are caused by modifiers, which cannot be localized within an elementary tree. (They could be extracted as tree sets, as suggested for English by (Xia, 2001), but the required variant of multi-component TAG would be non-local.)

Miscellaneous The TIGER annotation employs specific labels for coordinated categories, such as CNP for coordinated noun phrases. In order to provide the recursion that is necessary to encode coordination structures as auxiliary trees that adjoin to the first conjunct while providing a substitution node for the second conjunct, we turn the labels into their non-coordinated counterpart (e.g. CNP to NP). Punctuation marks are kept and generally encoded as modifier trees. Some also anchor trees with more structure, for example for a coordination.

3.3. Conversion statistics

The transformations for the English PTB increase the number of nodes in the treebank by 22%, that is on average 9.5 nodes per tree. On the German TIGER treebank, the presented transformations insert 6.5 nodes per tree on average, which increases the size of the treebank in terms of nodes by almost 25%. Note the different nature of the introduced nodes: while in the English treebank the majority of nodes are inserted for recursion, in the German treebank linguistic generalisations make up for most of the new nodes. The transformations are reversible. A parser based on the German PLTAG can thus be evaluated with respect to the original treebank for a better comparability with other parsers.

4. Lexicon Induction

After converting the PTB and the TIGER Treebank into PLTAG format, they can be used to induce a PLTAG lexicon, namely a canonical LTAG lexicon and the prediction lexicon.

4.1. Canonical Elementary Trees

The PLTAG canonical trees are extracted from the converted treebanks following the procedures described in (Xia et al., 2000).

We augment the English Penn Treebank with information on syntactic heads based on a slightly modified version

[4] An exception is parsing with more powerful grammar formalisms, such as LCFRS, e.g. (Maier, 2010), which are rarely used to date in NLP, mostly because of the prohibitive parsing complexity.

[5] We modified a script by Michael Schiehlen which was originally written for the NeGra corpus.

[6] TAG is not powerful enough to describe scrambling of more arguments in a linguistically adequate way (Becker et al., 1991).

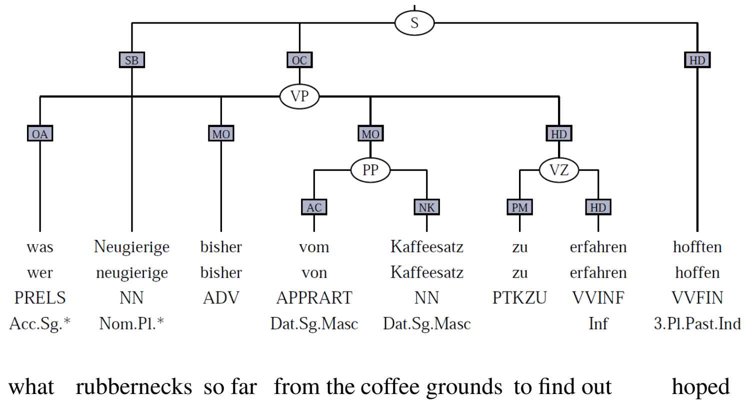
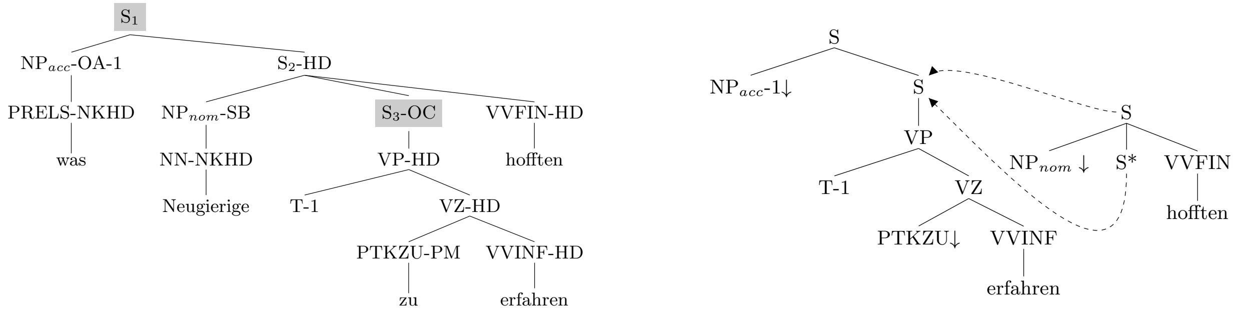

S
SB
OC
HD
VP
OA
MO
MO
HD
PP
VZ
AC
NK
PM
HD
was wer PRELS Acc.Sg.*
Neugierige neugierige NN Nom.Pl.*
bisher bisher ADV
vom von APPRART Dat.Sg.Masc
Kaffeesatz Kaffeesatz NN Dat.Sg.Masc
zu zu PTKZU
erfahren erfahren VVINF Inf
hofften hoffen VVFIN 3.Pl.Past.Ind
what rubbernecks so far from the coffee grounds to find out hoped

Figure 4: Tiger graph with a discontinuous constituent because of an extracted object: *what rubbernecks hoped to find out from reading tea leaves so far*



(a) Converted tree, omitting the two modifiers (The subscripts are used to be able to refer to the individual nodes in the text, they are not part of the alphabet.)

(b) Trace and filler are localized in one elementary tree, but will be further apart when the predicative auxiliary tree adjoins.

Figure 5: Localizing the long-distance relation in Figure 4

of (Magerman, 1994)'s head percolation table, in combination with more detailed heuristics for noun phrases (the head percolation table and code for NP heuristics are available at `http://www.coli.uni-saarland.de/~vera/page.php?id=corpora`). As a next step, subcategorization information from Propbank (Palmer et al., 2005) is added in, providing information about argument and modifier status, and encoding which lexical items should be part of the same elementary tree (currently, this is restricted to particle verbs like *show up* and some hand-coded constructions in which the first part is very predictive of the second part, such as *either..or*). For German, we infer the head-argument-modifier classification from the function labels annotated in TIGER. They also provide indicators concerning which lexical items should form a multi-anchored tree (particle verbs, collocational verb constructions, circumpositions, correlative conjunctions).

Elementary trees are determined by identifying the path from each lexeme up towards the root of the tree, proceeding as long as the node is the head child of its parent. When a node is its parent's argument, a substitution site is created and the elementary tree is encoded as an initial tree. Modifiers are encoded as modifier trees in the German lexicon, and as auxiliary trees in the English lexicon. The root and foot of an auxiliary tree are provided by the parent and the head sibling of the modifier node respectively. In Figure 3, the head paths are indicated by thicker edges and argument nodes are denoted by the subscript $_A$. Some of the extracted elementary trees are shown in Figure 6.

Recursive coordinating structures result in auxiliary trees anchored in the conjunction. After the core procedure, elementary trees of lexical items that are marked for constituting a multi-anchored tree are assembled. Figure 7 depicts examples.

The predicative auxiliary trees in the German PLTAG are generated following ideas from (Chen and Shanker, 2004). If a node n is an argument with respect to the spine ϕ, it is directly dominated by a node on ϕ and, there is node n' on ϕ dominating n, which has the same label as n, the elementary tree which corresponds to ϕ is excised as an auxiliary tree where n' is its root and n its foot. To avoid mal-formed elementary trees, we restrict n to be a clausal object (syntactic funtion OC). Figure 5 shows an example: S$_3$ is an argument with respect to $\phi_{hofften} = \{$VVFIN, S$_2\}$, and S$_2$ satisfies the criteria for being the root.

German sentence structure is often formulated within the topological field model, which is not explicitly annotated in TIGER, but could be inferred as it has been done in (Frank, 2001) for example. Modelling topological fields would however mean that we would have to encode obligatory adjunction and extract several different verbal trees for each of the German sentence types (verb in first, sec-

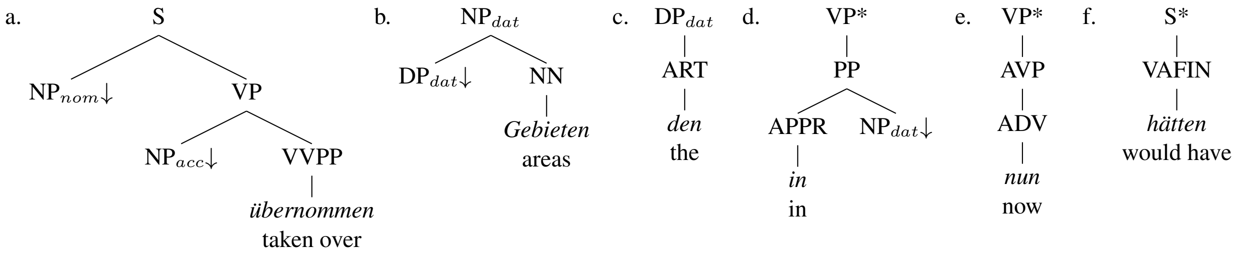


Figure 6: Some canonical initial (a - c) and modifier (d - f) trees extracted from the sentence in Figure 3.

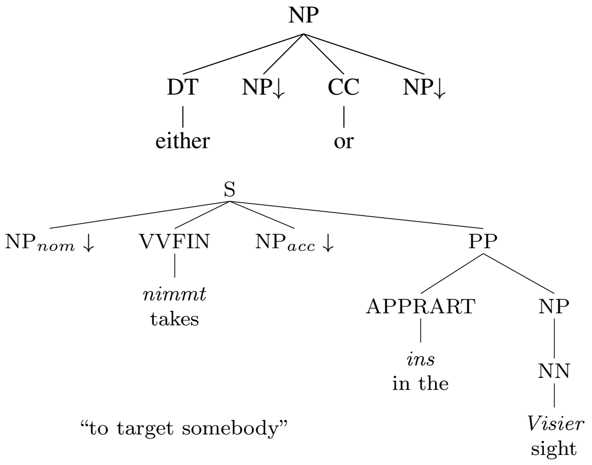


Figure 7: Multi-anchored trees

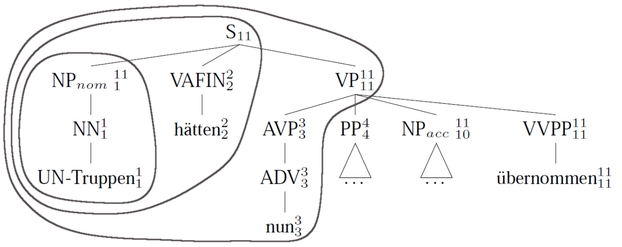


Figure 8: Connection paths for the first three words of the sentence from Figure 3. Nodes are numbered according to the canonical tree to which they belong.

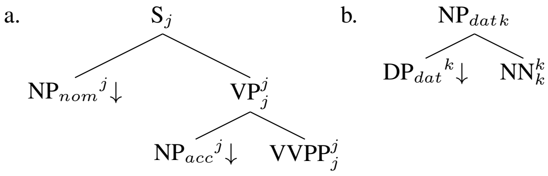

a.
S_j
$NP_{nom}{}^{j}\downarrow$
VP^{j}_{j}
$NP_{acc}{}^{j}\downarrow$
$VVPP^{j}_{j}$
b.
$NP_{dat\,k}$
$DP_{dat}{}^{k}\downarrow$
NN^{k}_{k}

Figure 9: Two prediction trees extracted from the sentence in Figure 3. j and k are the prediction markers.

ond or last position), resulting in a huge lexicon. Using a similar extraction procedure as for English instead, enables us to extract verbal trees that generalise to several sentence types. Consider for example the elementary trees (a) and (f) in Figure 6: all three sentence types can be generated, depending on where (f) sister-adjoins to node S of (a).

4.2. Prediction Trees

Given the segmentation of PLTAG treebank trees into canonical trees, the prediction trees are induced using the notion of connection paths (Lombardo and Sturt, 2002). A connection path for words $w_1 \ldots w_n$ is the minimal amount of structure that is needed to connect the words $w_1 \ldots w_n$ into the same syntactic tree. This amount of structure is indicated with circles in Figure 8. The structure which is required by the connection path of words $w_1 \ldots w_n$ but which is not part of the elementary trees that are anchored in words $w_1 \ldots w_n$ constitutes the prediction tree. In Figure 8, this occurs for example at the word *hätten*: the S node and the NP node of the elementary tree with index 11 have to be predicted in order to connect all seen words.

Following the definition for prediction trees in Section 2.2., the extracted prediction tree has the shape given in Figure 9(a) derived from the canonical tree in Figure 6(a). The prediction tree (b) is needed to integrate the determiner *den* before having seen its noun. Finally, a third prediction tree which has the same structure as (b) but in accusative case is extracted for a fully incremental derivation of the example sentence.

If nodes from two or more different elementary trees are needed by the connection path, a pre-combined prediction tree is generated. It has unique indices for nodes that originate from different elementary trees. The advantage is that during parsing, derivations can be restricted to the integration of only one prediction tree at a time.

Even though the same definition for prediction trees is used for both grammars, the prediction granularity differs due to the German sentence structure. While in English the subject usually is the only argument which occurs left of the verb that provides the lexical anchor, typically producing sentential predictions only down to the VP level (see Fig. 1(d) for an example), in German the verbal anchor can be right of all arguments, leading to prediction trees as in Figure 9(a). Although this is formally not a problem, there is no psycholinguistic motivation for such a difference in prediction granularity, and we expect the size of the prediction lexicon for German to be larger than for English. This finding thus highlights the need for research to find the optimal prediction grain size.

5. Statistics

The English PLTAG is extracted from Sections 02-21 of the PTB, Section 23 is used to calculate the coverage. The German TIGER Treebank is divided into three sets following the methodology suggested in (Dubey, 2005), resulting

	Templates (Canon. lexicon)					Lexicalized	Pre-combined	Coverage
	Initial	Auxiliary	Modifier	Sum	thereof unique	trees	pred. trees	(in %)
ENGLISH	2,858	3,842	-	**6,700**	3,412	111,705	2,595	99.7
GERMAN	1,860	880	2,619	**5,359**	2,577	144,886	4,407	99.5
GERMAN_CASE	2,742	1,251	3,424	**7,417**	3,645	164,598	6,004	99.3

Table 1: Number of tree types extracted with respect to different grammars. Those grammars do not encode multi-anchored trees. The coverage refers to template tokens extracted from converted unseen data.

in 45,428 sentences for training and 2,523 sentences for development and testing each.

The treebank conversion and extraction procedures generate complete correct tree structures for 97.6% of all trees in the PTB and for 99.8% of all trees in the TIGER Treebank. Loss of conversion coverage in the PTB conversion are due to fragment (FRAG) nodes, inconsistencies and errors in the annotation, and the fact that the structures are still too flat for a modifier of a node to adjoin between two arguments of the same node. For this latter case, complete trees are generated, but the modifier leaves will be in the wrong order. Such cases account for 20% of the conversion errors. Due to sister-adjunction, this problem, which would be more frequent in German, does not occur with TIGER. The failed sentences in German all contain instances of non-recursive coordination.

Details about the sizes of the induced grammars are presented in Table 1. The number of extracted templates (i.e. unlexicalized elementary trees) is of the same order of magnitude as the LTAG lexica extracted by (Xia et al., 2000) and (Chen and Shanker, 2004) from the PTB. For German, we report details about a version of a lexicon from which the case annotation has been disregarded, for better comparison with the English lexicon. However, even GERMAN_CASE, which includes the case marking, is still of manageable size, and the following numbers will be based on this lexicon.

As expected because of the different prediction granularities, considerably more prediction trees are induced for German than for the English. However, neither in the PTB nor in the TIGER Treebank more than 5 trees have to be pre-combined in order to achieve full connectivity. Among the instances where a prediction tree is needed in the PTB, 88.3% of the cases use one prediction tree at a time (92.7% for German), in 10% of the cases two prediction trees have to be combined (7% for German) and in less than 1% of cases predicted nodes from more than three lexical anchors are required.

Even though the coverage percentages on converted unseen test data sound satisfactory ($>$ 99% for both English and German), they indicate that the grammars do not converge, which is confirmed by the graph in Figure 10. Thus even after having seen all training data, new templates still occur in unseen data. However, given the template frequency distribution, we consider this fact not to be problematic for parsing: for German, the 3,645 templates that have been seen only once during training account for less than 0.45% of all template tokens. In contrast, there are only 114 templates with a frequency of 1000 or higher, but they cover 83.5% of all template occurrences. The situation is similar for the English lexicon.

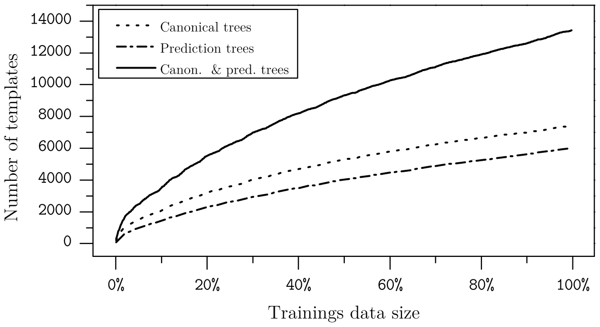


Figure 10: Growth of the lexicon GERMAN_CASE during training

In the English lexicon the average ambiguity is 2.37 trees per word. With 1.96 trees per word, it is lower for GERMAN_CASE, which is plausible because of the richer morphology. If lemmata instead of surface forms are considered in German, the number approaches the English one. The distribution is Zipfian as can be expected for language data. There are a few words which anchor lots of different trees. The most ambiguous ones in English are *and* (578 trees), *or* (219 trees), *as*, *in*, *but* and *is*, and *und* ('and', 533 trees), the comma (249 trees), *ist* ('is') and *oder* ('or') in German.

6. Related Work

We are not the first ones to convert the PTB into TAG format and extract a TAG lexicon from it. However, our treebank and lexicon differs from earlier approaches (Xia et al., 2000; Chen and Shanker, 2004) in that it adds the linguistically motivated NP disambiguation from (Vadas and Curran, 2007) (as opposed to heuristically annotated NPs), and in that it extracts the PLTAG prediction lexicon and encodes also multi-anchored trees.

Neumann (2003) also uses a recursive, head-driven extraction procedure to obtain stochastic lexicalized tree grammars from (untransformed) German and English treebanks. However, modification is not factored out of the trees in terms of adjunction, so the lexicon is much larger than ours and does not generalise well to unseen data. Neumann (2003) induces 12k tree templates from Sections 02–04 of the PTB as opposed to the roughly 6k tree templates extracted by (Xia et al., 2000), (Chen and Shanker, 2004) or our method from Sections 02–21, and 10k templates from a small portion ($<$ 4500 sentences) of the NeGra corpus

(Skut et al., 1997).
In contrast, Frank (2001) first heavily restructures the German NeGra Treebank to be able to extract a linguistically sound LTAG lexicon. Unfortunately, neither the conversion and extraction rules nor the grammar are available as resources.

7. Conclusion

We presented the first resources that are available for a recent psycholinguistically motivated, incremental version of TAG: German and English PLTAG treebanks of derived trees and linguistically motivated lexica, converted and extracted from large, annotated treebanks. Those resources represent a valuable contribution to various fields, especially since for German no LTAG treebank and lexicon have been available to date.

8. References

Tilman Becker, Aravind K. Joshi, and Owen Rambow. 1991. Long-Distance Scrambling and Tree Adjoining Grammars. In *Proceedings of the fifth Conference of the EACL*.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Aoife Cahill. 2004. *Parsing with Automatically Acquired, Wide-Coverage, Robust, Probabilistic LFG Approximations*. Ph.D. thesis, Dublin City University.

Eugene Charniak. 1996. Tree-bank Grammars. In *Proceedings of the AAAI-96*.

John Chen and Vijay K. Shanker. 2004. Automated extraction of TAGs from the Penn Treebank. *New developments in parsing technology*.

David Chiang. 2000. Statistical Parsing with an Automatically-Extracted Tree Adjoining Grammar. In *Proceedings ACL*, volume 38.

Vera Demberg and Frank Keller. 2008. A Psycholinguistically Motivated Version of TAG. In *Proceedings of TAG+9*.

Vera Demberg-Winterfors. 2010. *A Broad-Coverage Model of Prediction in Human Sentence Processing*. Ph.D. thesis, University of Edinburgh.

Amit Dubey. 2005. *Statistical Parsing for German: Modeling syntactic properties and annotation differences*. Ph.D. thesis, Universität des Saarlandes.

Anette Frank. 2001. Treebank Conversion. Converting the NEGRA treebank to an LTAG grammar. In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis*.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of computer and system sciences*, 10(1):136–163.

Miriam Kaeshammer. 2012. A German Treebank and Lexicon for Tree-Adjoining Grammars. Master's thesis, Saarland University.

Yuki Kamide, Christoph Scheepers, and Gerry T. M. Altmann. 2003. Integration of syntactic and semantic information in predictive processing: Cross-linguistic evidence from German and English. *Journal of Psycholinguistic Research*, 32.

Lars Konieczny. 2000. Locality and parsing complexity. *Journal of Psycholinguistic Research*, 29(6).

Vincenzo Lombardo and Patrick Sturt. 2002. Incrementality and Lexicalism. In *Lexical Representations in Sentence Processing*.

David M. Magerman. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Stanford University.

Wolfgang Maier. 2010. Direct Parsing of Discontinuous Constituents in German. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*.

Mitchell P. Marcus, Mary A. Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2).

Günter Neumann. 2003. A Uniform Method for Automatically Extracting Stochastic Lexicalized Tree Grammars from Treebanks and HPSG. *Treebanks: Building and Using Parsed Corpora*.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*.

Adrian Staub and Charles Clifton. 2006. Syntactic prediction in language comprehension: Evidence from either ...or. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32.

Patrick Sturt and Vincenzo Lombardo. 2005. Processing coordinate structures: Incrementality and connectedness. *Cognitive Science*, 29.

Patrizia Tabossi, Rachele Fanari, and Kinou Wolf. 2005. Spoken idiom recognition: Meaning retrieval and word expectancy. *Journal of Psycholinguistic Research*, 34.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and Julie C. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268.

David Vadas and James R. Curran. 2007. Adding Noun Phrase Structure to the Penn Treebank. In *Proceedings ACL*, volume 45.

Jos J. A. van Berkum, Colin M. Brown, and Peter Hagoort. 1999. Early referential context effects in sentence processing: Evidence from event-related brain potentials. *Journal of Memory and Language*, 41.

Fei Xia, Martha Palmer, and Aravind Joshi. 2000. A Uniform Method of Grammar Extraction and its Applications. In *Proceedings of the 2000 Joint SIGDAT conference on EMNLP/VLC*.

Fei Xia. 2001. *Automatic Grammar Generation From Two Different Perspectives*. Ph.D. thesis, University of Pennsylvania.

XTAG Research Group. 2001. A lexicalized tree adjoining grammar for english. Technical Report IRCS-01-03, IRCS, University of Pennsylvania.