Approximating theoretical linguistics classification in real data: the case of German *nach* particle verbs

Boris HASELBACH[1] *Kerstin* ECKART[1] *Wolfgang* SEEKER[1]
Kurt EBERLE[2,1] *Ulrich* HEID[3,1]
(1) IMS, Universität Stuttgart, Germany
(2) Lingenio GmbH, Heidelberg, Germany
(3) IWIST, Universität Hildesheim, Germany
haselbbs@ims.uni-stuttgart.de, eckartkn@ims.uni-stuttgart.de,
seeker@ims.uni-stuttgart.de, eberle@ims.uni-stuttgart.de,
heid@ims.uni-stuttgart.de

ABSTRACT
Testing a theory against real world data can sometimes be helpful in figuring out the shortcomings of your current theory. In this paper, we test a theory about the syntax-semantics interface of German *nach*-particle verbs against data from a web corpus in order to see if we can use our automatic NLP machinery to corroborate the predictions of the theory. We use state-of-the-art parsers to automatically annotate our data with the features predicted by the theory and then apply a standard clustering approach to approximate the *nach*-particle verb classes of the theory. The results of our experiment not only help us highlighting the more problematic parts of the theory but also teach us about the strengths and weaknesses of our automatic analysis tools.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE, L_2 (OPTIONAL, AND ON SAME PAGE)

Corpusbasierte Überprüfung einer semantischen Klassifikation deutscher *nach*-Partikelverben

Um Unzulänglichkeiten einer Theorie auszumachen ist es mitunter vonnöten, Hypothesen gegen echtes Textmaterial abzugleichen. In diesem Beitrag soll diskutiert werden, wie Vorhersagen einer Theorie zum syntaktischen und semantischen Verhalten deutscher *nach*-Partikelverben gegen Netztexte abgeglichen werden können und wie dabei eine automatische Textverarbeitung unterstützend zum Tragen kommt. Es werden Parser des letzten Stands der Forschung verwendet um die Daten mit den von der Theorie vorhergesagten Merkmalen zu annotieren bevor ein standardisiertes Clustering-Verfahren angewandt wird um die theoretischen *nach*-Partikel-Verb-Klassen nachzubilden. Die Resultate des Experiments unterstreichen nicht nur Problemfälle der Theorie sondern zeigen auch die Stärken und Schwächen der automatischen Analyse.

KEYWORDS: particle verbs, syntax-semantics interface, German, web data, parser combination.

KEYWORDS IN L_2: Partikelverben, Syntax-Semantik-Schnittstelle, Deutsch, Netztexte, Parser-Kombination.

Proceedings of COLING 2012: Technical Papers, pages 1113–1128,
COLING 2012, Mumbai, December 2012.

1 Introduction

We test a theoretically well-motivated hypothesis about the syntax-semantics interface of German *nach*-particle verbs (*nach*-PV) on data from the "real world", to gain insight into the strengths and weaknesses of the theory as well as the limitations of our NLP-methodology. The theory we want to test is a word-syntactic approach to *nach*-PVs implemented in a generative framework. The analysis of five different readings of *nach* in combination with a verb makes different predictions on the argument structure of the complex verb, in particular with respect to dative and marginally to accusative arguments: the verb semantics (especially the *nach* particle) determines the argument structure of the verb. In order to test this assumption, we will turn the argumentation around: to what extent can we use automatically gained syntactic information to recreate the *nach*-PV classes. For this, we automatically annotate large amounts of data with their syntactic structure and identify the argument structure of each verb. Based on this information, we then apply a standard clustering technique to recreate the *nach*-PV classes proposed by the theory. As syntactic indicators for the clustering, dative and accusative arguments seem to be ideal candidates, as they are partially in the scope of the theoretical description. But we also incrementally add further features of potential relevance for the *nach*-PV reading classification: the form of arguments, prepositional phrases (form of preposition, governed case, form of embedded object), adverbials (adverbs and predicative adjectives), as well as clausal objects.[1].

Although automatic parsers are high quality and reliable tools nowadays, their performance degrades when applied to unrestricted all-domain data. Since we work on web text, we apply two very different parsers, which we then combine for a more reliable annotation. For a small subset of the *nach*-PV data, we use a manually created gold standard for dative and accusative arguments. Evaluating the parsers against this gold standard enables us to pinpoint the advantages of each parser and develop a combination scheme.

The paper is structured as follows. In Section 2, we introduce the theory about the *nach*-PV classes and their related argument structures. In Sections 3 to 5, we present the study in which we apply the theoretical approach to the real world data. In Section 3, we present the data we use: a German web corpus, a small gold standard, and a manual classification of all *nach*-PV lemmas from the corpus according to the theory from Section 2. The gold standard is for the evaluation and combination of the parsers, which extract syntactic features related to the *nach*-PVs (Section 4). The classification of *nach*-PV lemmas is used for the evaluation of the clustering based on the extracted syntactic features (Section 5). We conclude by discussing the lessons learned from each step of this study.

2 Phenomena and linguistic modeling

The German verb particle *nach* (≈ 'after') shows a range of different meanings. Haselbach (2011) provides a partial classification of *nach* into five readings that behave differently with respect to licensing a dative argument:

1. ⊕DAT: direction reading (DIR)
2. ⊕DAT: copy manner reading (MAN)
3. ⊖DAT: copy creation reading (CRE)
4. ⊖DAT: once-more/restitution reading (OMR)
5. ⊖DAT: continuation reading (CONT)

This classification does not cover yet e.g. an intensifying reading such as in *nachdenken* ('to reflect') or a prove/check reading such as in *nachprüfen* ('to recheck').

[1]For similar work on the German verb particle *an* (≈ 'on') see Springorum et al. (2012)

2.1 Modeling at the syntax-semantics interface

(Haselbach, 2011) provides a syntactico-semantic modeling of the five readings of *nach* in terms of Discourse Representation Theory (e.g. Kamp and Reyle, 1993; Roßdeutscher and Kamp, 2010), combined with word-syntactic principles from Distributed Morphology (Halle and Marantz, 1993). Haselbach (2011) implements them by means of the extended VP-shell hypothesis (Larson, 1990). His syntactico-semantic modeling comes close to Nicol's (2002) implementation who argues that verb particles are spelled out instances of functional heads in the verbal domain. Precisely, Haselbach (2011) argues that (i) *nach* either represents the head of a functional projection *w*P which is above VP and projects a dative argument in its specifier, cf. (1-a); or (ii) it is the realization of the head of a functional projection *x*P, which is also above VP however does not project a dative argument in its specifier, cf. (1-b).

(1) a. [$_{wP}$ DP$_{DAT}$ [$_{w'}$ *w*=“*nach*” VP]]
b. [$_{xP}$ *x*=“*nach*” VP]

The idea is that *nach* adds a second eventuality to the semantics by presupposition. The functional difference between *w* and *x* is that *w*, i.e. if a dative is present in the structure, allows *nach* to access event properties in the underlying VP, whereas *x* allows *nach* to access state properties in the underlying VP. Accessing event properties here means that properties of the event in the VP are assigned to the event presupposed by *nach*. Accessing event properties can either be the direction (class DIR) or the manner (class MAN) of an event. By presupposing a second **event** in the semantics, a slot for a further argument is created in the specifier of the functional projection headed by *nach*, i.e. in Spec,*w*P. This argument, which surfaces as dative, is interpreted as a participant in the presupposed event. Accessing state properties, on the other hand, means that properties of a state, if present, within the VP are assigned a presupposed **state**. These can be result or progression state properties. A result state property can either be the existence of an object, i.e. for creation verbs (class CRE), or predicational states contributed by a deadjectival or denominal verb (class OMR). Progressive state properties are stative run-time properties of an event, i.e. the state that can be described by means of the event taking place (class CONT). In these cases, no additional argument slot is semantically present, and thus no dative is licensed.

2.1.1 Five readings of *nach* and the dative

Distinguishing two groups of *nach* reading (plus vs. minus dative), Haselbach (2011) claims that *nach* expressing the meaning “following NP$_{DAT}$” (**directional**: DIR; cf. (2-a)) and *nach* expressing the meaning “do such as NP$_{DAT}$” (**manner**: MAN; cf. (2-b)) take a dative argument.

(2) a. Der Hund rannte dem Hasen nach.
the dog ran the.DAT hare after
“The dog ran after the hare.”
b. Das Mädchen betete der Mutter (den Psalm) nach.
the girl prayed the.DAT mother (the.ACC psalm) after
“The girl copied the mother's praying (of the psalm).”

As opposed to the two readings of *nach* in the context of which a dative is present, there are three readings of *nach* where no dative is present. The first reading is the creation, or copy object reading which can be paraphrased as “making a copy of Y” (**creation**: CRE; cf. (3-a)). The second one is the **once-more/restitution** reading (OMR), which itself has two sub-meanings: a repetitive (once-more) and a restitution reading; cf. (3-b). Haselbach (2011) groups these two readings together as it is not *nach* that is liable for the semantic distinction,

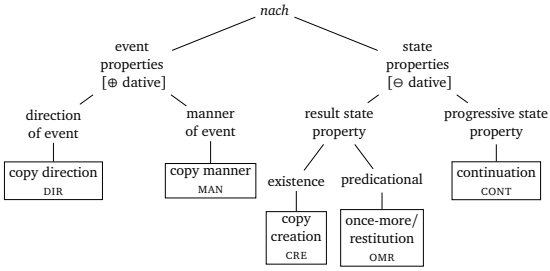

Figure 1: Classification of *nach* readings

but it is considered to be a discourse effect of how the eventualities emerging in the derivation are ordered temporally. The third reading without dative is the **continuation** reading (CONT; cf. (3-c)). In this reading, *nach* conveys a meaning that can be roughly paraphrased as "do something longer than expected". Figure 1 gives an overview of all readings.

(3) a. Der Junge baute den Eiffelturm nach.
the boy built the.ACC Eiffel Tower after
"The boy made a copy of the Eiffel Tower."
b. Der Schmied schärfte das Messer nach.
the blacksmith sharpened the.ACC knife after
"The blacksmith re-sharpened the knife."
c. Die Banane reifte nach.
the banana ripened after
"The banana continued ripening."

2.1.2 Five readings of *nach* and the accusative

Haselbach (2011) stays agnostic about the role of direct objects (i.e. accusative NPs) with respect to the readings of *nach*. With the DIR class, an accusative object may be present or absent, the latter being preferred, cf. (4).

(4) a. Der Hund rannte dem Hasen nach.
the dog ran the.DAT hare after
"The dog ran after the hare."
b. Der Lausbub warf ihr den Ball nach.
the scallywag threw her.DAT the.ACC ball after
"The scallywag threw the ball after her."

For the MAN reading, we encounter both predicates with and without an accusative object, as in (5). Thus, an accusative object does not correlate with the MAN-reading.

(5) Homer tanzte Marge (den Tango) nach.
Homer danced Marge.DAT (the.ACC tango) after
"Homer copied Marge's dancing (of the tango)."

For both the CRE and OMR classes, an accusative object seems to be obligatory. If no dative is present (which is fundamental for these readings), the accusative in (6-a) for CRE, and (7) for OMR cannot be left out. Thus, we consider the direct object to be crucial for these two classes. Replacing the accusative by a dative argument, cf. (6-b), also leads to grammaticality. Then, *nach* is considered to belong to the MAN class.

(6) a. Die Oma strickte *(die Mütze) nach.
the grandmother knitted the.ACC cap after
"The grandmother copied the cap by knitting."
b. Die Oma strickte der Uroma nach.
the grandmother knitted the.DAT great granny after
"The granny copied the great granny's knitting manner."

(7) Der Opa salzte *(die Suppe) nach.
the grandfather salted the.ACC soup after
"The grandfather added more salt to the soup."

The continuation reading of *nach* is not compatible with an accusative argument. This is because this reading seems to correlate with verbs where the verb-internal argument surfaces as subject, i.e. with anti-causative verbs such as in (8). In combination with a verb such as *reifen* ('to ripen') as in (8), neither an accusative, nor a dative, nor both can be present.

(8) Die Banane reifte (*dem Apfel) (*den Pfirsich) nach.
the banana ripened the.DAT apple the.ACC peach after
"The banana continued ripening."

2.1.3 MAN and CRE: one or two classes?

As we already saw in example (6-a) in Section 2.1.2, the *nach*-PV readings MAN and CRE are quite close. Now one could ask whether these two classes are really distinct. Note that the *nach*-PV classes discussed by Haselbach (2011) are interpretation-driven, i.e. in the context of a particular verb, the particle *nach* can evolve a certain interpretation with respect to the nature of the verb. This does not mean that each and every verb does only allow one interpretation. On the contrary, there are many verbs in the context of which *nach* exhibits several of the readings presented above. For example, the complex verb *nachtanzen* (derived from *tanzen*, 'to dance') shows at least three readings: (i) a DIR reading ("follow someone dancing"), (ii) a MAN reading ("copy somebody's dancing manner"), and (iii) a CRE reading ("copy a dance"). DIR and MAN require a dative, i.e. the "somebody" followed or copied, while CRE does not. Nevertheless, the interpretations of MAN and CRE are quite close to each other; verbs expressing the manner of process, such as *stricken* ('to knit', i.e. to produce with wool), *sprechen* ('to speak', i.e. to produce speech), or even *tanzen* ('to dance', i.e. to produce a dance), can also be interpreted as verbs of creation if used with an accusative object (mostly incremental theme, cf. Dowty 1991). Thus, we expect that these two classes are empirically not easily ascertainable.

2.2 Indicators of semantic readings observable in corpus data

In principle, dative and accusative could function as indicators for an automatic approximation of the *nach*-PV readings. However, they are problematic as there is no 1-to-1 relation between indicators and readings, cf. Table 1: datives seem to be better indicators (separating DIR/MAN from the rest) than accusatives, which are unevenly distributed. However, the discriminative power of dative is weakened by the closeness of MAN and CRE, cf. Section 2.1.3. Given this situation, we will check to what extent the use of further features will improve the result of an automatic classification. We will use, incrementally, details of the dative and accusative arguments, PPs, adverbials, as well as clausal complements.

3 Data

As our real world data, we use a fragment of a cleaned version of the German web corpus deWaC (Baroni and Kilgarriff, 2006). We automatically extracted 270k sentences containing at least one *nach*-PV.

indicator	DIR	MAN	CRE	OMR	CONT
dative	+	+	−	−	−
accusative	(−)	?	+	+	−

Table 1: Argument structure indicators for *nach*-PV classes

3.1 Manual gold standard for syntactic criteria for *nach*-PV

From this 270k *nach* sub-corpus, we extracted 270 sentences containing *nach* verbs (i) of each reading (i.e. DIR, MAN, CRE, OMR, and CONT) and (ii) of the frequency ranges high, middle, and low frequency. The set of 270 sentences functions as the gold standard, which we use to evaluate the automatic dependency parsers.[2]

Each *nach*-PV token was annotated by five annotators: two linguistically trained, three untrained. The annotation contains the dative argument and/or the accusative argument of each *nach*-PV token, if present. The linguistically untrained annotators only indicated the presence of a dependent dative or accusative argument. The linguistically trained annotators marked the extension of the argument under consideration, i.e. they annotated the entire textual string. Table 2a shows the inter-annotator agreement. The identification of dative and accusative (Boolean feature) is feasible even for untrained annotators (substantial agreement[3] on all annotators); the linguists even achieved an almost perfect agreement on the extension of the arguments. For the gold standard, the longest extension possible, i.e. a dependent NP containing all modifiers such as adjoined relatives clauses, etc., was taken.

3.2 Interaction of indicators with syntactic constructions

The syntactic features dative and accusative interact with certain "argument-structure-changing" constructions found in the corpus, which will impact on parsing. Basically, we encounter two types of construction that interfere with the argument structure of the verb: constructions that "reduce" the argument structure at the surface and constructions that seemingly "extend" it.

Argument structure reduction. The diatheses passive and middle, as well as null instantiations[4] (NI) "reduce" the argument structure of the verb. In passives (eventive, stative, impersonal, etc.) the subject of the verb is demoted and the direct object surfaces as subject, marked with nominative case. Eventive passives which are indicated with the auxiliary *werden* are identifiable, however other types of passives such as adjectival passives, which are identical with predicative adjective constructions, cannot be identified by the parsers. Middles, e.g. (9), behave similar to the passive with respect to subject and object. In middles, the internal object, which is marked with accusative in the standard active form, surfaces as nominative. The true external subject is demoted. However, middles are usually not identifyed by automatic parsers.

(9) Die Rumba tanzt sich leicht nach.
the rumba dances REFL easy after
"The rumba dances easily."

Another issue that arises is the pragmatically context-driven demotion of arguments. In a sentence such as (10-a), a direct object is clearly expected for verb *schärfen* ('to sharpen') to

[2] In the original gold standard set, there were 277 sentences. However, seven did not contain a *nach*-PV. The parser erroneously identified *nach* and verb as a *nach*-PV; e.g. *Falten Sie das Papier der Länge nach* ('Fold the paper lengthwise').

[3] Significance according to Landis and Koch (1977).

[4] Cf. Fillmore et al. (2003).

predicate over. However, in operating instructions for things that need to be sharp and that are operated manually, e.g. a knife or a pair of scissors, it seem perfectly fine to leave out the direct object. Similarly, in (10-b), the theory would predict a dative argument of *nachtanzen*, which seems to be correct if the *nachtanzen*-clause would occur in isolation. However in this example, the dative can be omitted easily as the reference of the dancing manner can be identified locally in the same sentence with the dance instructor.

(10) a. Wenn Sie mehr Druck ausüben müssen, sofort Ø nachschärfen!
if you more pressure exert must immediately Ø.ACC after-sharpen
"If you have to exert more pressure, re-sharpen (the knife) immediately!"
b. Der Tanzlehrer tanzte genau vor und ich tanzte Ø nach.
the dance instructor danced exactly before and I danced Ø.DAT after
"The dance instructor showed how to dance and I followed his dancing manner."

Argument structure extension. On the other side, there are constructions that seemingly "extend" the argument structure: ACI-constructions (*Accusativus cum infinitivo*), dative benefactives, and accusative temporal adverbials increase the number of accusative- or dative-marked NPs recognized by a parser. These constructions mentioned above do not extend the argument structure of the verb, however they change the observable amount of argument-like phrases by raising the subject of an embedded clause to the object of the matrix clause (ACI-construction as in (11)), or by adding a noun phrase that could erroneously be identified as an argument of the verb (dative benefactive as in (12-a) and accusative temporal adverbial as in (12-b)).

(11) Ich höre die Glocke nachklingen.
I hear the.ACC bell after-sound
"I hear the bell linger on."

(12) a. Die Oma strickte dem Baby die Mütze nach.
the grandmother knitted the.DAT baby the cap after
"The grandmother made a knitted copy of the cap for the baby."
b. Die Banane reifte eine Woche nach.
the banana ripened a.ACC week after
"The banana continued ripening for one week."

Table 2b shows the agreement of the linguistically trained annotators on the valency-changing constructions and their frequencies in the gold standard, where they affect 32.49 % of the *nach*-PV instances. Dative benefactives and accusative temporal adverbials are to be added.

a.

indicator	κ (all annot.)	κ (trained annot.)
accusative	0.699	0.967
dative	0.869	0.985

b.

phenomenon	κ	frequency
passive	0.971	67
middle	1.000	4
ACI	1.000	4
NI	0.774	15

Table 2: Inter-annotator agreement on a. indicators and b. valency-changing constructions

3.3 Manual *nach*-PV classification

To obtain a gold standard of the *nach*-PV lemmas with respect to the five *nach* readings described above, three annotators, who were familiar with the classes by Haselbach (2011), manually classified 475 *nach*-PV lemmas that where extracted from the corpus. Each annotator decided for each lemma (without context) if it exhibits one particular *nach*-reading, e.g. DIR. If at least two annotators labeled a lemma with a reading, the lemma was accounted to the *nach*-reading (majority decision). Multiple labels for one lemma were allowed (reflecting polysemy).

Table 3 shows the verb classes and verb class combinations of the *nach*-PV lemmas identified by the annotators. 246 of 475 *nach*-PV lemmas were classified as belonging to at least one of the five classes described by Haselbach (2011). Table 3 also illustrates that a manual classification of the *nach*-PV lemmas without context is a non-trivial task. Albeit the majority of *nach*-PV lemmas annotated with at least one of Haselbach (2011)'s classes clearly belong to one class (cf. rows 1 to 5 in Table 3) – or at least, as expected, to the mixed class {MAN,CRE} – there are many verbs that show different *nach* readings. This, of course, is not fatal for the theory, on the contrary, it is expected. However, for the automatic identification this might pose a problem. The ranks 1 to 5 (more than 10 lemma types) mostly cover single classes. Under the top-5 there is only one complex class, the MAN/CRE-class, which is expected as discussed in Section 2.1. Examples of the classes that are classified as expected are: *nachfetten* ('to regrease') or *nachsalzen* ('to add more salt') for OMR; *nachlaufen* ('to run after sb.') or *nachblicken* ('to gaze after sb.') for DIR; *nachbacken* (lit.: 'after'+'bake', to copy sth. by baking it/to copy sb.'s baking of sth.) for {MAN,CRE}; or *nachbluten* (lit.: 'after'+'bleed', to continue bleeding) for CONT.

Nevertheless, the verbs in the tail of list, i.e. from rank 6 on, seem randomly distributed. This shows that an *ad hoc* classification of the verbs without context is difficult because annotators might come up spontaneously with a reading that is rather coerced than predicted, or the other way around they might miss a reading because it is less common but perfectly grammatical. The unbalanced distribution of the verbs in the lower part of the list therefore also indicate that there might be verbs in the upper, apparently clear part of the table that might also be coerced to a particular reading of *nach*, and it might also have happened by accident that the annotators did not come up with a particular reading.[5]

rank	reading class	frequency	rank	reading class	frequency
1	OMR	67		{MAN,CRE,OMR}	6
2	DIR	58	9	{DIR,CONT}	5
3	{MAN,CRE}	26	10	{CRE,OMR}	4
4	MAN	20	11	{DIR,CRE}	3
5	CONT	19		{DIR,OMR}	3
6	{MAN,OMR}	8	12	{DIR,OMR,CONT}	2
	{DIR,MAN,CRE}	8		CRE	2
7	{DIR,MAN}	7	13	{MAN,CRE,CONT}	1
8	{OMR,CONT}	6		{DIR,MAN,CRE,CONT}	1
				overall	246

Table 3: Manual *nach*-PV classification

Additionally, the inter-annotator agreement of the classification is poor, cf. Table 4. The probability that all three annotators agreed on a particular class is not bad (P), however, as the distribution of the *nach*-readings over the lemmas is rather unbalanced the probability of an accidental match with the binary features is high (Pe). Thus κ is poor.

4 Tools and analyses of corpus data

To identify different classes of *nach*-PVs based on the corpus data, we make use of two dependency parsers and a relational database infrastructure (cf. Eckart et al., 2010), to extract

[5] An example in the data set is *nachlabern* (lit.: 'after'+'babble', paraphrase: derogatively reciting sth. in a babbling manner), which is labeled as exclusively belonging to the MAN class. A careful scrutiny however shows that *nachlabern* can also occur with the CRE reading.

nach-reading	P	Pe	κ
DIR	0.889	0.689	0.642
MAN	0.716	0.642	0.205
CRE	0.865	0.774	0.403
OMR	0.755	0.626	0.346
CONT	0.876	0.811	0.344

Table 4: Inter-annotator agreement on *nach* reading wrt. lemmas

syntactic features appearing with the *nach*-PVs. We chose two parsers based on different concepts to complement one another, i.e. by taking the individual strengths of each parser into account when extracting indicators from their results.

The **Bohnet-Parser** (BP), cf. Bohnet (2010), is a data-driven state-of-the-art dependency parser. It makes use of a rich feature model and a second order maximum-spanning-tree algorithm (McDonald and Pereira, 2006). The parser also includes its own processing pipeline containing statistical lemmatization, part-of-speech tagging, and morphological tagging on an already tokenized input. The output structure are non-projective dependency trees in the tabular representation format of CoNLL 2009's shared task. Regarding labeled syntactic accuracy in this shared task, the Bohnet-Parser was the second best system and the best system for English and German. The model we utilized in our experiments was trained on a dependency version of the German TiGer treebank (Brants et al., 2002), as described in Seeker and Kuhn (2012).

FSPar (FP), cf. Schiehlen (2003), is a rule-based dependency parser based on the approach by Abney (1996) of partial parsing by finite state cascades. FSPar also processes its own internal pipeline, which includes lexically informed tokenizing and lemmatizing and part-of-speech tagging with the TreeTagger, cf. Schmid (1994). Not only the tokenizing step but also the parsing makes use of a large integrated lexical knowledge base, e.g. including named entities. FSPar generates underspecified dependency graphs. Underspecification is applied with respect to head selection as well as dependency labels. If the attachment to a head is ambiguous for a specific token or if the head is part of a coordination or is a combined verb form, more than one head token is given. Multiple or underspecified dependency labels occur either because of mulitple head possiblilties or because of ambiguous dependency relations.

4.1 Parsing results: recognizing the indicators

Both parsers were evaluated against the test set described in Section 3.1. Three evaluation criteria were applied, each time taking all 277 sentences into account. The first evaluation is on *nach*-PV recognition, including cases where a sentence contains more than one or no *nach*-PV. The sentences that contain no *nach*-PV are difficult for the parsers as they contain the token *nach* and a verb for which the *nach* could denote a separable verb particle but does not so in the sentence under analysis. The other two criteria are the recognition of dative and accusative arguments, where in a correct case the parser extracted an argument of the expected case, and its head is contained in the argument string represented in the gold standard.

Table 5 shows the results of the evaluation. For the Bohnet-Parser, verbs are identified as *nach*-PVs, in case of *nach* being a part of the lemma, or having a token *nach* being the dependent of the verb, where the dependency label or the part-of-speech tag identified a separable verb particle. Accusative and dative arguments are identified by the respective dependency label on a relation having the *nach*-PV as its head.

Bohnet-Parser's *nach*-PV recognition suffers from the fact that the statistical lemmatizer sometimes produces a wrong lemma, even if the right token is recognized as a *nach*-PV. Dative recognition outperforms accusative recognition because dative arguments are morphologically marked more clearly. So although the Bohnet-Parser recognizes a dative only with a recall of 56.67, if it does annotate one, it is mostly correct with a precision of 87.18.

	Bohnet-Parser				FSPar		
	prec	rec	f1		prec	rec	f1
NPV recognition	93.62	95.65	94.62		97.53	100.0	98.75
dative recognition	87.18	56.67	68.69	upper bound	61.11	91.67	73.33
				lower bound	50.00	75.00	60.00
				chance	52.22	78.33	62.67
accusative recognition	53.12	67.11	59.30	upper bound	46.21	88.16	60.63
				lower bound	26.90	51.32	35.29
				chance	32.41	61.84	42.53

Table 5: Bohnet-Parser and FSpar on 277-sentences gold standard

Nach-PV recognition has a recall of 100.0 for FSPar, which is due to the facts that (i) we used the parser to identify sentences containing at least one *nach*-PV, so FSPar introduced all sentences including the sentences which contain no *nach*-PV and (ii) FSPar makes use of a huge lexicon so the correct lemma does not have to be generated. Due to the underspecified output of FSPar, three values are given for its argument evaluations. The first value results from a credulous interpretation of the underspecified output, constituting an *upper bound*, which takes all cases into account, where the right annotation could be derived from the underspecified one. The second value results from a strict interpretation and introduces a *lower bound*, as only those cases have been counted as correct where the right annotation was the only annotation by FSPar. The third value, called *chance* in the table, was calculated by randomly choosing an annotation from those proposed by FSPar.

4.2 More reliability by voting-based combination

To extract reading indicators from the large data set, we make use of our findings on the test set. The quality of the accusative and dative argument recognition is seen as an approximation of the overall parsing quality relative to the task of indicator identification. To extract more reliable indicators, we combine the results of FSPar and the Bohnet-Parser. By this approach, we trust in the hypothesis that the combined result exceeds the best single result, which has been confirmed for a set of speech recognition systems by Fiscus (1997), for a set of constituency-based parsers by Henderson and Brill (1999) and for a set of dependency parsers by Zeman and Žabokrtský (2005), among others. As our combination is task specific, we do not make use of complex heuristics or approaches handling the complete parse, but define some extraction rules based on the evaluation results from Section 4.1.

Tables 6 show the combination rules which we applied in the indicator extraction. Due to the underspecification representation of FSPar the combination rules distinguish between the two binary features "*nach*-PV has an argument of case X", where X is in {DAT,ACC}, and the feature denoting the argument form. Tables 6a and b show the combination rules for the binary features. $\oplus$DAT and $\oplus$ACC denote that the parser identified an argument in the respective case. For FSPar, this is split up in cases where the output of FSPar is not underspecified, i.e. where there was only a single result (*s*) and cases where the respecive annotation was one possibility

in a set of multiple annotations (*m*). ⊖DAT and ⊖ACC denote that the parser did not identify an argument of the respective case. In cases where both parsers agree, no further rule has to be applied. Where they disagree, we decided for the following rules depending on the results of the evaluation from Section 4.1:

a.

FP \ BP	⊕DAT	⊖DAT
⊕DAT **s**	⊕DAT	⊕DAT
m	⊕DAT	⊕DAT
⊖DAT	⊕DAT	⊖DAT

b.

FP \ BP	⊕ACC	⊖ACC
⊕ACC **s**	⊕ACC	⊖ACC
m	⊕ACC	⊕ACC
⊖ACC	⊖ACC	⊖ACC

c.

FP \ BP	⊕DAT	⊕ACC	0
⊕DAT **s**	⊕DAT	⊕DAT	⊕DAT
⊖ACC	⊕DAT	⊕DAT	0
⊕ACC **s**	⊕ACC	⊕ACC	⊕ACC
⊖DAT	⊕DAT	⊕ACC	0
⊕DAT ⊕ACC	⊕DAT	⊕ACC	0
0	⊕DAT	⊕ACC	0

Table 6: Combination rules for a. datives and b. accusatives per annotation and c. per argument

As the precision of Bohnet-Parser's dative recognition is high (87.18), we take the result of the Bohnet-Parser, whenever it identifies a dative. Regarding recall however, FSPar's result exedes the Bohnet-Parser even in the lower bound and has a very high value for the upper bound (91.67) which is an important value for the binary feature. Therefore we only decide for ⊖DAT in cases where both parsers agree on that.

For arguments in accusative case, the only high value in the evaluation was the upper bound recall of FSPar (88.16). This is not surprising, as arguments in accusative case are highly case syncretistic in German. We decided thus to let FSPar overwrite the decision of the Bohnet-Parser in the cases where an argument in accusative case was among the results of FSPar or where FSPar did not recognize an argument in accusative case at all. In cases where the Bohnet-Parser did not identify an argument in accusative case, but FSPar did so explicitly, i.e. the single case, we do not let FSPar overwrite because the values of FSPar's lower bound, which take exactly those single cases into account, are very low ($f1 = 35.29$).

Table 6 shows the rules we utilized in extracting the argument form. There we have the cases that the Bohnet-Parser specifies a form to be an argument in dative case (⊕DAT), in accusative case (⊕ACC), or neither (0). FSPar identifies a form as single dative (⊕DAT, *s*), as underspecified dative but from a representation of which no accusative can be derived (⊕DAT, ⊖ACC), as single accusative (⊕ACC, *s*), as underspecified accusative but no dative (⊕ACC, ⊖DAT), as underspecified between at least dative and accusative (⊕DAT, ⊕ACC), or as none of these (0).

Again we trust the dative recognition of each parser, which is only overwritten in two cases: (i) FSPar rules out a dative in favor of a single accusative and (ii) Bohnet-Parser neither annotates accusative nor dative and FSPar states the dative in a multiple result. These decisions deviate from the ones in the binary features. The latter decision takes into account that now a particular argument has to be decided upon and identifiying the respective dative argument drops FSPar to the chance value. The first decision is a more debatable one, and serves to balance the decisions for accusative arguments which are overall more frequent. The same applies for the other cases, in which FSPar proposes an accusative, while the Bohnet-Parser does not. If FSPar proposes ⊕DAT and ⊕ACC, or neither of them, we opt for the Bohnet-Parser. For all other clustering features, we use the Bohnet-Parser output as long as FSPar does not contradict it.

5 Automatic clustering of *nach*-PV lemmas into semantic classes

5.1 Experimental setup

As input data, we use the 270k sentences extracted from the SDeWaC corpus, which are automatically parsed by the Bohnet-Parser and FSPar. From these automatic syntactic structures we then extract five different types of features starting with the presence of dative and accusative arguments. In order to see the effect of features beyond argument structure, as described in Section 2.2, we successively add four other types of features: (i) the word form of the dative/accusative arguments, (ii) a combination of preposition form, case value, and prepositional object form for each prepositional dependent of the verb, (iii) the word form of adverbials depending on the verb, and (iv) the presence of clausal objects.

We use Ward's algorithm (Ward, 1963) to produce the clustering of the verbs in our gold standard with the number of output clusters set to 18.[6] In order to evaluate our clusters, we use the v-measure proposed in Rosenberg and Hirschberg (2007), which is defined as the harmonic mean of homogeneity and completeness. Both metrics are defined based on entropy of the clusters, where homogeneity measures the distribution of gold standard classes within each cluster and completeness measures the distribution of clusters within each gold standard class.[7]

features (added up)	homogeneity		completeness		v-measure	
	BP	FSPar	BP	FSPar	BP	FSPar
	combined		combined		combined	
dat,acc	32.96	37.62	28.20	30.11	30.39	33.45
	38.23		30.23		33.76	
⊕ datform,accform	33.24	33.53	30.27	28.19	31.68	30.62
	36.04		29.82		32.46	
⊕ pp-form-case-pobjform	34.40	35.18	30.65	29.56	32.42	32.13
	36.08		31.32		33.53	
⊕ adverbials	35.78	37.77	33.56	32.11	34.64	34.71
	41.76		34.89		38.02	
⊕ clausal objects	39.56	39.31	35.18	33.30	37.24	36.05
	41.49		35.09		38.02	

Table 7: Clustering: features extracted by Bohnet-Parser, FSPar, and their combination

5.2 Results and evaluation against human gold standard

Table 7 shows the result of the clustering broken down for the individual parsers as well as their combination. Focusing on the results for the Bohnet-Parser, we see that all features successively add to the overall performance. This shows that the additional features also contribute information to the formation of verb classes. If we check the results for FSPar and the combined feature extraction, we see a drop in performance when the word forms of the dative and accusative arguments are added, which is then compensated by the other three features which improve performance. The highest value is achieved if the combined feature

[6] Verbs can be in more than one of the theoretically predicted classes. Since the clustering algorithm does not allow for an instance to be in more than one cluster, we assume each combination of theoretical readings to be one class in the clustering. We however only consider those that appear in the gold standard and we also remove all verbs that belong to neither of the five *nach*-PV classes.

[7] The ideal case for homogeneity occurs when there is only one single class in each cluster whereas for completeness, the ideal case occurs when there is only one cluster for each of the gold standard classes.

extraction is applied. The reason for the drop for the first features is the ambiguous output of FSPar, that possibly adds more word forms to the features for the clustering than are actually correct. This is due to the fact that oftentimes FSPar gives more than one possibility for the dative or accusative argument and it is not possible to automatically choose between them without first disambiguating them. In the combined system, the drop is much smaller than for FSPar alone, because we can use the Bohnet-Parser to restrict the options that FSPar offers.

nach-reading	frequency	homogeneity	completeness	v-measure
CONT	10	78.53	66.30	71.90
{MAN,CRE}	24	42.82	58.45	49.43
OMR	51	45.35	44.82	45.08
MAN	16	28.02	56.85	37.54
DIR	51	37.05	37.19	37.12
{MAN,CRE,CONT}	1	65.95	100.00	79.48
{DIR,MAN,CRE,CONT}	1	44.05	100.00	61.16
{DIR,OMR,CONT}	2	45.17	86.83	59.43
{DIR,MAN}	4	44.32	78.85	56.75
CRE	1	39.61	100.00	56.75
{DIR,CRE}	2	40.59	86.83	55.32
{DIR,MAN,CRE}	5	44.53	73.28	55.40
{DIR,CONT}	3	38.95	79.12	52.20
{DIR,OMR}	3	36.65	79.12	50.09
{CRE,OMR}	3	33.17	86.74	47.99
{OMR,CONT}	5	31.96	79.36	45.56
{MAN,OMR}	7	31.72	72.20	44.08
{MAN,CRE,OMR}	4	25.58	73.66	37.98
overall	193	41.49	35.09	38.02

Table 8: Clustering: detail analysis: Bohnet-FSPar combination and all features

The other finding from the evaluation is the rather low overall performance. With a best value of 38.02 % v-measure it seems that we simply cannot recreate the classes that our theory predicts. However, it is worth taking a closer look at the results for the individual classes, since it turns out that the clustering quality varies greatly with the class that we are trying to produce. Table 8 shows the results of the clustering using the combined system, broken down for the individual classes. The table also splits the classes into high frequency and low frequency classes, showing the bigger classes first. As can be seen from the results, there are five classes that contain at least 10 lemmata, the biggest of them containing 51. It turns out that for three of them, we get much higher results than expectable from the average score (up to 35 % better for the CONT class). For the second and third class, we get results a little below 50 %, but for the MAN and DIR class, we get results slightly below the average. For the smaller classes with size less than 10, we mostly get results higher than the average, but these classes are too small to be really conclusive. In summary, we find that while the clustering works reasonably well for three of the bigger classes, the results are unsatisfactory for the other two. An explanation for the inferior performance on the MAN class can be found in the theory: as we discussed already in Section 2.1.3, the classes MAN and CRE are very closely related and could be considered one class since it is often not easy to distinguish the two meanings. The clustering seems to have similar problem singling them out. The performance on the DIR class can to a certain extent be explained by the argument structure of these verbs. As we show in Table 1, a DIR verb can have an accusative argument even though it would normally be avoided. That means that some of the features

that are available to the clustering are less informative than for other classes. As a point in case consider the CONT class, which comes out very nicely in the clustering. As shown in Table 1, this class has the most distinct argument distribution compared to all other classes.

One should also keep in mind that we are working with a tool chain of automatic tools that itself has several drawbacks, which influence the performance of the clustering. This includes the quality of the parsers, which although being state-of-the-art are still far from being perfect, and also the clustering algorithm, which can make incorrect decisions. Finally, one needs to take into account that our gold standard is not optimal because of the difficulties annotating our five verb classes.

Results and interpretation of the experiments: lessons learned

Theoretical results. The *nach*-PV class we were able to identify most precisely is the continuation class (CONT) for which no dative or accusative is predicted, cf. also Table 1. As assumed in Section 2.1.3, we saw that the theoretically motivated *nach*-PV classes MAN and CRE empirically collapse. We can conclude that argument structure, among others, is a fairly good indicator for automatically identifying the combined class {MAN,CRE}. However for the individual classes MAN and CRE, as well as for the directional class DIR, this is not the case. Here, more research is needed to pin down clear criteria for the selection of these classes.

Technological results. Regarding the NLP-methodology, our findings mainly address three topics: the quality of the single parsers, the combination of the parsing results and the application of a small high qualitiy sample as an approximation for the complete data set. We started with two single parsers, each evaluated against a small gold standard for *nach*-PV argument structures. Both parsers fell below expectations. The results of the state-of-the-art Bohnet-Parser decreased due to the unrestricted all-domain data. And FSPar, which preserves underspecified structures, did not reach an f-score higher than 73.33 even in the upper bound. While already the identification of *nach*-PVs was difficult due to sentences containing no *nach*-PV and due to complex lemmas, the creation of the gold standard also showed a lot of valency-changing constructions, most of them being difficult to annotate for the parsers. As we expect the gold standard to be representative for the whole data set, these difficult constructions should be found there in a similar distribution. This definitely has an impact on the features used in the clustering and therefore on the clustering results. So even by applying the best tools available their performace leaves much room for improvement. Nevertheless the parser combination showed the expected effect as the v-measure for the features from parser combination always exceeds the best single result. This also supports the applicabillity of the gold standard as approximation of the data set, as the combination rules were based on it.

Outlook. To benefit from this findings, we intend to add more parsers to the result combination and for example apply a majority voting scheme. While the two parsers we utilized were applicable right away, it might seem a good idea to complement our parsers with parsers that can be adapted to the task. Furthermore, one could try to apply other clustering techniques, e.g. fuzzy clustering to give greater emphasis on the fact that one lemma can be found in more than one class. Concerning the low agreement on the *nach*-PV lemma classification, we think that one could improve this by either taking more annotators into account or by classifying the *nach*-PV lemmas in expected and unexpected (coerced) contexts for each reading, and then measure their acceptability. This would rather approximate the conjectured continuum-like character of the distribution of the *nach* readings over the verb lemmas.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) via Projects B3, B4 and D8 of the SFB 732 "Incremental Specification in Context".

References

Abney, S. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344.

Baroni, M. and Kilgarriff, A. (2006). Large linguistically-processed web corpora for multiple languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2006*, pages 87–90, Trento.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING) 2010*, pages 89–97, Beijing.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Eckart, K., Eberle, K., and Heid, U. (2010). An infrastructure for more reliable corpus analysis. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (LREC 2010)*, pages 8–14, Valletta. European Language Resources Association (ELRA).

Fillmore, C. J., Johnson, C. R., and Petruck, Miriam R. L. (2003). Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354.

Halle, M. and Marantz, A. (1993). Distributed Morphology and the pieces of inflection. In Hale, K. and Keyser, S. J., editors, *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, volume 24 of *Current Studies in Linguistics*, pages 111–176. MIT Press, Cambridge, MA.

Haselbach, B. (2011). Deconstructing the German verb particle *nach* at the syntax-semantics interface. In Baunaz, L., Bentea, A., and Blochowiak, J., editors, *Generative Grammar in Geneva (GG@G) 7*, pages 71–92. Department of Linguistics, University of Geneva, Geneva.

Henderson, J. C. and Brill, E. (1999). Exploiting diversity in natural language processing: Combining parsers. In *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing (EMNLP-99)*, pages 187–194, College Park, Maryland.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, Dordrecht/Boston/London.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Larson, R. K. (1990). Double object revisited: Reply to Jackendoff. *Linguistic Inquiry*, 21(4):589–632.

McDonald, R. T. and Pereira, F. C. N. (2006). Online learning of approximate dependency parsing algorithms. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.

Nicol, F. (2002). Extended VP-shells and the verb-particle construction. In Dehé, N., McIntyre, A., Jackendoff, R., and Urban, S., editors, *Verb-Particle Explorations*, volume 1 of *Interface Explorations*, pages 165–190. Mouton de Gruyter, Berlin.

Rosenberg, A. and Hirschberg, J. (2007). V-Measure : A conditional entropy-based external cluster evaluation measure. In *Empirical Methods of Natural Language Processing (EMNLP) 2007*, number June, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Roßdeutscher, A. and Kamp, H. (2010). Syntactic and semantic constraints on the formation and interpretation of *-ung*-nouns. In Rathert, M. and Alexiadou, A., editors, *The Seamntics of Nominalizations across Languages and Frameworks*, volume 22 of *Interface Explorations*, pages 169–214. Mouton de Gruyter, Berlin.

Schiehlen, M. (2003). A cascaded finite-state parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2003*, pages 163–166, Budapest.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester.

Seeker, W. and Kuhn, J. (2012). Making ellipses explicit in dependency conversion for a german treebank. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Springorum, S., Schulte im Walde, S., and Roßdeutscher, A. (2012). Automatic classification of German *an* particle verbs. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 73–80, Istanbul, Turkey.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Zeman, D. and Žabokrtský, Z. (2005). Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 171–178, Vancouver, British Columbia. Association for Computational Linguistics.