A Compact but Linguistically Detailed Database for German Verb Subcategorisation relying on Dependency Parses from Web Corpora: Tool, Guidelines and Resource

Silke Scheible, Sabine Schulte im Walde, Marion Weller, Max Kisselew
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
{scheible,schulte,wellermn,kisselmx}@ims.uni-stuttgart.de

Abstract

This paper describes the *SubCat-Extractor* as a novel tool to obtain verb subcategorisation data from parsed German web corpora. The SubCat-Extractor is based on a set of detailed rules that go beyond what is directly accessible in the parses. The extracted subcategorisation database is represented in a compact but linguistically detailed and flexible format, comprising various aspects of verb information, complement information and sentence information, within a one-line-per-clause style. We describe the tool, the extraction rules and the obtained resource database, as well as actual and potential uses in computational linguistics.

1 Introduction

Within the area of (automatic) lexical acquisition, the definition of lexical verb information has been a major focus, because verbs play a central role for the structure and the meaning of sentences and discourse. On the one hand, this has led to a range of manually or semi-automatically developed lexical resources focusing on verb information, such as the Levin classes (Levin, 1993), VerbNet (Kipper Schuler, 2006), FrameNet[1] (Fillmore et al., 2003), and PropBank (Palmer et al., 2005). On the other hand, we find automatic approaches to the induction of verb subcategorisation information at the syntax-semantics interface for a large number of languages, including Briscoe and Carroll (1997) for English; Sarkar and Zeman (2000) for Czech; Schulte im Walde (2002) for German; and Messiant (2008) for French. This basic kind of verb knowledge has been shown to be useful in many NLP tasks such as information extraction (Surdeanu et al., 2003; Venturi et al., 2009), parsing (Carroll et al., 1998; Carroll and Fang, 2004) and word sense disambiguation (Kohomban and Lee, 2005; McCarthy et al., 2007).

Subcategorisation information is not directly accessible in most standard annotated corpora, and thus typically requires a complex approach to induce verb knowledge at the syntax-semantic interface, cf. Schulte im Walde (2009) for an overview of methodologies. Even more, with the advent of web corpora, empirical linguistic researchers aim to rely on large corpus resources but have to face data where not only deep tools but also standard tools such as tokenisers and taggers often fail.

We describe a novel tool to extract verb subcategorisation data from parsed German web corpora. While relying on a dependency parser, our extraction was based on a set of detailed guidelines to maximise the linguistic value of the subcategorisation information but nevertheless represent the data in a compact, flexible format. In the following, we outline our subcategorisation extractor and describe the format of the subcategorisation database, as well as actual and potential uses in computational linguistics.

2 Subcategorisation Extraction: Tool, Rules and Resource Database

This section provides an overview of the *SubCat-Extractor*, a new tool for extracting verb subcategorisation information. The goal of the SubCat-Extractor is to extract verbs with their complements from parsed German data following a special set of extraction rules devised for this purpose.

[1] Even though the FrameNet approach does not only include knowledge about verbal predicates, they play a major role in the actual lexicons.

Position	Word	Lemma	POS	Morphology	Head	Dependency Relation
1	Er	er	PPER	nom, sg, masc, 3	2	SB (subject)
2	fliegt	fliegen	VVFIN	sg, 3, pres, ind	0	–
3	am	an	APPRART	dat, sg, neut	2	MO (modifier)
4	Wochenende	Wochenende	NN	dat, sg, neut	3	NK (noun kernel element)
5	nach	nach	APPR		2	MO (modifier)
6	Berlin	Berlin	NE	dat, sg, neut	5	NK (noun kernel element)
7	.	–	$.		6	–

Table 1: Example input.

In this section, we describe the input format for the SubCat-Extractor (Section 2.1), the specificities of the extraction rules (Section 2.2), the output format (Section 2.3) and the induced subcategorisation database (Section 2.4) in some detail.

2.1 Input Format

The input format required by the SubCat-Extractor is parsed text produced by Bernd Bohnet's MATE dependency parser (Bohnet, 2010). The parses are defined according to the tab-separated CoNNL[2] format, so in principle any parser output in CoNNL format can be processed by the SubCat-Extractor. Since the extraction rules rely on part-of-speech and syntactic function information in the parses, the respective format specifications have to be taken into account, too: The SubCat-Extractor tool is specified for part-of-speech tags from the *STTS* tagset (Schiller et al., 1999) and syntactic functions from *TIGER* (Brants et al., 2004; Seeker and Kuhn, 2012).

Table 1 shows an example sentence from the Bohnet parser that can serve as input to the SubCat-Extractor: *Er fliegt am Wochenende nach Berlin.* 'He flies to Berlin at the weekend'. For simplicity, we omit columns that consistently do not carry information: in the actual parser output, some columns used for evaluation purposes do not provide information for our parsing purposes. Accordingly, the information in Table 1 is restricted to the following information: the first column shows the sentence position, the second column shows the actual word type, the third column shows the lemma, the forth column shows the part-of-speech, the fifth column shows morphological information, the sixth column shows the head of the dependency relation, and the seventh column specifies the dependency relation, i.e., the syntactic function. For applying the SubCat-Extractor, each sentence must be followed by an empty line.

2.2 Extraction Rules

The SubCat-Extractor considers any verbs that are POS-tagged as finite (V*FIN), infinite (V*INF), or participial (V*PP) in the input files. We have devised detailed rules to extract the subcategorisation information, going beyond what is directly accessible in the parses. In particular, our rules include the following cases:

- Identification of relevant dependants of finite full verbs, across tenses.
- Identification of the auxiliaries *sein* 'to be' and *haben* 'to have' and modal verbs as full verbs, excluding all other instances from consideration.
- Identification of relevant dependants of infinite verb forms occurring with finite auxiliaries/modals.
- Distinguishing between active/passive voice.
- Resolving particle verbs.

An example of only indirectly accessible information in the parses is the definition of subjects, which –in the parses– are always attached to the finite verb; so in a sentence like *Die Mutter würde Suppe machen.* 'The mother might make soup.' we have to induce that *Mutter* 'mother' is the subject of *machen* 'make' because it is not a dependant of the full verb.

Appendix A provides more details of our rules, which represent the core of the SubCat-Extractor, showing under which conditions the rules apply, and what information is extracted. The list might serve as guidelines for anyone interested in applying or extending the SubCat-Extractor. Examples of the rules can be found in Appendix B. The complete guidelines are available from `www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/subcat-extractor.en.html`.

[2] `www.clips.ua.ac.be/conll/`

2.3 Output Format

The output of the SubCat-Extractor represents a compact but linguistically detailed database for German verb subcategorisation: It contains the extracted verbs along with the following tab-separated information:

(1) verb information;
(2) subcategorisation information;
(3) applied rule;
(4) whole sentence.

In the following, this information is described in more detail.

(1) Verb Information: Information on the extracted target verb consists of the following four parts (separated by colons):

1. *Dependency relation of the target verb*, according to the TIGER annotation scheme. For verbs located at the root position of the parse the relation is '–'. If the verb is included in a passive construction, the relation is prefixed with the label 'PAS_'.
2. *Part-of-speech (POS) tag* of the target verb, according to STTS.
3. *Position* of the target verb in the sentence, with the count starting at zero.
4. *Lemma* of the target verb.

Examples of this verb information are

- `--:VVFIN:2:planen`
- `OC:VVPP:4:entscheiden`
- `PAS_OC:VVINF:9:beantworten`

Of special interest concerning German particle verbs is the following specification: In cases where the SubCat-Extractor locates a verb particle in the sentence (with dependency relation 'SVP') that directly depends on the target verb, the particle is added as prefix to the lemma. An example of this procedure is *Petacchi schied verletzt aus.* → `--:VVFIN:1:ausscheiden`.

(2) Subcategorisation Information: The subcategorisation contains all complements $\texttt{Comp}_i$ of a given target verb as determined by the extraction rules in Appendix A, disregarding the distinction between arguments and adjuncts. The information is listed within angle brackets, and individual complements are separated by pipe symbols:

$\texttt{<Comp}_1\texttt{|Comp}_2\texttt{|...|Comp}_n\texttt{>}$

The complements included in the subcategorisation information are distinguished as follows:

(a) ***All complements (but PPs):*** SB (subject), EP (expletive), SBP (passivised subject), MO (modifier; restricted to adverbs), OA (accusative object), OA2 (ditto, in case there are two OAs in the same clause), OC (clausal object), OG (genitive object), PG (phrasal genitive), DA (dative object), PD (predicate), NG (negation), and AG (genitive attribute) use the same format as the verb information described above, i.e.

1. Dependency relation of the complement.
2. POS tag of the complement.
3. Position of the complement in the sentence.
4. Lemma of the complement.

An example complement (a subject represented by a personal pronoun (PPER), *ich* 'I') would be `SB:PPER:8:ich`.

An important feature of the subcategorisation extraction is that any subject (SB) tagged as relative pronoun (PRELS) is resolved to its ancestor, for example: *Kinder, die müde sind, ...* ('children who are tired, ...') → `<SB:NN:0:Kind|PD:ADJD:3:müde>`.

(b) ***PPs:*** MO (modifier; excluding adverbs), MNR (postnominal modifier), and OP (prepositional object with POS tag APPR (preposition) or APPRART (preposition incorporating article)), as well as CVC (collocational verb construction) introduce prepositional phrases (PPs). For this reason, the individual entries are further extended by adding the arguments of the prepositions. Double colons are used to separate preposition information from PP argument information:

1. Dependency relation of the preposition.
2. POS tag of the preposition.
3. Position of the preposition in the sentence.
4. Lemma of the preposition.

double colon ::

5. POS tag of the PP argument.
6. Case of the PP argument.
7. Position of the PP argument.
8. Lemma of the PP argument.

An example PP complement (*im Sommer* 'in the summer') would be `MO:APPRART:6:in::NN:dat:7:Sommer`.

Verb Information	Subcategorisation Information & Sentence(s)
–:VVFIN:1:fliegen	<SB:PPER:0:er|MO:APPRART:2:an::NN:dat:3:Wochenende|MO:APPR:4:nach::NE:dat:5:Berlin> _[Er]SB [[fliegt]]– [am]MO Wochenende [nach]MO Berlin ._
–:VVFIN:1:ausscheiden	<SB:NE:0:Petacchi|MO:VVPP:2:verletzen> _[Petacchi]SB [[schied]]– [verletzt]MO:OTHER [[aus]]SVP ._
–:VVFIN:2:stattfinden	<SB:NN:1:Kulturfestival|MO:APPRART:3:in::NN:dat:4:Sommer> _Zahlreiche [Kulturfestivals]SB [[finden]]– [im]MO Sommer [[statt]]SVP ._
OC:VVINF:6:verstehen	<SB:PIS:1:man|OA:NN:3:Begriff|CP:KOUS:0:wenn> _Wenn man den [Begriff]OA der Netzwerkeffekte [[verstehen]]OC *will* , . . ._
OC:VVPP:6:fahren	<SB:NE:1:Zabel|MO:ADV:3:gerne|MO:APPR:4:in::NE:dat:5:Gelb> _Erik Zabel ∗wäre∗ [gerne]MO:ADV [in]MO Gelb [[gefahren]]OC [. . .]_ _Erik [Zabel]SB [[wäre]]– gerne in Gelb gefahren [. . .]_
PAS_OC:VVPP:5:kaufen	<SB:NN:0:Tier|MO:APPR:2:aus::NN:dat:4:Grund> _Tiere *werden* [aus]MO verschiedensten Gründen [[gekauft]]OC ._

Table 2: Example output.

If a PP involves coordination, both parts are resolved and included. For example: *im Sommer und Winter* induces `MO:APPRART:6:in::NN:dat:7:Sommer|` `MO:APPRART:6:in::NN:dat:9:Winter`.

(c) ***Conjunctions:*** The conjunction POS tags KON, CJ, CD, and – are excluded from consideration. The PPs are an exception to this (see above).

(3) Applied Rule: The rule that was applied to extract the verb and subcategorisation information is denoted, cf. Appendix A.

(4) Sentence: Finally, the whole sentence in which the target verb occurs is listed with the following mark-up:

- Double brackets *[[. . .]]* denote the verb.
- Single brackets followed by a label *[. . .]LABEL* denote complements of the target verb and their dependency relations.
- Curly brackets {. . . } denote the parent of the target verb.
- Asterisks ∗. . . ∗ are used to mark up a finite (auxiliary) verb on which the target verb depends and whose complements are added to the target verb's subcategorisation.
- Whenever the dependants of a finite (auxiliary) verb are included in the frame, a second sentence is added to the output showing the dependants of the respective finite verb (see example *fahren* in Table 2).

Examples Table 2 provides examples of the subcategorisation output, including those mentioned in the preceding parts of this section.

2.4 Subcategorisation Resource

So far, we have applied the SubCat-Extractor to dependency parses of the German web corpus *sdeWaC* (Faaß and Eckart, 2013),[3] a cleaned version of the German web corpus *deWaC* created by the *WaCky* group (Baroni et al., 2009). The corpus cleaning had focused mainly on removing duplicates from the *deWaC*, and on disregarding sentences that were syntactically ill-formed (relying on a parsability index provided by a standard dependency parser (Schiehlen, 2003)). The sdeWaC contains approx. 880 million words and is provided by `wacky.sslmit.unibo.it/`.

The *sdeWaC* subcategorisation database comprises 73,745,759 lines (representing the number of extracted target verb clauses). 63,463,223 (86%) of the target verb tokens appeared in active voice, and 10,282,536 (14%) of them appeared in passive voice. Table 3 shows the distribution of the verb clauses over full, auxiliary and modal verbs.

POS	Number of Clauses
VAFIN	11,395,914
VAINF	901,106
VAPP	302,586
VMFIN	348,056
VMINF	4,373
VMPP	5,959
VVFIN	33,640,028
VVINF	11,410,381
VVIZU	1,129,094
VVPP	14,608,262

Table 3: Full, auxiliary and modal verb clauses.

[3] `www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/sdewac.en.html`

3 Applications

The subcategorisation extraction tool and –more specifically– the subcategorisation resource described in the previous section are of great potential use because the information is represented in a compact format, but nevertheless with sufficient details for many research questions. Furthermore, the linear one-line-per-clause format allows quick and easy access to the data; in many cases, basic unix tools or simple perl or python scripts can be used, rather than going into the complexity of parse structures for each research question. The following paragraphs introduce applications of the database within our research project.

Subcategorisation Frame Lexicon As a natural and immediately subsequent step, we induced a subcategorisation frame lexicon from the verb data. Taking voice into account, we summed over the various complement combinations a verb lemma appeared with. For example, among the most frequent subcategorisation frames for the verb *glauben* 'believe' are a subcategorised clause 'believe that' (freq: 52,710), a subcategorised prepositional phrase with preposition an_{acc} 'believe in' (freq: 4,596) and an indirect object 'trust s.o.' (freq: 2,514). In addition, we took the actual complement heads into account. For example, among the most frequent combinations of heads that are subjects and indirect objects of *glauben* are <*man, Umfrage*> 'one, survey' and <*keiner, ihm*> 'nobody, him'. Paying attention to a specific complement type (e.g., the direct object within a transitive frame), we induced information that is relevant for collocation analyses. For example, among the most frequent indirect objects of *glauben* in a transitive frame are *Wort* 'word', *Bericht* 'report', and *Aussage* 'statement'. The subcategorisation frame lexicon has not been evaluated by itself but by application to various research studies (see below).

Subcategorisation Information for Statistical Machine Translation (SMT) Weller et al. (2013) is an example of research that applied our subcategorisation data. They improved the prediction on the case of noun phrases within an SMT system by integrating quantitative information about verb subcategorisation frames and verb–complement syntactic strength.

Prediction of Passives-of-Reflexives Zarries et al. (2013) exploited the linguistic and formatting advantages of our data, when they predicted the potential of building 'passives of reflexives' for German transitive verbs, such as

Erst wird sich$_{REFL}$ *geküsst, ...*
'First is REFL kissed, ...'.

They used the one-line-per-clause format to identify relevant subcategorisation frames of verbs and to restrict the types of noun complement heads that were allowed for specific syntactic functions.

Classification of Prototypical vs. Metaphorical Uses of Perception Verbs David (2013) used the subcategorisation information and the sentence information for (i) a manual inspection, (ii) corpus-based annotation and (iii) an automatic classification of prototypical vs. metaphorical uses of a selection of German perception verbs. The sentence information (cf. Section 2.3) in connection with the compact verb and subcategorisation supported the annotation purposes of perception verb senses; the verb information and the subcategorisation information were exploited as classification features. Relying on our subcategorisation database, a Decision Tree classification resulted in 55-60% accuracy scores in the 3-way and 4-way classifications.

Potential Uses In order to illustrate the potential of the information provided by our subcategorisation database, we add ideas of potential uses.

- *Complement order variations with regard to the verb type, the clause type and the subcategorisation frame:*
  The one-line-per-clause format provides verb information regarding the verb dependency and the position of the verb in the clause, as well as types and positions of the various complements, so it should be straightforward to quantify over the complement order variations ('scrambling') in relation to the verb information.
- *Extraction of light-verb constructions ('Funktionsverbgefüge') with prepositional objects:*
  The eight-tuple information in combination with the verb information should enable an easy access to light-verb constructions, as all relevant information is within one line of the subcategorisation database.

- *Quantification of verb modalities:*
  Since the information of whether a full verb depends on a modal verb (or not) is kept in the sentence information, the subcategorisation database should be useful to explore and quantify the modal conditions of verb types (in combination with specific types of complement heads).

4 Discussion

Section 2 introduced the SubCat-Extractor as a new tool for extracting verb subcategorisation information. The goal of the SubCat-Extractor is to extract German verbs along with their complements from parsed German data in tab-separated CoNNL format. We have devised detailed rules to extract the subcategorisation information from the dependency relations, going beyond what is directly accessible. So far, we have applied the SubCat-Extractor to dependency parses of a German web corpus, sdeWaC, comprising approx. 880 million words. Section 3 provided some actual and potential uses of the subcategorisation data.

The SubCat-Extractor is, of course, not restricted to be used for parses of only corpora from the web. It can be applied to any kind of corpus data, given that the corpus data is parsed by a parser with CoNNL format output, using the STTS tagset and the TIGER node set. We however defined the rules of the SubCat-Extractor in such a way that they are robust towards a large amount of noise in the underlying data. Since the MATE parser would always generate a parse for a sentence, and integrate erroneous as well as correct words and phrases, the rules of the SubCat-Extractor need to ensure a reliability filter for erroneous dependencies. For example, the sdeWaC web corpus parses commonly identify more than one subject for a full verb, because complement inflections (and thus case prediction) might be erroneous. Our rule set aims to extract at most one subject per full verb. In sum, we presented

- a ***new tool*** *(SubCat-Extractor)* that can be applied to German dependency parses and should be robust to extract verb subcategorisation information from web corpora,
- a ***new verb subcategorisation database*** obtained from the sdeWaC, with compact but nevertheless linguistically detailed information, and
- a ***new subcategorisation frame lexicon*** induced from the subcategorisation database.

The tool, the subcategorisation database and the subcategorisation frame lexicon are freely available for education, research and other non-commercial purposes:

- *tool:*
  www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/subcat-extractor.en.html
- *database/lexicon:*
  www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/subcat-database.en.html

Acknowledgements

The research presented in this work was funded by the DFG Research Project *Distributional Approaches to Semantic Relatedness* (Max Kisselew, Silke Scheible, Marion Weller), and the DFG Heisenberg Fellowship SCHU-2580/1-1 (Sabine Schulte im Walde).

Appendix A. Extraction Rules.

The *SubCat-Extractor* rules specify (i) the types of verbs that are considered for extraction, and (ii) the dependants of these verbs that are included in the subcategorisation information.

1) Extraction of finite verbs

Extraction rules for the finite verb types VVFIN **(a)**, VMFIN **(b)**, and VAFIN **(c)**:

Conditions:

- **(a)**: No special conditions.
- **(b)**: VMFIN does not depend on a V* (i.e. VMFIN is a full verb).
- **(c)**: VAFIN does not depend on a V* (i.e. VAFIN is a full verb).
  Special case (c'): a PD (predicate) depends on VAFIN.

Extract:

- **(a)**, **(b)**, **(c)**: All dependants of the finite verb.
- **(c')**: Also extract all dependants of PD as complements of VAFIN.

2) Extraction of participle verbs

For V*PP we distinguish between four cases:

i) Compound tense: VVPP (a), VMPP (b), and VAPP (c) are extracted if the following applies:

Conditions:

- **(a)**, **(b)**, **(c)**: The sentence contains a finite verb (VAFIN or VMFIN).
- **(a)**, **(b)**, **(c)**: The participle verb is not a PD.
- **(a)**, **(b)**, **(c)**: The participle verb directly depends on a VA* whose head is *sein* 'to be' or *haben* 'to have'.
- **(b)**, **(c)**: There is no V* in the sentence that depends on the VMPP/VAPP.

Extract:

- **(a)**, **(b)**, **(c)**: All dependants of the participle verb and all complements of the finite verb.

ii) Passive: VVPP (a), VMPP (b), and VAPP (c) are extracted if the following conditions apply, and the participle verb is marked as passive.

Conditions:

- **(a)**, **(b)**, **(c)**: The sentence contains a finite verb (VAFIN or VMFIN).
- **(a)**, **(b)**, **(c)**: The participle verb is not a PD.
- **(a)**, **(b)**, **(c)**: The participle verb directly depends on a VA* whose head is *werden*.

Extract:

- **(a)**, **(b)**, **(c)**: All dependants of the participle verb and all complements of the finite verb.

iii) Past participle dependent on full verb: VVPP is extracted if the following conditions apply, and the participle verb is marked as passive.

Conditions:

- The sentence contains a finite verb.
- The participle verb is not a PD.
- The participle verb directly or indirectly depends on the finite verb.
- The participle verb directly depends on a full verb VV*.

Extract:

- All dependants of the participle verb and all complements of the finite verb.

iv) Predicative pronoun: Predicative pronouns are extracted if the following conditions apply, and the participle verb is marked as passive.

Conditions:

- The sentence contains a finite verb.
- The participle verb is a PD.
- The participle verb directly or indirectly depends on the finite verb.

Extract:

- All dependants of the participle verb and all complements of the finite verb.

3) Extraction of infinitival verbs

For V*INF we distinguish two cases:

i) V*INF without *zu*, in combination with a modal verb or in a compound tense (future): VVINF **(a)**, VMINF **(b)**, and VAINF **(c)** are extracted as follows:

Conditions:

- **(a)**, **(b)**, **(c)**: Sentence contains a finite verb.
- **(a)**, **(b)**, **(c)**: V*INF has no particle *zu*.
- **(a)**, **(b)**, **(c)**: V*INF directly depends on VM* or VA* with head *werden*.
- **(b)**, **(c)**: The sentence does not contain a V* that depends on VMINF or VAINF.

Extract:

- **(a)**, **(b)**, **(c)**: All dependants of V*INF.
- **(a)**, **(b)**, **(c)**: All complements of V*INF.

ii) V*INF with *zu*: VVINF **(a)**, VMINF **(b)**, and VAINF **(c)** are extracted as follows:

Conditions:

- **(a) (b)**, **(c)**: Sentence contains a finite verb.
- **(a) (b)**, **(c)**: V*INF has a particle *zu*.
- **(a) (b)**, **(c)**: V*INF directly depends on a VV* or a VA*.
  Special case (ii'): VA* has head *sein*.
- **(b)**, **(c)**: The sentence does not contain a verb V* that depends on VMINF/VAINF.

Extract:

- **(a)**: All dependants of V*INF.
- **Special case (ii'):** Complements of the finite verb.

In the case of ii', V*INF is marked as passive.

Appendix B. Rule Examples.

Rule	Category	Examples	Glosses
Finite verbs			
1 (a)	VVFIN	Er *fliegt* am Wochenende nach New York. Das Kind *singt* schon seit Stunden. Sie *kauften* sich drei Blumen.	He *flies* to New York at the weekend. The child has been *singing* for hours. They *bought* (themselves) three flowers.
1 (b)	VMFIN	Er *will* das Auto. Er *darf* das bestimmt nicht.	He *wants* the car. He *may* certainly not.
1 (c)	VAFIN	Das Kind *hat* viele Autos. Peter *ist* im Kindergarten.	The child *has* many cars. Peter *is* in the kindergarden.
1 (c')	VAFIN	Die Eltern *sind* am meisten *betroffen*. Gegen ihn *ist* Anklage *erhoben* wegen . . . Sie *waren* so *geliebt*.	The parents *are affected* the most. He *is charged* with . . . They *were* so *beloved*.
Participle verbs: compound tense			
2 (i)(a)	VVPP	Die Mutter hat die Suppe *gekocht*. Die Mutter muss die Suppe *gekocht* haben. Das Kind ist weit *geschwommen*. Das Kind wird weit *geschwommen* sein.	The mother has *cooked* the soup. The mother must have *cooked* the soup. The child has *swum* far. The child will have *swum* far.
2 (i)(b)	VMPP	Er hat das unbedingt *gewollt*.	He absolutely *wanted* this.
2 (i)(c)	VAPP	Das Kind wird viele Autos *gehabt* haben. Peter wird im Kindergarten *gewesen* sein.	The child will have *had* many cars. Peter will have *been* in the kindergarden.
Participle verbs: passive			
2 (ii)(a)	VVPP	Die Suppe wird *gekocht*. Die Suppe soll *gekocht* werden. Die Suppe hat *gekocht* werden müssen.	The soup is being *cooked*. The soup should be *cooked*. The soup has had to be *cooked*.
Participle verbs: past participle dependent on full verb			
2 (iii)(a)	VVPP	Wir fühlen uns davon *betroffen*. Die Sachen gehen immer *verloren*.	We feel *affected* by that. The things always *get lost*.
Participle verbs: predicative pronoun			
2 (iv)[4]	V*PP	Die Eltern sind am meisten *betroffen*. Die Eltern bleiben am meisten *betroffen*. Sie waren so *geliebt*.	The parents are *affected* the most. The parents remain *affected* the most. They were so *beloved*.
Infinitival verbs without particle *zu*			
3 (i)(a)	VVINF	Er will *gehen*. Er darf sich das Auto morgen *kaufen*.	He wants to *go*. He may *buy* (himself) the car tomorrow.
3 (i)(b)	VMINF	Er wird das morgen *dürfen*. Er will das morgen *dürfen*.	He will *be allowed* (to do) this tomorrow. He wants to *be allowed* (to do) this tomorrow.
3 (i)(c)	VAINF	Er darf das Auto morgen *haben*. Er will morgen rechtzeitig da *sein*.	He may *have* the car tomorrow. He wants to *be* there in time tomorrow.
Infinitival verbs with particle *zu*			
3 (ii)(a)	VVINF	Er entscheidet zu *gehen*. Er hat gestern entschieden zu *gehen*. Er hat ihm befohlen zu *gehen*.	He decides to *leave*. Yesterday, he decided to *leave*. He told him to *leave*.
3 (ii)(b)	VMINF	Er hat sich entschieden mehr Inhalte zu *wollen*.	He decided to *want* more content.
3 (ii)(c)	VAINF	Er hat sich vorgenommen Zeit zu *haben*. Er hat vorgeschlagen dabei zu *sein*.	He intended to *have* time. He suggested to *be* there.
3 (ii')	V*INF	Die Hinweise sind zu *beachten*. Die Frage ist leicht zu *beantworten*. Die Hilfsmittel sind da zu *sein*.	The indications are to be *respected*. The question is easy to *answer*. The tools are to *be* there.

Table 4: Examples of sentences and applied rules.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Ted Briscoe and John Carroll. 1997. Automatic Extraction of Subcategorization from Corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, pages 356–363, Washington, DC.

John Carroll and Alex C. Fang. 2004. The Automatic Acquisition of Verb Subcategorisations and their Impact on the Performance of an HPSG Parser. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 107–114, Sanya City, China.

John Carroll, Guido Minnen, and Ted Briscoe. 1998. Can Subcategorisation Probabilities Help a Statistical Parser? In *Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*, pages 118–126, Montreal, Canada.

Benjamin David. 2013. Deutsche Wahrnehmungsverben: Bedeutungsverschiebungen und deren manuelle und automatische Klassifikation. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – a Corpus of Parsable Sentences from the Web. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*, Darmstadt, Germany. To appear.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania, Computer and Information Science.

Upali S. Kohomban and Wee Sun Lee. 2005. Learning Semantic Classes for Word Sense Disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 34–41, Ann Arbor, MI.

Beth Levin. 1993. *English Verb Classes and Alternations*. The University of Chicago Press.

Diana McCarthy, Sriram Venkatapathy, and Aravind K. Joshi. 2007. Detecting Compositionality of Verb-Object Combinations using Selectional Preferences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 369–379.

Cédric Messiant. 2008. A Subcategorization Acquisition System for French Verbs. In *Proceedings of the Student Research Workshop at the 46th Annual Meeting of the Association for Computational Linguistics*, pages 55–60, Columbus, OH.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated Resource of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Anoop Sarkar and Daniel Zeman. 2000. Automatic Extraction of Subcategorization Frames for Czech. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 691–697, Saarbrücken, Germany.

Michael Schiehlen. 2003. A Cascaded Finite-State Parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 163–166, Budapest, Hungary.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen, 1999. *Guidelines für das Tagging deutscher Textcorpora mit STTS*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, and Seminar für Sprachwissenschaft, Universität Tübingen.

Sabine Schulte im Walde. 2002. A Subcategorisation Lexicon for German Verbs induced from a Lexicalised PCFG. In *Proceedings of the 3rd Conference on Language Resources and Evaluation*, volume IV, pages 1351–1357, Las Palmas de Gran Canaria, Spain.

Sabine Schulte im Walde. 2009. The Induction of Verb Frames and Verb Classes from Corpora. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook.*, volume 2 of *Handbooks of Linguistics and Communication Science*, chapter 44, pages 952–971. Mouton de Gruyter, Berlin.

Wolfgang Seeker and Jonas Kuhn. 2012. Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3132–3139, Istanbul, Turkey.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 8–15, Sapporo, Japan.

Giulia Venturi, Simonetta Montemagni, Simone Marchi, Yutaka Sasaki, Paul Thompson, John McNaught, and Sophia Ananiadou. 2009. Bootstrapping a Verb Lexicon for Biomedical Information Extraction. In Alexander Gelbukh, editor, *Linguistics and Intelligent Text Processing*, pages 137–148. Springer, Heidelberg.

Marion Weller, Alex Fraser, and Sabine Schulte im Walde. 2013. Using Subcategorization Knowledge to improve Case Prediction for Translation to German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria. To appear.

Sina Zarrieß, Florian Schäfer, and Sabine Schulte im Walde. 2013. Passives of Reflexives: A Corpus Study. Talk at the International Conference *Linguistic Evidence 2013 – Berlin Special: Empirical, Theoretical and Computational Perspectives*, Humboldt-Universität zu Berlin, Germany.