Computational Linguistics in the Netherlands Journal 4 (2014) 149-170 Submitted 06/2014; Published 12/2014

Looking for Cluster Creepers in Dutch Treebanks.
Dat we ons daar nog kunnen mee bezig houden.

Liesbeth Augustinus LIESBETH@CCL.KULEUVEN.BE
Frank Van Eynde FRANK@CCL.KULEUVEN.BE

Centre for Computational Linguistics, KU Leuven, Belgium

Abstract

In Dutch V-final clauses the verbs tend to form a cluster which cannot be split up by nonverbal material. However, Haeseryn et al. (1997) as well as other studies on the phenomenon list several cases in which the verb cluster may be interrupted by *cluster creepers.* The most common examples are constructions with separable verb particles, but examples with nouns, adjectives, and adverbs are attested as well.

Since the majority of the data in previous studies is collected by introspection and elicitation, it is interesting to compare those findings to corpus data. The corpus analysis is based on data from two Dutch treebanks (CGN and LASSY), which allow to take into account regional and/or stylistic variation. This is an important aspect for the analysis, since cluster creeping is reported to be a typical property of spoken and regional variants of Dutch.

The goal of this corpus-based investigation is on the one hand to provide insight in the frequency of the phenomenon, and on the other hand to classify the types of cluster creepers. Besides the linguistic analysis, methodological issues regarding the extraction of the relevant data from the treebanks will be addressed as well.

1. Introduction

1.1 Dutch Clause Structure

There are two fixed positions in the Dutch sentence. Those positions are known as *poles.* In verb-initial clauses, such as example (1a), the finite verb *heeft* 'has' occupies the first pole, while the past participle *gedronken* 'drunk' is in the second pole.[1] In subordinate clauses, such as example (1b), the complementizer *dat* 'that' takes up the first pole, while the verbal elements *beschouwd wordt* 'is considered' occupy the second pole. (1b) shows that the second pole may consist of multiple elements, but it can also be empty, as in example (1c) (Haeseryn et al. 1997, pp.1225-1226).

(1) a. Z'n broer *heeft* altijd al graag een glas bier *gedronken.*
his brother has always already gladly a glas beer drunk
'His brother has always enjoyed a glass of beer.'

b. (Het blijkt) *dat* hij zowat overal ter wereld als een autoriteit *beschouwd wordt.*
it seems that he almost everywhere in the world as an authority considered is
'(It seems) that he is considered to be an authority almost all over the world.'

c. Z'n broer *drinkt* graag een glas bier.
his brother drinks gladly a glass beer
'His brother likes to drink a glass of beer.'

The poles divide sentences into *topological fields*: The *voorveld* is the part before the first pole, the *middenveld* is the part between the poles, and the *naveld* is the part after the second pole.

The sequence of verbs in the second pole is called the *werkwoordelijke eindgroep* 'lit: verbal end group' or *verb cluster.* The nonverbal elements appear before or after the second pole (2a-b).

[1] Dutch verb-initial clauses comprise V-first and V-second clauses.

The intrusion of nonverbal material in the verb cluster, as in (2c), is considered ungrammatical. Canonically, nonverbal elements are not allowed in the verb cluster (Haeseryn et al. 1997, p.1355).

(2) a. (Hij beweerde) dat hij het gisteren aan de leraar *had verteld*.
he claimed that he it yesterday to the teacher had told

b. (Hij beweerde) dat hij het gisteren *had verteld* aan de leraar.
he claimed that he it yesterday had told to the teacher
'(He claimed) that he told the teacher about it yesterday.'

c. * (Hij beweerde) dat hij het gisteren *had* aan de leraar *verteld*.
he claimed that he it yesterday had to the teacher told

1.2 Cluster Creepers

Although the impenetrability of the verb cluster is the norm in most constructions, there are some exceptions. Example (3) shows a construction in which the verb cluster is interrupted by the adjective *schuldig* 'guilty', but which is nonetheless well-formed.

(3) De verdachte ontkent tot op heden zich aan zwendel te hebben *schuldig* gemaakt.
The suspect denies until at present himself to fraud to have guilty made
'Until today the suspect denies being guilty of fraud.'

According to Haeseryn et al. (1997), instances of *cluster creeping* occur more often in Belgian Dutch compared to Dutch spoken in the Netherlands.

1.2.1 A typology of cluster creepers

Haeseryn et al. (1997) mention three types of *cluster creepers*:

1. The most typical cluster creepers are *inherent* parts of the verb phrase, such as predicative adjectives and nonverbal parts of idiomatic expressions. Usually, those elements occur just before the second pole, as in (4a), but they can also be included in the verb cluster, as in (4b) (Haeseryn et al. 1997, p.1358). Note that (3) is also an example of this type.

   (4) a. ... dat hij zich niet *bang* zal laten maken.
   ... that he himself not afraid will let make

   b. ... dat hij zich niet zal laten *bang* maken.
   ... that he himself not will let afraid make
   '... that he will not be frightened.'

2. A second category of cluster creepers consists of stranded adpositions, often being the second part of pronominal adverbs. Canonically those adpositions are realised before the verb cluster (5a), but they may also occur within the cluster (5b).

   (5) a. ... dat hij daar nog *aan* moet denken.
   ... that he there still on must think

   b. ... dat hij daar nog moet *aan* denken.
   ... that he there still must on think
   '... that he still needs to think about that.'

   This type of *adposition stranding* within the cluster is considered typical of Belgian Dutch (Haeseryn et al. 1997, p.1362).

3. A third type that is also typical of Belgian Dutch but less common than adposition stranding is cluster creeping by an object or an adverbial modifier (Haeseryn et al. 1997, p.1362):

(6) a. ... dat de Rode Duivels nog *twee doelpunten* moeten scoren.
... that the Red Devils still two goals must score
b. ... dat de Rode Duivels nog moeten *twee doelpunten* scoren.
... that the Red Devils still must two goals score
'... that the Red Devils still need to score two goals.'

Haegeman and van Riemsdijk (1986) discuss several constructions for West-Flemish, a regional variant of Dutch spoken in Belgium, such as (7a). Most speakers consider the corresponding construction in (Standard) Dutch ungrammatical (7b). What differentiates (7b) from (6b) is the presence of a determiner: While cluster creeping by bare nominals is more common, NPs with a determiner are rarely used in the verb cluster.

(7) a. WF ... da Jan wilt *een hus* kopen.
... that Jan wants a house buy
b. DU * ... dat Jan wil *een huis* kopen.
... that Jan wants a house buy
'... that Jan wants to buy a house.'

Besides *genuine* cases of cluster creeping, Haeseryn et al. (1997) mention several constructions that look like cluster creeping but should not be treated as such. For example, separable verb particles (SVPs) are not considered as cluster creepers if they occur within the verb cluster. They argue that in the case of SVPs, constructions in which the SVP is realised in front of the verb cluster (8a) are less preferred than constructions in which the SVP is realised within the cluster (in front of the main verb or as a part of it), as in (8b) (Haeseryn et al. 1997, pp.1357-1358).[2]

(8) a. ... dat hij haar *op* moet bellen.
... that he her up must call
b. ... dat hij haar moet *op*bellen.
... that he her must up-call
'... that he must call her.'

The fact that Haeseryn et al. (1997) do not treat SVPs as real cluster creepers as opposed to inherent sentence parts leads to classification problems, since the distinction between SVPs and inherent parts of the sentence is often hard to draw (Haeseryn et al. 1997, p.1359). Consider for example *koffiedrinken* 'drink coffee' versus *champagne drinken* 'drink champagne'. Are those examples separable verbs or regular combinations of a verb and a noun?

In order to avoid this uncertainty, we will treat both SVPs and inherent parts of the verb phrase as cluster creepers, which is in line with amongst others Evers (2003) and Wurmbrand (2005).

1.2.2 Position of the cluster creepers

Cluster creeping is only possible if the main verb does not occur at the front of the cluster, since the nonverbal element cannot occur after the main verb, as shown in (9).

(9) a. * ... dat hij gedronken *koffie* heeft.
... that he drunk coffee has
Intended: '... that he has drunk coffee.'
b. * ... dat hij drinken *koffie* wil.
... that he drink coffee wants
Intended: '... that he wants to drink coffee.'

[2] Haeseryn et al. (1997) consider constructions like (8a) typical of spoken (Netherlandic) Dutch.

Therefore, cluster creeping occurs more often in infinitive constructions than in constructions with a participle, since infinitives are usually realised at the end of the verb cluster, as opposed to participles (Haeseryn et al. 1997, pp.1355-1356). See also Hoekstra (2010, pp.178-179) for a discussion on the relation between verb order within the cluster and cluster creeping.

The canonical position of a cluster creeper is just before the main verb, but in clusters with more than two verbs it may also occur more to the front of the verb cluster, as in (10) (Haeseryn et al. 1997, p.1357).

(10) ... dat hij haar had *op* moeten bellen.
... that he her had up must call
'... that he had to call her.'

2. Goals

The methodology used for this research is corpus-based, in the sense that treebanks (i.e. syntactically annotated corpora) will be used to verify the claims about cluster creeping. This corpus-based investigation will provide insight in the frequency of the phenomenon, which makes it possible to compare constructions that are theoretically possible to the constructions that are *actually* used. More specifically, we will classify the types of cluster creepers according to their syntactic function and their phrasal category or part-of-speech (POS) in order to investigate whether the types of cluster creepers mentioned in Haeseryn et al. (1997) are reflected in the corpus data, or whether the data reveal other categories, aiming at a more complete description of the possible cluster creepers in Dutch.

Furthermore, we will consider the occurrence of cluster creepers in spoken versus written language, as well as their occurrence in clusters containing participles versus clusters with infinitives.

Specifically, we aim at extracting and investigating corpus examples like the following:

(11) a. we hebben zo nog ne politieker die ons daar altijd ook doet *aan* denken.
we have so another a politician that us there always also does on think
'we have another polititian of that kind who always reminds us of that. [CGN, fvc701156_222]'

b. . . . aan iedereen die toen de toekomst van dit land, van de huidige en
. . . to everyone that then the future of this country of the present and
toekomstige generaties hebben *veilig* gesteld.
future generations has save put
'. . . to everyone who back then has saveguarded the future of this country, of the current and future generations.' [LASSY, dpc-vhs-000745-nl-sen.p.13.s.3]

Section 3 gives a formal definition of the concept 'verb cluster', and it provides an overview of the relevant constructions, i.e. constructions containing a verb cluster (which may contain cluster creepers). Section 4 describes the two treebanks used for the corpus study (CGN and LASSY). Section 5 explains how the queries are constructed in order to extract the relevant constructions. Section 6 presents and discusses the results of the treebank investigation. Those results largely confirm the claims made in Haeseryn et al. (1997), but they also contain a surprise, i.e. *multiple cluster creepers*, as in (12).

(12) ... dat we ons daar nog kunnen *mee bezig* houden.
... that we us there still can with busy keep
'... that we can still keep ourselves busy with that.'

Section 7 sums up the conclusions and points out some topics for future research.

3. Defining the verb cluster

In order to retrieve constructions with a cluster creeper, we first have to define precisely what a verb cluster is. Generalizing from the examples in section 1.1, we define a verb cluster as a sequence of two or more verbs in the second pole of the clause. The sequence is ordered in two ways. One concerns the order of selection. In *zou hebben gedronken* 'would have drunk', for instance, the finite modal auxiliary *zou* selects a bare infinitive, i.e. *hebben* 'have', which in turn selects a past participle, i.e. *gedronken* 'drunk'. The last verb in this chain is the 'main' verb. (13) defines the selection order in the cluster in general terms.[3]

(13) $(\mathrm{V}_{finite})\ (\mathrm{V}_{(te)inf})*\ (\mathrm{V}_{psp})$

In words, a cluster has at most one finite verb (modulo coordination), followed by 0, 1 or more bare and/or *te* infinitives, followed by at most one past participle (modulo coordination). Table 1 provides some examples of verb clusters. In verb-initial clauses ($\mathrm{V}_{initial}$) the cluster only contains non-finite forms, since the finite verb is in the first pole. In verb-final clauses (V_{final}) the cluster also contains the finite verb. The 'main' verb is a past participle, a bare infinitive or a *te*-infinitive. The clusters are in italics.

	$\mathrm{V}_{initial}$	V_{final}
Past Part	INF+PSP Hij zou gisteren koffie *hebben gedronken.* 'He would have drunk coffee yesterday.'	FINITE+INF+PSP . . . dat hij gisteren koffie *zou hebben gedronken.* '. . . that he would have drunk coffee yesterday.'
Bare inf	INF+INF Hij zal morgen koffie *willen drinken.* 'He will drink coffee tomorrow.'	FINITE+INF+INF . . . dat hij morgen koffie *zal willen drinken.* '. . . that he will drink coffee tomorrow.'
te-inf	INF+*te*-INF Hij heeft gisteren koffie *proberen te drinken.* 'He has tried to drink coffee yesterday.'	FINITE+INF+*te*-INF . . . dat hij gisteren koffie *heeft proberen te drinken.* '. . . that he has tried to drink coffee yesterday.'

Table 1: Verb clusters

The second way in which the sequences are ordered is the linear order. This order canonically coincides with the order of selection, as in *zou hebben gedronken* 'would have drunk' and the other examples in Table 1. Alternative orders are also possible, though. The finite verb may also occur as the last element in the cluster, as in *gedronken hebben zou.* The past participle can occupy any position within the cluster, e.g. *zou gedronken hebben* 'would drunk have', *gedronken zou hebben* 'drunk would have'.

Clauses with a *te*-infinitive are tricky, since this infinitive may either be the last member of the cluster or the first member of the *naveld.* The former is invariably the case if its selector is a so-called *Infinitivus Pro Participio* (IPP), i.e. an infinitive which is selected by the perfect auxiliary.[4] Relevant examples are given in the last row of Table 1. Notice that the selector of the *te*-infinitive is the IPP *proberen*, which in its turn is selected by the auxiliary of the perfect. These examples can be contrasted with those in (14), where the perfect auxiliary is combined with the (expected) past participle.

(14) a. Hij heeft gisteren *geprobeerd* koffie te drinken.
he has yesterday tried coffee to drink
'He has tried to drink coffee yesterday.'

[3] For a comprehensive list of the verbs which can take a nonfinal position in the cluster, see Augustinus and Van Eynde (2012).

[4] The name IPP captures the fact that such auxiliaries normally require a past participle.

b. . . . dat hij gisteren *heeft geprobeerd* koffie te drinken.
. . . that he yesterday has tried coffee to drink
'. . . that he has tried to drink coffee yesterday.'

In these clauses the *te*-infinitive is in the *naveld.*

Independent evidence for the distinction is provided by the reordering possibilities. While *te*-infinitives which are part of the verb cluster may appear in other positions than the last one, the *te*-infinitives in the *naveld* must follow those which are part of the cluster.

(15) a. . . . dat hij gisteren koffie *proberen te drinken heeft.*
. . . that he yesterday coffee try to drink has
'. . . that he has tried to drink coffee yesterday.'

b. * . . . dat hij gisteren koffie geprobeerd te drinken heeft.
. . . that he yesterday coffee tried to drink has

This criterion is also applicable to combinations in which the selector of the *te*-infinitive is a finite verb, as in (16).

(16) a. . . . dat hij koffie probeerde te drinken.
. . . that he coffee tried to drink
'. . . that he tried to drink coffee.'

b. * . . . dat hij koffie te drinken probeerde.
. . . that he coffee to drink tried

The ungrammaticality of the second clause shows that the *te*-infinitive is in the *naveld.*

4. Data set

For the corpus study we use the CGN Treebank and LASSY Small. Those treebanks for respectively spoken and written Dutch each contain ca. one million tokens. As the corpora are more or less equal in size, they are suited for comparing written to spoken language data.

4.1 CGN

The Corpus Gesproken Nederlands (CGN) (Oostdijk et al. 2002) is an annotated corpus of spoken Dutch.[5] It consists of recorded speech which is orthographically transcribed, resulting in a corpus of ca. ten million words, of which one million is syntactically analysed. That syntactically annotated part of CGN will be referred to as the *CGN treebank.*

Two thirds of the corpus data consists of Dutch spoken in the Netherlands, whereas one third of the data comprises Dutch spoken in Flanders, the Dutch speaking part of Belgium. The corpus contains both dialogues and monologues, and is further divided into specific genres. The division into subcorpora allows to investigate stylistic variation (e.g. by comparing spontaneous conversations to news reports), as well as regional variation (by comparing Dutch spoken in Belgium to Dutch spoken in the Netherlands).

4.1.1 Contents

Table 2 presents the contents of the CGN treebank. The label N is used to refer to the Dutch data, while the label V refers to the Flemish data. The labels A to O refer to the different types of speech that the corpus comprises. The parts A to H contain dialogues, whereas the parts I to O consist of monologues. # Sentences refers to the number of sentences (or utterances) in each subcorpus; # Words refers to the number of words (excluding punctuation).

[5] `http://lands.let.ru.nl/cgn`

Components	# Sentences	# Words	# Sentences	# Words	# Sentences	# Words
	N		V		TOTAL	
A. Spontaneous conversations (’face-to-face’)	50,239	302,828	22,881	147,418	73,120	450,246
B. Interviews with teachers of Dutch	2,484	25,724	4,289	34,158	6,773	59,882
C. Telephone conversations (recorded via a switchboard)	11,649	70,084	3,142	19,984	14,791	90,068
D. Telephone conversations (recorded on MD)	0	0	929	6,309	929	6,309
E. Simulated business negotiations	3,123	25,524	0	0	3,123	25,524
F. Interviews/discussions/debates (broadcast)	6,290	75,167	2,617	25,122	8,907	100,289
G. (Political) discussions/debates/ meetings (non-broadcast)	1,166	25,125	543	9,009	1,709	34,134
H. Lessons recorded in the classroom	3,064	26,004	1,395	10,116	4,459	36,120
I. Live (sports) commentaries (broadcast)	2,251	25,002	1,026	10,147	3,277	35,149
J. Newsreports (broadcast)	2,259	25,084	536	7,686	2,795	32,770
K. News (broadcast)	1,923	25,353	558	7,306	2,481	32,659
L. Commentaries/columns/reviews (broadcast)	1,857	25,082	601	7,431	2,458	32,513
M. Ceremonious speeches/sermons	444	5,190	107	1,894	551	7,084
N. Lectures/seminars	593	14,921	701	8,159	1,294	23,080
O. Read speech	0	0	3,256	44,144	3,256	44,144
Complete corpus	**87,342**	**671,088**	**42,581**	**338,883**	**129,923**	**1,009,971**

Table 2: Contents of the CGN treebank

The word and sentence counts in Table 2 are based on the CGN Treebank version 2.0.1, converted to the Alpino-XML data format.[6]

Each sentence in the corpus has a unique identifier, e.g. [fva400392_6] for the sentence in (17).

(17) awel ’k ga ne keer een typisch voorbeeld geven.
well I go a time a typical example give
‘well, I’ll give a typical example.’ [CGN, fva400392_6]

The sentence ID refers to the origin of the fragment (in this case V, for the Flemish part), the component (in this case A, for the subcorpus containing spontaneous conversations), the fragment number (400392), and the sentence number (6).[7]

4.1.2 Linguistic annotations

The CGN Treebank contains pos tags (Van Eynde 2004) as well as syntactic annotations (Hoekstra et al. 2003). The resulting syntactic structures can be represented as tree structures, cf. Figure 1.

[6] `http://www.let.rug.nl/vannoord/Lassy/alpino_ds.dtd`

[7] In the official release, it is not encoded in the identifier whether the sentence occurs in the Dutch or the Flemish data. This information was added afterwards (based on the information in the corpus).

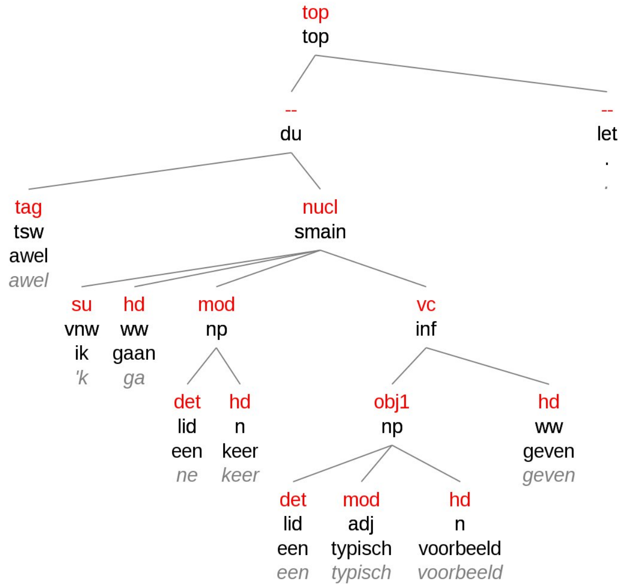


Figure 1: Tree representation of a CGN sentence (fva400392_6)

The annotations of the CGN treebank are manually corrected, which makes the treebank a high-quality resource for linguistic research. The annotations on sentence level have an accuracy of 97.53% (Fersøe et al. 2006).

4.2 LASSY

The LASSY treebank (Large Scale Syntactic Annotation of written Dutch) (van Noord et al. 2013) is a corpus of syntactically annotated sentences.[8] The project resulted in the construction of two treebanks: LASSY Small and LASSY Large. For the purpose of this research LASSY Small is used, since it is complementary to the CGN treebank.

4.2.1 Contents

LASSY Small is a one million word corpus of written Dutch. Table 3 provides an overview of the contents of the LASSY Small treebank.
The word and sentence counts in the table are based on version 1.1 of the LASSY Small treebank.

Each sentence in the corpus has a unique ID, e.g. [dpc-bal-001239-nl-sen.p.15.s.2] for the sentence in (18).

(18) Laat ik een voorbeeld geven.
let I an example give
'Let me give an example.' [LASSY, dpc-bal-001239-nl-sen.p.15.s.2]

The sentence ID refers to the subcorpus (in this case *DPC-bal*), the text number (001239), and the location within the text (page 15 sentence 2). The division into subcorpora allows to investigate stylistic variation (e.g. by comparing newspaper articles to law texts).

[8] http://www.let.rug.nl/~vannoord/Lassy
[9] Paulussen et al. (2006), http://www.kuleuven-kulak.be/DPC

Treebank	Contents	# Sentences	# Words
DPC	Dutch side of the Dutch Parallel Corpus [dpc][9]	11,716	193,029
Wikipedia	Dutch Wikipedia pages [wiki]	7,341	83,360
WR-P-E	E-magazines [WR-P-E-C], news letters [WR-P-E-E], Teletext pages [WR-P-E-H], Web sites [WR-P-E-I], Wikipedia pages [WR-P-E-J]	14,420	232,631
WR-P-P	Books [WR-P-P-B], brochures [WR-P-P-C], guides and manuals [WR-P-P-E], law texts [WR-P-P-F], newspapers [WR-P-P-G], periodicals and magazines [WR-P-P-H], policy documents [WR-P-P-I], proceedings [WR-P-P-J], reports [WR-P-P-K], surveys [WR-P-P-L]	17,691	281,424
WS-U	auto cues [WS-U-E-A], news scripts [WS-U-T-A], texts for the visually impaired [WS-U-T-B]	14,032	184,611
LASSY Small	**Complete treebank**	**65,200**	**975,055**

Table 3: Contents of LASSY Small

4.2.2 Linguistic annotations

LASSY Small is manually corrected after automatic parsing with the Alpino parser (van Noord 2006),[10] a dependency parser for Dutch. The general lay-out of the treebank is very similar to the CGN treebank, as it contains the same POS tags, and almost the same syntactic annotations (van Noord et al. 2011). The main annotation difference is the use of indexed nodes, as illustrated in Figure 2. Since *ik* 'I' is both the subject of *laten* 'let' and the embedded verb *geven* 'give', it is also included as the subject of the verbal complement (VC) in the form of an *index node*.

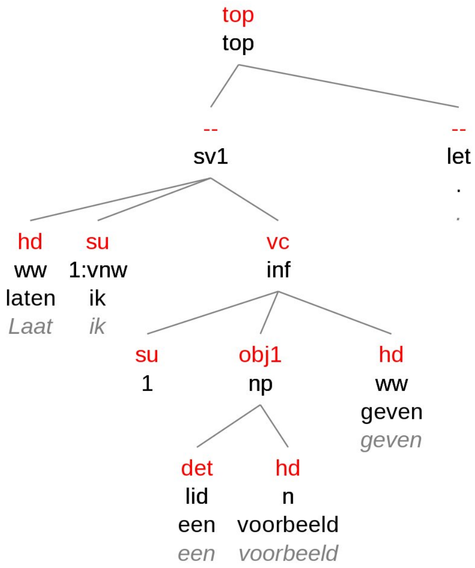


Figure 2: Tree representation of a LASSY sentence (dpc-bal-001239-nl-sen.p.15.s.2)

[10] http://www.let.rug.nl/vannoord/alp/Alpino

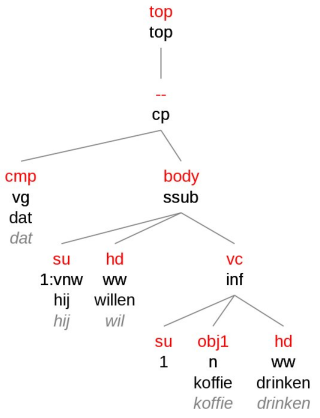
Because of the corrections, LASSY Small is a high-quality resource: The annotations on sentence level have an accuracy of 97.8%; the accuracy of the syntactic annotations on node level is 99.8% (Jongejan et al. 2011).

5. Querying the treebanks using GrETEL

Both the CGN Treebank and LASSY can be queried with XPath, a W3C standard query language for XML trees.[11] This can be done using the GrETEL search engine.[12] In this search tool the user has two ways of entering a syntactic query. The first approach is called Example-based Querying (Augustinus et al. 2012, Augustinus et al. 2013) which consists of a query procedure in several steps, starting from a natural language example and resulting in an automatically generated XPath query, which is then used to query the treebanks. The matching sentences are returned to the user, who can inspect them in more detail. The second approach consists of directly formulating an XPath query that describes the syntactic pattern the user is looking for, which is then processed in the same way as in the first approach.

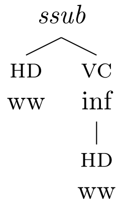
For the research presented here, we started off from XPath queries generated using the example-based method, which were then manually refined by adding more constraints in order to look for more specific constructions. For example, the input construction in (19) was used to automatically derive the query in (20a).[13] (20b) is a visual representation of the query, i.e. a subtree of the parsed example in (19).

(19) ... dat hij koffie wil drinken.
... that he coffee want drink
'... that he wants to drink coffee.'



(20) a.
```
//node[@cat="ssub" and
      node[@rel="hd" and @pt="ww"] and
      node[@rel="vc" and @cat="inf" and
         node[@rel="hd" and @pt="ww"] ] ]
```

b.



[11] http://www.w3.org/TR/xpath

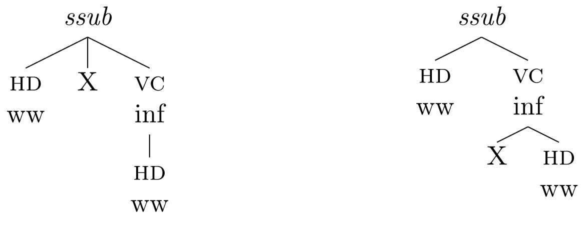
[12] **Gr**eedy **E**xtraction of **T**rees for **E**mpirical **L**inguistics, http://nederbooms.ccl.kuleuven.be/eng/gretel

[13] In order to derive the XPath query, we indicated pos for the verbs in the example sentence in the GrETEL engine.

The query in (20a) extracts V-final constructions (ssub) with a verb (ww) as head daughter (HD) and a verbal complement (VC) in the form of a bare infinitive (inf). The XPath engine does not take into account the order of the nodes; for the query in (20a) it also returns constructions in which the verb follows the infinitive.

Note that the XPath engine performs a *greedy* search,[14] i.e. queries like (20a) do not only return constructions where a finite verb and a bare infinitive cluster in the second pole, but also the constructions where another element intervenes between the finite verb and the second pole. So constructions like the ones in (21) are included as well.[15]

(21)



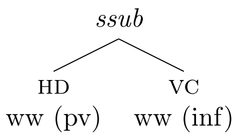
The XPath expressions can be further specified or generalized by adding or removing constraints. For example, by adding the constraint @wvorm="pv" (for 'persoonsvorm') to the node of the selecting verb, we state that the selecting verb should be a finite form. Greedy search furthermore means that the query in (20b) returns all matches containing *at least* a verb and a bare infinitive, so it will also return constructions with more than two verb forms. In order to keep control of the cluster length, we added a constraint stating that the vc node should have no more than one verbal daughter, using the not()-function. The resulting query is shown in (22).

(22)
```
//node[@cat="ssub" and
      node[@rel="hd" and @pt="ww" and @wvorm="pv"] and
      node[@rel="vc" and @cat="inf" and
          node[@rel="hd" and @pt="ww"] and
          not(node[@rel="vc" and (@cat="inf" or @cat="ti" or @cat="ppart" or
          @pt="ww")]) ] ]
```

The queries above look for non-terminal VC nodes, i.e. the VC nodes containing more than one daughter, e.g. a verb and a direct object, such as *koffie drinken* 'drink coffee' in (19). If the VC node consists of one word, it is represented as a terminal node in the treebank. To retrieve the constructions with terminal VCs, the query in (23a) is used.[16] The query tree is presented in (23b).

(23) a.
```
//node[@cat="ssub" and
      node[@rel="hd" and @pt="ww" and @wvorm="pv"] and
      node[@rel="vc" and @pt="ww" and @wvorm="inf"] ]
```

b.



The queries in (22) and (23a) were used to extract constructions of the type 'Vfinal, finite + infinitive', i.e. category f in Table 4 (see infra). The other clustering constructions are found by means of adaptations and extensions of the queries presented in this section. Constructions with

[14] The notion *greedy* is used in a similar way as pattern matching with regular expressions, see a.o. Jurafsky and Martin (2009, p.56); XPath expressions are greedy in the sense that they match with as much of a tree pattern as they can.

[15] 'X' stands for any sequence of nodes that may occur in that position.

[16] The not() condition need not be stated here, since the terminal VC node cannot have any embedded VCs.

more than two verb forms in the cluster have another vc node embedded under the vc. V-initial constructions can be retrieved by changing the label `ssub` to `smain` for V-second clauses or to `sv1` for V-first clauses. The label `ppart` is used for non-terminal past participles (i.e. participial phrases), whereas the Dutch label `vd` (for 'voltooid deelwoord') is used for terminal nodes.

For example, the query in (24) returns V-initial constructions with a finite verb, a bare infinitive and a past participle, i.e. category d in Table 4 (see infra).[17]

(24)
```
//node[(@cat="smain" or @cat="sv1") and
        node[@rel="hd" and @pt="ww" and @wvorm="pv"] and
        node[@rel="vc" and @cat="inf" and
            node[@rel="hd" and @pt="ww"] and
            node[@rel="vc" and @cat="ppart" and
                node[@rel="hd" and @pt="ww"] ] ] ]
```

6. Results

6.1 Identifying the clusters

Even though the treebank annotations do not contain a separate tag for clustering verbs, it is possible to automatically extract clustering constructions using the relevant queries (see section 5). Table 4 presents the treebank counts for the constructions with at least two verb forms in the cluster. For each construction, the total number of occurrences is the sum of the queries for non-terminal vcs and terminal vcs.

As motivated in section 1.2.2, we want to separate the constructions that potentially contain cluster creepers from the constructions that do not. Since cluster creeping is excluded in constructions in which the main verb occurs at the beginning of the cluster, the results were split up into two categories: Clusters in which the main verb is not the first verb in the cluster (mv $\neq$ 1), and clusters in which it is (mv = 1).

In section 3 it was already mentioned that constructions with *te* infinitives are not necessarily clustering. Those constructions can be split up into constructions where the *te*-infinitive is a part of the cluster, as in (25), and constructions in which it is not, as in (26) and (27).

(25) 'k ben blij dat ik zo veel belangstelling *heb weten te wekken.*
I am happy that I so much interest have known to raise

'I am glad that I have been able to raise so much interest.' [CGN, fnf007126_142]

(26) en ik denk dat men daarin *moet trachten* het juiste evenwicht te zoeken.
and I think that one there-in has try the right balance to search

'and I think that one has to try to find the right balance in that.' [CGN, fvg600012_38]

(27) Nu pas kunnen de bedrijven *proberen* wat terúg te verdienen.
now only can the companies try something back to gain

'Only now the companies can try to gain something back.'[LASSY, WS-U-E-A-0000000042.p.31.s.8]

In constructions with ipp, such as (25), the *te*-infinitive is part of the verb cluster. In (26) the cluster consists of a finite verb and a bare infinitive, whereas the *te*-infinitive is in the *naveld.* (26) has thus the same type of cluster as the constructions in category f. (27) does not contain a verb cluster: the verb *proberen* 'try' is the only verb in the second pole, whereas the *te*-infinitive is in the *naveld.* Since we are interested in constructions with at least two verbs in the second pole, such constructions were removed from the data set.

[17] Also in this case the `not()` function need not be stated. As the past participle is the last element of the cluster, it does not matter whether it has any embedded (extraposed) vc nodes.

Since constructions with a *te*-infinitive in the *naveld* are tagged similarly to clustering constructions (i.e. both constructions received a `vc` tag in the treebanks), we have limited the set of clustering constructions containing a *te*-infinitive to the set of IPP constructions (containing at most one *te*-infinitive), as those constructions are always clustering.

	CGN			LASSY		
Cluster Type	MV $\neq 1$	MV $= 1$	Sum (#)	MV $\neq 1$	MV $= 1$	Sum (#)
a) Vfinal, finite + past part	1664	1626	3290	3544	1519	5063
b) Vfinal, finite + inf + past part	152	138	290	443	262	705
c) Vfinal, finite + inf + inf + past part	11	11	22	20	10	30
d) Vinitial, inf + past part	127	356	483	830	532	1362
e) Vinitial, inf + inf + past part	10	18	28	37	29	66
f) Vfinal, finite + inf	3472	43	3515	2989	6	2995
g) Vfinal, finite + inf + inf	438	0	438	298	0	298
h) Vfinal, finite + inf + inf + inf	14	0	14	5	0	5
i) Vinitial, inf + inf	1715	1	1716	653	0	653
j) Vinitial, inf + inf + inf	49	0	49	9	0	9
k) Vfinal, finite + inf (IPP) + *te* inf	3	0	3	10	0	10
l) Vinitial, inf (IPP) + *te* inf	14	0	14	19	0	19
Sum (#)	**7669**	**2193**	**9862**	**8857**	**2358**	**11215**
Sum (%)	**77.76**	**22.24**	**100**	**78.97**	**21.03**	**100**

Table 4: Clustering constructions in CGN and LASSY

We have found 9862 clustering constructions in CGN and 11215 in LASSY. Neither of the treebanks contains clusters with more than four verbs. In LASSY, the majority of the clustering constructions contains a past participle (categories a-e), whereas in CGN, the clusters containing bare infinitives occur more frequently (categories f-j).

The results show that the proportion of clusters that potentially contain cluster creepers, i.e. the clusters in which the main verb is not the first verb in the cluster (MV $\neq 1$), is more or less equal in both treebanks, i.e. 77.76% in CGN and 78.97% in LASSY.

6.2 Cluster creepers

After having collected the set of clustering constructions, we extracted the constructions with cluster creepers, i.e. constructions in which nonverbal elements occur between the verbs in the second pole.[18] Due to the treebank design, it is not possible to extract all constructions with cluster creepers in that way, however. Separable verb particles (SVPs) are only tagged separately if they are written as a separate word, but not if they are written as a part of the verb, as in example (8b). In the LASSY treebank they can be extracted in another way, but not in CGN, as will be explained in section 6.3. Therefore, this section focuses on the cluster creepers that are written as a separate word.

Since the set of constructions with cluster creepers is low in comparison to the set of all clustering constructions, the results were manually verified after the automatic extraction.

Even though the quality of the annotations in both LASSY and CGN is very high, the treebanks contain some annotation errors that are problematic for this research. For example, sentences that are erroneously tagged as V-final whereas they are V-initial.

In (28), for instance, the clause after 'uh' is tagged `ssub` instead of `smain`.

[18] For the extraction of cluster creepers, we started from the XPath queries defined in section 5. Since it is hard to determine the linear order of the nodes in an elegant way using XPath, we have used XQuery scripts in which we defined constraints for extracting the constructions in which nonverbal elements occur between the verbs. As an example, the XQuery script which was used to find cluster creepers in two-verb clusters with a finite verb and an infinitive is included as an appendix to this paper.

(28) dan kan ik uh ik kan ’m in de keuken nergens inpluggen vrienden.
then can I uh I can him in the kitchen nowhere plug in friends
‘then I can’t plug it in in the kitchen, friends.’ [CGN, fna000573_58]

Besides the elimination of annotation errors, two types of false positives were filtered out semi-manually. The first type concerns constructions with stopgaps, corrections, and/or interruptions, such as the examples in (29). Those constructions were mainly encountered in CGN.

(29) a. maar wat wij merkten in Frankrijk was dikwijls dat ge ’s middags soms
but what we noticed in France was often that you at lunchtime sometimes
zeer goede menu’s kondt gebr- allee eten dus hè.
very good menu’s could use.INTERRUPTED go.STOPGAP eat thus hè
‘What we often noticed in France was that you sometimes could use- well eat very good menu’s at lunchtime.’ [CGN, fva400295_400]

b. enfin ik weet niet hoe ik het moet uh omschrijven uh.
well I know not how I it must uh describe uh
‘well I don’t know how I have to uh describe it.’ [CGN, fva400534_85]

The second type of false positives is the occurrence of punctuation marks within the verb cluster. Those examples were exclusively found in LASSY.

(30) Het is dus niet zo dat deze tanks al eerder “gekannibaliseerd” waren om er
it is thus not so that this tanks already before “cannibalised” were for there
bruikbare onderdelen uit te halen.
usable parts out to get
‘It is thus not the case that these tanks were “cannibalised” before to get useful parts out of it.’ [LASSY, WR-P-E-I-0000013937.p.4.s.235]

Table 5 presents the results for both treebanks. To compare the amount of cluster creepers to the set of clustering constructions that may allow cluster creepers, i.e. the constructions in which the main verb is not the first verb of the clusters, the numbers for those constructions are included in this table as well (MV $\neq 1$).

CLUSTER TYPE	CGN	LASSY	SUM
a) Vfinal, finite + past part	23	11	34
b) Vfinal, finite + inf + past part	2	0	2
c) Vfinal, finite + inf + inf + past part	1	0	1
d) Vinitial, inf + past part	1	1	2
e) Vinitial, inf + inf + past part	0	0	0
f) Vfinal, finite + inf	79	7	86
g) Vfinal, finite + inf + inf	20	0	20
h) Vfinal, finite + inf + inf + inf	1	0	1
i) Vinitial, inf + inf	49	0	49
j) Vinitial, inf + inf + inf	4	0	4
k) Vfinal, finite + inf (IPP) + *te* inf	0	2	2
l) Vinitial, inf (IPP) + *te* inf	3	3	6
SUM	**183**	**24**	**207**
MV $\neq$ 1	**7669**	**8857**	**16526**

Table 5: Frequency of cluster creepers in CGN and LASSY

Compared to the large amount of clustering constructions, the results in Table 5 show that cluster creeping is a very infrequent phenomenon in both CGN and LASSY. In CGN, we have encountered 183 constructions with cluster creepers, whereas in LASSY we have only found 24. So, cluster creeping occurs more frequently in the spoken data (CGN) than in the written data (LASSY). The constructions account for 2.4% of all clusters that potentially allow cluster creepers (mv $\neq 1$) in CGN, and for less than 0.3% of those constructions in LASSY.

6.2.1 Single cluster creepers

Despite the low number of corpus examples, the constructions with cluster creepers show a large variety of cluster creepers, both in category and syntactic function. The three types mentioned in Haeseryn et al. (1997) are all present in the data: The sentences in (31) show cluster creeping by a predicative adjective (31a) and by a part of a fixed expression (31b). (32) is an example of adposition stranding within the cluster. In (33a) the cluster is interrupted by an object, and in (33b) by an adverbial modifier.

(31) a. de dokters zeggen wel dat ‘t gaat *goed* komen.
the doctors say that it goes good come
‘The doctors say that it will be fine.’ [CGN, fva400370_6]

b. ’k zeg dat gaat moeten beginnen *op gang* komen hè.
I say that goes must begin on pace come
‘I say that should start to get going.’ [CGN, fva400643_87]

(32) De plicht die hem nu roept, kan hem straks de mooiste baan kosten waar een
the duty than him now calls can him later the most-beautiful job cost where a
Beier kan *van* dromen.
Bavarian can of dream
‘The duty that calls him now can cost him the most beautiful job a Bavarian can dream of.’ [LASSY, WR-P-P-I-0000000033.p.21.s.4]

(33) a. als ze moeten *teksten* schrijven dan schrijven ze die met de PC.
if they must texts write then write they them with the PC
‘If they have to write texts then they write them with a PC.’ [CGN, fvb400165_130]

b. maar normaal moet ge dat kunnen *zo* regelen dus dat dat wegblijft dus dat
but normally must you that can so arrange thus that that away-stays thus that
‘t niet verschijnt.
it not appears
‘But normally you have to arrange that in such a way that it stays away so that it does not appear.’ [CGN, fva400079_264]

An overview of all creeper types is provided in Table 6. The labels in the columns indicate the syntactic function (dependency relation): Separable verb particle (svp), prepositional complement (pc), direct object (obj1), predicative complement (predc), location or direction complement (ld), indirect object (obj2), modifier (mod), and predicative modifier (predm). The left part of the table concerns complements selected by the verb, whereas the right part concerns modifiers.

The labels in the rows indicate the lexical categories (pos) at the top half of the table and the phrasal categories at the bottom part of the table. 14 instances of cluster creeping show a combination of several categories. They are not included in Table 6, but will be discussed in this section as well (see 6.2.2).

	SVP	PC	OBJ1	PREDC	LD	OBJ2	MOD	PREDM	SUM (#)	SUM (%)
PREP	12	37	0	2	12	0	7	0	**70**	**36.27**
ADJ	13	0	0	20	0	0	11	0	**44**	**22.80**
N	5	0	16	0	0	0	0	0	**21**	**10.88**
ADV	5	0	0	0	2	0	6	1	**14**	**7.25**
PRON	0	0	4	1	1	0	5	0	**11**	**5.70**
PP	4	1	0	2	7	1	5	0	**20**	**10.36**
NP	0	0	8	0	0	0	1	0	**9**	**4.66**
AP	0	0	0	1	0	0	2	0	**3**	**1.55**
ADVP	0	0	0	0	0	0	1	0	**1**	**0.52**
SUM (#)	**39**	**38**	**28**	**26**	**22**	**1**	**38**	**1**	**193**	
SUM (%)	**20.21**	**19.69**	**14.51**	**13.47**	**11.40**	**0.52**	**19.69**	**0.52**		**100**

Table 6: Types of cluster creepers in CGN and LASSY

As expected, the largest category consists of cluster creepers where an SVP occurs within the cluster, as in (34).

(34) Ik heb mijn agenda niet hoeven *om* te gooien om die man te kunnen ontvangen (...)
I have my diary not need down to throw to that man to can receive
I did not have to completely change my schedule to be able to receive that man (...)' [LASSY, dpc-rou-000479-nl-sen.p.10.s.14]

As mentioned above, the results do not include the cases of cluster creeping with separable verbs in which the SVP and the verb are written as one word.

Another major group are the prepositional complements. They include the cases of adposition stranding illustrated in (32).

The other frequently occurring creeper types are also mentioned in Haeseryn et al. (1997), i.e. predicative adjectives (31a), direct objects (33a), and modifiers (33b).

More remarkable examples in the data set are the constructions in which a full phrase occurs within the cluster, such as the prepositional indirect object in (35a) and the location complement in (35b).

(35) a. (...) 'k weet 'k ik niet of dat 'k ik dat nu moet laten weten aan hem of dat 'k ik
(... I know I I not or that I I that now must let know to him or that I I
dat eerst moet *aan mijn kot* vragen (...)
that first must to my student's apartment ask (...))
'(...) I don't know whether I should let him know or that I should ask (the people of) my student's apartment first.' [CGN, fva400507_4]

b. dat die nu moet *in de Verenigde Staten* blijven in Miami bij de familie (...)
that that now must in the United States stay in Miami with the family (...)
'that he now has to stay in the United States in Miami with his family (...)' [CGN, fvj600261_9]

A final note on Table 6 concerns the four instances of phrasal SVPs. Those constructions all contain fixed expressions, such as the example given in (31b). Nonverbal parts of fixed expressions are tagged as SVPs in the treebanks, but one could also classify those constructions as PCs.

6.2.2 Multiple cluster creepers

The 14 constructions that are not included in Table 6 form a heterogenous group that is not encountered in the literature on cluster creeping. Those examples contain multiple cluster creepers.

It is hard to draw any generalizations over this kind of constructions. Out of the 14 instances, 10 cluster creepers consist of a modifier, combined with a direct object, a predicative complement, a prepositional complement or a locational/directional complement. With regard to the syntactic category of the complex cluster creepers, any combination of lexical and phrasal categories seems to be possible. Some examples are given in (36).

(36) a. (...) den dokter heeft eerst moeten *tien minuten die twee vrouwen* kalmeren voor
(...) the doctor has first must ten minutes those two women calm-down before
ie het onderzoek kon doen.
he the examination could do

'(...) The doctor first had to calm down those two women for ten minutes before he could do the examination.' [CGN, fvn400019_191]

b. (...) alhoewel dat ik er wel 'ns zou *graag aan* meedoen.
(...) although that I there indeed some time would gladly on participate

'(...) although I would like to participate in that.' [CGN, fvb400165_191]

c. (...) als je zeg maar homo bent en dan uh ja gewoon nie ja je weet niet
(...) if you say but gay are and then uh yeah just not yeah you know not
hoe je het met je ouders moet *'t erover* hebben (...)
how you it with your parents must it there-over have (...)

'(...) for example if you are gay and you don't know how you should talk about it with your parents.' [CGN, fna000541_298]

In (36a) the cluster contains a temporal modifier and a direct object NP. (36b) is a combination of an adverbial modifier and adposition stranding. In (36c) not only the preposition occurs within the cluster, but the PC as a whole is realised in situ. Moreover, the cluster is interrupted by the direct object as well. Not surprisingly, all instances of such *complex* creeping constructions occur in the spoken data (CGN).

6.2.3 Position of the cluster creepers

Another aspect regarding cluster creeping is the position of the nonverbal elements. In section 1.2.2 it was said that in clusters with more than two verbs, the nonverbal element typically occurs right in front of the main verb.

In the data set, there are 30 cases of cluster creeping in constructions with three or four verb forms. In 18 cases, the cluster creeper occurs just in front of the main verb, as in (37a), whereas in 12 constructions, they occupy a more leftward position in the cluster, as in (37b). The numbers confirm the statement of Haeseryn et al. (1997), but the amount of relevant examples in the treebanks is very low.

(37) a. (...) iemand die zich heeft weten *binnen* te werken in kringen met een hoog
(...) someone who himself has know in to work in circles with a high
sociaal aanzien (...)
social standing (...)

'(...) someone who has managed to work his way up into high society (...)' [LASSY, dpc-ind-001652-nl-sen.p.11.s.1]

b. dus dat huisje wat we daar hebben *neer* laten zetten (...)
so that house.DIM what we there have down let put (...)

'so that little house that we got built over there.' [CGN, fni007330_43]

6.2.4 Language-internal variation

Haeseryn et al. (1997) state that cluster creeping is more typical in Belgian Dutch compared to Netherlandic Dutch. Since CGN contains meta-information on the origin of the data, it is possible to verify that aspect in the treebank results as well. Out of the 183 occurrences of cluster creeping in CGN, 145 constructions are part of the Belgian data set, while the remaining 38 constructions occur in the Netherlandic data, so the data indeed show that cluster creeping is more common in Belgian Dutch. In section 4 it was mentioned that CGN contains twice as much Netherlandic data as Belgian data. If we normalise the data, it turns out that cluster creeping occurs 7.6 times more often in the Belgian data compared to the Netherlandic part of the corpus.

6.3 A note on separable verbs in LASSY

As mentioned in section 6.2, separable verbs may be written as one word if the SVP occurs next to the verb. In those cases the SVPs are not individually tagged in the treebanks.

It is possible, however, to detect the clusters containing an SVP by extracting the root forms of the verbs in the clustering constructions in LASSY. In the `root` tag of the verb the root and the SVP are separated by an underscore, e.g. bel_op for the verb *opbellen* 'call'. The numbers are given in Table 7.[19]

	MV $\neq 1$	MV $= 1$	SUM
Separable verbs	**2556**	390	**2946**
Non-separable verbs	6301	1968	8269
SUM	8857	2358	**11215**

Table 7: Distribution of separable verbs within clusters in LASSY

The results show that there are 2556 occurrences of cluster creeping by an SVP in the LASSY treebank, indicating that such constructions are relatively frequent, in contrast to the observations in Table 6. The creeping constructions account for 22.8% of all clustering constructions in LASSY, and for 86.8% of all separable verbs in clusters in LASSY.

Note that separable verbs are only represented as such in the root forms but not in the lemmas. It is possible to retrieve SVPs in this way in the LASSY treebank, but not in CGN, since the CGN treebank only includes lemmas but no root forms. It is thus not possible to compare the results in Table 7 to the frequency of separable verbs in CGN. Exploring alternative ways of retrieving those constructions in CGN remains future work.

7. Conclusions and future work

This paper investigated the occurrence of cluster creepers in the CGN and LASSY treebanks. Since those treebanks do not contain a specific tag for clustering verbs, we first had to define which constructions we consider as verb clusters before extracting the relevant constructions. Compared to the large amount of clustering constructions, the treebanks show that cluster creeping is a low-frequent phenomenon in Dutch, except in the case of SVPs. Despite the small set of treebank results, the variety of the creeper types turned out to be rather large. All categories mentioned in (Haeseryn et al. 1997) are included in the data. Moreover, a subset of the cluster creepers consists of a combination of several creeper types. Those constructions are not mentioned in the literature on the phenomenon, showing that corpus-based research can add additional insights into linguistic phenomena.

Further work is needed on how to deal with the inconsistent spelling in Dutch regarding separable verb particles, as well as with the problematic annotation of separable verbs in the treebanks.

[19] The results include the examples with the separately tagged SVPs as well.

As we only found some examples of cluster creeping in the data, it would be interesting to investigate the phenomenon in a larger corpus, for example the SoNaR treebank (Oostdijk et al. 2013). That treebank of written Dutch not only contains more data (500M words), it also covers a larger variety of text types.

Acknowledgments

We thank the audience of the CLIN conference (Leiden, January 17, 2014) and the anonymous reviewers for their comments. The research presented in this paper is part of a project on complement raising and cluster formation in Dutch, sponsored by FWO Vlaanderen (2011-2015, G.0.559.11.N.10).

References

Augustinus, L. and F. Van Eynde (2012), A Treebank-based Investigation of IPP-triggering Verbs in Dutch, *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT11)*, Edições Colibri, Lisbon.

Augustinus, L., V. Vandeghinste, and F. Van Eynde (2012), Example-Based Treebank Querying, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*, European Language Resources Association (ELRA), Istanbul, pp. 3161–3167.

Augustinus, L., V. Vandeghinste, I. Schuurman, and F. Van Eynde (2013), Example-Based Treebank Querying with GrETEL - now also for Spoken Dutch, *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, NEALT Proceedings Series 16, Oslo, pp. 423–428.

Evers, A. (2003), Verbal Clusters and Cluster Creepers, *in* Seuren, P.A.M. and G. Kempen, editors, *Verb Constructions in German and Dutch*, John Benjamins, Amsterdam/Philadelphia, pp. 43–89.

Fersøe, H., S. Olsen, C. Navarretta, and B. Jongejan (2006), Validation Report Corpus Gesproken Nederlands 1.0 Linguistic Validation, *Technical report*, Center for Sprogteknologi, University of Copenhagen.

Haegeman, L. and H. van Riemsdijk (1986), Verb Projection Raising. Scope and the Typology of Rules Affecting Verbs., *Linguistic Inquiry* **17**, pp. 417–466.

Haeseryn, W., K. Romijn, G. Geerts, J. de Rooij, and M. van den Toorn (1997), *Algemene Nederlandse Spraakkunst*, second ed., Martinus Nijhoff/Wolters Plantyn, Groningen/Deurne.

Hoekstra, E. (2010), On the interruption of Verb-Raising clusters by nonverbal material, *Structure Preserved. Studies in Syntax for Jan Koster*, John Benjamins, Amsterdam, pp. 175–183.

Hoekstra, H., M. Moortgat, B. Renmans, M. Schouppe, I. Schuurman, and T. van der Wouden (2003), *CGN Syntactische Annotatie.* 77p.

Jongejan, B., S. Olsen, and H. Fersøe (2011), Validation Report Lassy Corpora Linguistic Validation, *Technical report*, Center for Sprogteknologi, University of Copenhagen.

Jurafsky, D. and J. Martin (2009), *Speech and Language Processing*, 2nd ed., Pearson Education, New Jersey.

Oostdijk, N., M. Reynaert, V. Hoste, and I. Schuurman (2013), The construction of a 500-million-word reference corpus of contemporary written Dutch, *in* Spyns, P. and J. Odijk, editors, *Essential Speech and Language Technology for Dutch: resources, tools and applications*, Springer, pp. 219–247.

Oostdijk, N., W. Goedertier, F. Van Eynde, L. Boves, J.-P. Martens, M. Moortgat, and H. Baayen (2002), Experiences from the Spoken Dutch Corpus Project, *in* Rodriguez, Manuel Gonzalez and Carmen Paz Saurez Araujo, editors, *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, pp. 340–347.

Paulussen, H., L. Macken, J. Truskina, P. Desmet, and W. Vandeweghe (2006), Dutch Parallel Corpus: a multifunctional and multilingual corpus, *Cahiers de l'Institut de Linguistique de Louvain, CILL* **32** (1-4), pp. 269–285.

Van Eynde, F. (2004), Part of Speech Tagging en Lemmatisering van het Corpus Gesproken Nederlands, 87p.

van Noord, G. (2006), At Last Parsing Is Now Operational, *in* Mertens, P., C. Fairon, A. Dister, and P. Watrin, editors, *TALN 2006. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pp. 20–42.

van Noord, G., G. Bouma, F. Van Eynde, D. de Kok, J. van der Linde, I. Schuurman, E. Tjong Kim Sang, and V. Vandeghinste (2013), Large Scale Syntactic Annotation of Written Dutch: Lassy, *in* Spyns, P. and J. Odijk, editors, *Essential Speech and Language Technology for Dutch: resources, tools and applications*, Springer.

van Noord, G., I. Schuurman, and G. Bouma (2011), *Lassy Syntactische Annotatie, Revision 19455.* 208p.

Wurmbrand, S. (2005), Verb Clusters, Verb Raising, and Restructuring, *in* Everaert, M. and H. van Riemsdijk, editors, *The Blackwell Companion to Syntax*, Vol. V, Blackwell, Oxford, chapter 75, pp. 229–343.

Appendix: XQuery script for cluster creepers

This XQuery script looks for cluster creepers in V-final finite-infinitive clusters:[20]

```
(: XPath extracts  V-final finite-infinitive clusters in the LASSY small treebank :)
for $xp in db:open("LASSY_ID")/treebank/alpino_ds
//node[@cat="ssub" and node[@rel="hd" and @pt="ww" and @wvorm="pv"] and
node[@rel="vc" and @cat="inf" and node[@rel="hd" and @pt="ww"] and
not(node[@rel="vc" and (@cat="inf" or @cat="ti" or @cat="ppart" or @pt="ww")])]]

(: get sentence ID:)
let $sentenceid := ($xp/ancestor::alpino_ds/@id)

(: get sentence:)
let $sentence := ($xp/ancestor::alpino_ds/sentence)

(: get finite verb and infinitive :)
let $finite := ($xp/ node[@rel="hd" and @pt="ww" and @wvorm="pv"]/@word)
let $infinitive := ($xp/node[@rel="vc" and @cat="inf"]/node[@rel="hd" and @pt="ww"]/@word)

(: get position of the finite verb and the infinitive :)
let $finiteposition := ($xp/ node[@rel="hd" and @pt="ww" and @wvorm="pv"]/@begin)
let $infinitiveposition := ($xp/node[@rel="vc" and @cat="inf"]/node[@rel="hd" and @pt="ww"]/@begin)

(: get cluster creepers :)
(: finite - infinitive :)
let $creepers1 := ($xp/descendant::node[(number(@begin) > number($finiteposition)) and
(number(@begin) < number($infinitiveposition))])
(: infinitive - finite :)
let $creepers2 := ($xp/descendant::node[(number(@begin) < number($finiteposition)) and
(number(@begin) > number($infinitiveposition))])

(: only return constructions with cluster creepers :)
where ($creepers1 or $creepers2)

(: return sentences, verb cluster, cluster creepers (word, syntactic function and POS tag) :)
return
if (number($finiteposition) < number($infinitiveposition))
then <match>{data($sentenceid)}#{data($sentence)}
#FINITE-INFINITIVE#{data($finite)}-{data($infinitive)}
#{data($creepers1/@word)}#{data($creepers1/@rel)}#{data($creepers1/@pt)}</match>

else
<match>{data($sentenceid)}#{data($sentence)}
#INFINITIVE-FINITE#{data($infinitive)}-{data($finite)}
#{data($creepers2/@word)}#{data($creepers2/@rel)}#{data($creepers2/@pt)}</match>
```

[20] Comments are put between (: and :).