Computational Linguistics in the Netherlands Journal 15 (2026) 193-216 Submitted 01/2026; Published 06/2026

Parsing Dutch C-CLAMP
Unlocking 150 years of written Dutch for syntactic analysis

Gerlof Bouma[*] GERLOF.BOUMA@GU.SE
Evie Coussé[§] EVIE.COUSSE@GU.SE
Gertjan van Noord[¶] G.J.M.VAN.NOORD@RUG.NL

[*] *Språkbanken Text / Department of Swedish, Multilingualism, Language Technology, University of Gothenburg, Sweden*
[§] *Department of Languages and Literatures, University of Gothenburg, Sweden*
[¶] *Center for Language and Cognition, University of Groningen, The Netherlands*

Abstract

We present the parsed Gothenburg edition of the Dutch Corpus of Contemporary and Late Modern Periodicals (Dutch C-CLAMP) and the Dutch Verb Construction Database derived from this parsed corpus. Dutch C-CLAMP is a diachronic corpus of Dutch-language periodicals, with material from the 19th and 20th century from Belgium and The Netherlands. Both the parsed corpus and the Dutch Verb Construction Database will be made available to other researchers. In the paper we discuss the creation of the parsed corpus and offer a quantitative overview of the resource. In the second half of the paper we introduce the Dutch Verb Construction Database, a research database to support large scale diachronic investigation of verb constructions. We describe its extraction and present an evaluation of the database against manually annotated data. We end the paper with a small case study on verb order, exemplifying one type of research the database facilitates.

1. Introduction

We present the parsed Gothenburg edition of Dutch C-CLAMP and the Dutch Verb Construction Database derived from this parsed corpus. Both resources were developed within the research project *Verb Constructions in the Recent History of Dutch. A Constructional Network Perspective* (Swedish Research Council, nr. 2023-00873), which addresses system-wide grammatical change in the verb phrase. Dutch historical linguistics has focused on uncovering the history of only a limited set of verb constructions, more specifically of perfect and modal constructions, as illustrated in (1) and (2), marked in bold.

(1) *De politie* ***heeft*** *gisteren een verdachte* ***aangehouden****.*
the police has yesterday a suspect arrested
'The police have arrested a suspect yesterday.'

(2) *Kinderen* ***kunnen*** *hier nog veilig* ***fietsen****.*
Children can here still safely bike
'Children can still bike safely here.'

Several corpus studies have been devoted to uncovering the changing word order of these constructions in the subordinate clause (Coussé 2008, Coupé 2015), and changes in meaning and usage context that accompany the grammaticalization of these constructions (Coussé 2014, Byloo and Nuyts 2014, Nuyts et al. 2022). The history of other verb constructions remains largely underresearched for Dutch. One of aims of our project is to fill this gap by tracing the recent history of the entire set of verb constructions that exists today.

The Dutch Corpus of Contemporary and Late Modern Periodicals (Dutch C-CLAMP) compiled by Piersoul et al. (2021) offers an excellent starting point for this type of large-scale research in

terms of size and design.[1] It is a diachronic corpus of Dutch-language periodicals, with material from the 19th and 20th century, published in Belgium and The Netherlands. With its rich meta-data and linguistic annotations, Dutch C-CLAMP has already been the source of a range of studies into morphosyntactic change in the recent history of Dutch (e.g., De Troij and Van de Velde 2020, Van de Velde et al. 2020, Piersoul and de Velde 2023, Nijs et al. 2025, to name a few). However, the current version of the corpus does not include syntactic parses, which earlier work has shown facilitate large-scale studies of verb constructions in contemporary Dutch (Augustinus 2015, Bloem 2021, Coussé and Bouma 2022). We therefore further enrich the corpus with a layer of automatic parses, which provides detailed syntactic structure information for all sentences.

This fully automatic approach contrasts with the traditional approach in corpus-based diachronic syntax, which consists of constructing highly curated treebanks that are either fully manually annotated or combine automatic parsing with manual correction (see Piotrowski 2012, Taylor 2020 for an overview). Such resources are typically based on the phrase-structure formalism of the Penn Treebank (Marcus et al. 1993) or the dependency framework used in the PROIEL family of treebanks (Eckhoff et al. 2018), and have been developed for a range of historical language stages - yet, not for historical Dutch. The manual and semi-manual annotation workflows of historical treebanks limit scalability, restricting their application to smaller corpora, which make them unsuitable for the present project.

Instead, we use the Alpino parser (van Noord 2006) developed for contemporary Dutch to Dutch C-CLAMP, to add the syntactic annotation layer automatically, guided by the assumption that the language of the corpus, representing written Dutch of the past 150 years, is sufficiently similar to present-day Dutch. Historical materials present a challenge to tools developed for contemporary language due to orthographic variation, language change, and domain differences (Piotrowski 2012). We address some of these issues by applying the methods developed in van Cranenburgh and van Noord (2022), who parse a small corpus of nine nineteenth-century novels using Alpino by adding surface-level pre-annotations to the data to bridge differences in spelling, and to some degree grammar, between the historical texts and a parser trained on contemporary Dutch. Such surface-level transformations are prioritized over enhancing the parser's grammar, as adapting the grammar would require retraining core components such as the disambiguation model and the POS-tagger model, for which there is not enough historical data available. Furthermore, adaptations of the parser are likely to have the negative side-effect of reducing the quality of the parser on contemporary Dutch. The surface-level pre-annotation method has also been applied to a corpus of Dutch novels (van Cranenburgh 2022a, van Cranenburgh 2022b). An early precursor to the approach, incidentally with the shared goal of extracting a linguistic database from the parsed data, can be found in Pettersson et al. (2012).

The rest of the paper is structured as follows. We start by describing the source material of Dutch C-CLAMP (Section 2) and the creation of the parsed Gothenburg edition, which involved an iterative process of making adjustments to the corpus pre-processing step and to some extent to the parser to achieve the best parsing quality possible (Section 3). While these steps are primarily aimed at improving parsing quality, some also have the potential to uncover patterns of language change within the corpus. We therefore explore, as a short excursion, how meta-annotations and parsers coverage provide a first indication of changing spelling and syntactic complexity within the corpus (Section 4). The parsing of Dutch C-CLAMP is not the ultimate goal of our project, but rather a necessary step for the automatic extraction of verb constructions from the corpus. Section 5 outlines the creation of the Dutch Verb Construction Database, which consists of all verb constructions extracted from the parsed corpus, together with their context, relevant morphosyntactic information (such as word order, verb clustering), and meta-information. We also provide an evaluation of the database, and by extension, an extrinsic evaluation of the parsed corpus on which it is based. Section 6 presents a short case study based on the database to illustrate its analytical potential.

1. Dutch C-CLAMP is available through the Dutch Language Institute, `http://hdl.handle.net/10032/tm-a3-d3`.

Country	Periodical	First vol	Latest vol
Belgium	Dietsche Warande en Belfort	1900	1999
	Streven	1933	1947
	Nieuw Vlaams Tijdschrift	1946	1983
	Spiegel der Letteren	1957	1985
The Netherlands	De Gids	1837	1999
	Jaarboek Maatschappij der Nederlandse Letterkunde	1901	1999
	Groot Nederland	1903	1943
	De Nieuwe Taalgids	1907	1995
	De Gemeenschap	1925	1941
	Forum	1932	1935
	Ontmoeting	1946	1963
	De Nieuwe stem	1946	1967
	Forum der Letteren	1960	1995

Table 1: Periodicals per country

2. Source material

The text sources for Dutch C-CLAMP are linguistic and literary periodicals from Belgium and The Netherlands, published between 1837 and 1999. This selection is described and motivated in Piersoul et al. (2021). The digitized texts were all obtained from the digital library of Duch literature DBNL at the National Library of the Netherlands.[2]

The 19th century material comes exclusively from the literary periodical *De Gids*, published in the Netherlands. The longest-running literary periodical in the Dutch language area, De Gids was founded in 1837 and is still published today. Dutch C-CLAMP contains all volumes up to and including 1999. For the 20th century, Dutch C-CLAMP also contains twelve other periodicals from both Belgium and The Netherlands. Alltogether, the materal comprises almost 53 thousand articles, taken from 610 volumes of 13 periodicals. The periodicals and time spans are given in Table 1.

The corpus as described in Piersoul et al. (2021) and distributed through the Dutch Language Institute comes with several layers of processing, including sentence segmentation and tokenization, language filtering and spelling normalization, lemmatization and part-of-speech tagging. A basic requirement when trying to maximize parsing quality is full control over the early text processing steps, such as sentence segmentation and tokenization. Errors and choices that deviate from the parser's expectations at this stage are likely to lead to problems for the parser. The distributed corpus does not come with unsegmented sources, which means that fixing, say, an issue with the segmentation involves undoing segmentation. However, to do so accurately we need information about the structure of the text like paragraph breaks, which is not available to us in the corpus's segmented form. We have therefore chosen not to use the existing processed materials at all in the creation of the Gothenburg Dutch C-CLAMP, but to start from scratch with the selection taken from DBNL. All article texts were re-extracted from the TEI Lite files available at DBNL. As the texts contain material from several other languages, we language-filtered using the OpenLID fastText model (Burchell et al. 2023) at paragraph level. In this step, just over 2% of the data in terms alphanumerical tokens was removed.[3] Our method for spelling normalization will be discussed in

2. Some of the materials are still under copyright. Using the corpus therefore requires an agreement with the National Library.

3. In particular, we kept all paragraphs labelled by OpenLID as Dutch (94% of paragraphs), Limburgian (2.5%) or Northern Uzbeck (1%). The latter label was very frequently assigned to short paragraphs, for instance such that contained only names, initials or dates. The most frequently found foreign languages are French, German,

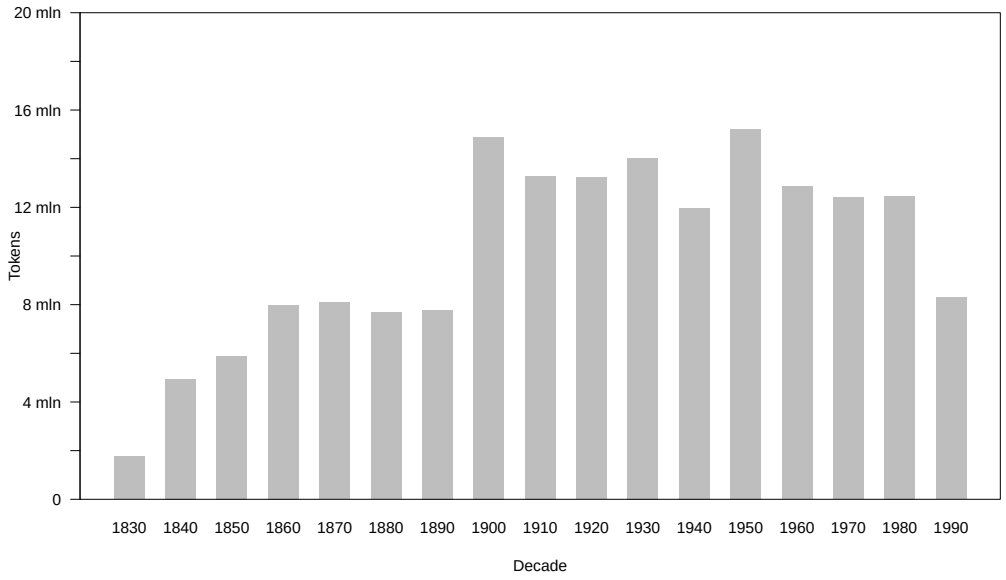

Figure 1: Corpus size per decade in terms of tokens excluding punctuation.

the next section. Lemmatization, morphological analysis and part-of-speech tagging is performed by the Alpino parser, whose application to the material is also discussed in the next section.

The selection of sources in the Gothenburg Dutch C-CLAMP is identical to the original "Leuven" Dutch C-CLAMP of Piersoul et al. (2021). Our alternative pre-processing path does however mean that the two versions of the corpus are not token identical, although the overlap is very high. The final Gothenburg Dutch C-CLAMP contains just over 8 million sentences in almost 173 million tokens. The temporal distribution if this material is given in Figure 1.

3. Parsing the C-CLAMP

For the linguistic enrichment of the data, we use the wide-coverage dependency parser for Dutch, Alpino.[4] The Alpino parser has primarily been developed on present-day Netherlandic Dutch newspaper text, which potentially makes Dutch C-CLAMP *out of domain* for three reasons: 1) In terms of genre, the corpus is comprised of literature and language periodicals, which contain language that is comparatively complex and use conventions and terminology not found in newspapers. 2) About a third of Dutch C-CLAMP is 19th century material and two-thirds pre-WWII. This means large parts of the corpus will differ from the present-day language in terms of vocabulary but also syntax, especially where it comes to the frequency of different constructions. On a surface level, historical material also presents a problem for Alpino when it comes to spelling, as there have been considerable orthographic reforms between 1800 and now in the Dutch language area (see Section 4.1 for a closer look). 3) Dutch C-CLAMP contains periodicals published in Belgium as well as the Netherlands and thus contains both Belgian Dutch and Netherlandic Dutch traits. Although originally and primarily developed on Netherlandic Dutch, Alpino has been used and further developed in projects involving

Afrikaans and English. Although there is some actual Afrikaans in the dataset, most of the material labelled as such was Dutch.

4. `https://www.let.rug.nl/vannoord/alp/Alpino/`

Belgian Dutch material, as well (van Noord et al. 2013, de Kok and van Noord 2017), so we might expect this not to be a big problem. We return to these issues in Section 4.2.

In the current section we will discuss our efforts to ameliorate Alpino's handling of the out-of-domain Dutch C-CLAMP data. In the next section, Section 4, we attempt to gauge how well the three potential problem sources are addressed, by correlating (a proxy for) parsing quality with variables like time and periodical and region.

3.1 Bringing the corpus to the parser

As a solution to the surface-level variation problems for Alpino, van Cranenburgh and van Noord (2022) develop a method in which they pre-process texts by adding a layer of *meta-annotations*: instructions to the parser that, for instance, tell it to treat a token in the text as if it had a different form. These annotations do not replace the original text but only supply additional information, so that the original text remains recoverable from the parsed structures. We adopt this method when parsing Dutch C-CLAMP. The following three types of meta-annotations are used:

- `[ @alt` *`FormA FormO`* `]`: treat token *`FormO`* as if it reads *`FormA`*. This gives a straightforward solution for spelling variation.
- `[ @mwu` *`FormO1 FormO2 ...`*`]`: treat adjacent tokens *`FormO1 FormO2 ...`* as if they were a multiword unit. This allows us to handle multiword units unknown to the parser, as well as cases where present-day spelling convention requires an expression to be written as one word, while historical convention did not.
- `[ @mwu_alt` *`FormA FormO1 FormO2 ...`*`]`: treat *`FormO1 FormO2 ...`* as if it reads *`FormA`* (a combination of the previous two instructions).
- `[ @phantom` *`FormA`* `]`: parse the string as if *`FormA`* had been present at this location. This can be used in combination with `@alt` annotations to handle tokens that need to be split in order for the parser to succeed.

To illustrate, we can take the following constructed example:

(3) *Dat heeft-ie wel is waar zoo gezegd...*
that has-he although so said
'Even though he said it like that...'

For this to be parsable by Alpino, it needs to be presented as the equivalent of *Dat heeft hij weliswaar zo gezegd...* The meta-annotations below achieve this:

(4) `Dat [ @alt heeft heeft-ie ] [ @phantom hij ] [ @mwu wel is waar ] [ @alt zo zoo ] gezegd...`

In the example, the meta-annotation also deals with the enclitic subject, which is not *just* a matter of orthography. As long as a problem in the input can be fixed on a surface level, the meta-annotation mechanism is suitable. A prominent case in point in our material is the handling of case marking in the older material, which requires removing inflection using `@alt` annotations and, in the case of certain genitives, effectively rewriting an NP into a PP by inserting the preposition *van* 'of' into the text using `@phantom`, as illustrated in (5):

(5) *o dagen mijner schande!*
oh days my.GEN disgrace
`o dagen [ @phantom van ] [ @alt mijn mijner ] schande!`
'Oh, days of my disgrace!'

Our implementation of this preprocessing step builds directly upon the tools for meta-annotation developed and released by van Cranenburgh and van Noord (2022). In addition to hand-written meta-annotation rules, van Cranenburgh and van Noord also propose a method for extracting spelling variants from the target corpus, which is extended in the work presented here, too.

3.2 Iterative improvement of parse quality

The meta-annotation mechanism gives us a way to help Alpino handle the out-of-domain material better, but it does not tell us *which* aspects of the input cause trouble. For this, we rely on *error mining* (de Kok et al. 2009), a method to identify sources of parsing failures based directly on the output of the parser itself, that is, without consulting a gold standard or human input on the parses. Central to the technique is the concept of *coverage*, which is the proportion of sentences that are assigned a full parse by the parser. A sentence that is not assigned a parse – but only receives fragment analyses – is generally indicative of a problem somewhere in the parsing pipeline, for instance in tokenization, with the parser's lexicon, or with the grammar. In the error mining method of de Kok et al. (2009), one starts by parsing the target material, after which lists of n-grams are produced that are associated with decreased coverage. Inspecting these lists allows the parser developer to spot problem areas, which can then be solved, for instance by making changes to the tokenizer or by adding new meta-annotations. The method is iterative: after the most likely error sources are addressed, the material is reparsed and re-mined for more potential error sources.

This iterative, data-driven method of error mining is not a deterministic process. It falls upon the researcher applying the method to choose what errors to address and how at each iteration, and when to stop the process. There is no guarantee that another researcher applying the same method to the same material will arrive at the same end results. Nevertheless, as a workflow, error mining has proven to be an effective strategy in the past, when developing symbolic parsers (van Noord 2004, Sagot and de la Clergerie 2006, de Kok et al. 2009, de Kok and van Noord 2017) as well as generators (Narayan and Gardent 2012). In our case, too, we managed to improve Alpino's coverage of Dutch C-CLAMP over several iterations. The applied fixes fall into different categories:

- Changes to the parser:
    - Addition of abbreviations to the sentence tokenizer, for instance *voorsz.* (*voorszegde* 'said'), *Tg.* (*[Nieuwe] Taalgids*), *vss.* (*verzen* 'verses'), *Alb.* (*Albert[us]*). Note that the tokenization problems that failing to recognize such abbreviations causes cannot be fixed by meta-annotations, as these apply after tokenization.
    - Additions to Alpino's lexicon, for instance conjunctive forms current in older text: *blijve* 'stay', *denke* 'think', *vergete* 'forget', *verlieze* 'lose' *versta*, *wete* 'know', *worde* 'become'; now all-but-lost selection patterns such as *betuigen* 'express' with a subordinate clause or *vergaderd* 'meet.PERFPART' selecting auxiliary *zijn*, 'be gathered'.
    - Adding material from C-CLAMP to the unsupervised training of Alpino's parse pruning component (van Noord 2009).

  These changes were contributed back to Alpino and are part of the current release of Alpino.

- Changes to the meta-annotation:
    - Expansion of the list of known spelling variants through additional variation patterns and extraction from C-CLAMP:
      van Cranenburgh and van Noord (2022) extract spelling variants from the target corpus by applying known graphemen sequence correspondences to words Alpino fails to recognize. If the form is recognized by Alpino after applying the correspondence, the form is added to the list of spelling variants. They report extracting 4k spelling variants this way (e.g.

zoo 'such' → *zo* from the correspondence [double vowel] → [single vowel], or *tusschen* 'between' → *tussen* from *sch* → *s*) this way.

In the work reported here, we use the same technique to extract a total of 55k spelling variants by considering a larger set of correspondence patterns and using C-CLAMP as the source corpus.

- Targeted handling of case marking in NPs:
  A particular problem parsing older stages of modern Dutch is (accusative) case marking on determiners and adjectives, which leads to forms not current in modern orthography. Since (non-pronominal) case carries little information in Dutch at this stage, we can treat this as essentially an orthographic issue and add meta-annotations to the string that drop this marking: *de-n ware-n kunstenaar* 'the-ACC.MASC true-ACC.MASC artist(MASC)' → *de ware kunstenaar*.

  Case-marked adjectives are frequently homographs of other items: in isolation, the form *waren* could also be (the highly frequent) verb 'were' or the noun 'wares'. We therefore add these annotations very restrictedly: they are only added to sequences of tokens that together *could* form an NP. We mark up a number of templatic NP structures, such as [Det N], [Det Adj N], [Det Adj Adj N], [Det Adj Conj Adj N], etc, where the different parts of speech stand for lists of suitable surface forms. The lists for Det and Conj are manually constructed, but the open word class lists are are mined from C-CLAMP as follows: For Adj, we take all word types with suffix *-en*, drop the final *-n*, apply the list of spelling variants and then use Alpino's lexical analysis component to retain only potential adjectives. In contrast to the construction of spelling variant lists, we here also include cases that would be recognized by Alpino. Again, *waren* is a case in point: it is included in Adj even though Alpino recognizes it as-is, as a verb or a noun. The list of Ns is constructed using a similar strategy. Including spelling variants, the resulting lists consist of approximately 760k Ns and 30k Adjs.

The meta-annotation pipeline used to pre-annotate the C-CLAMP can be downloaded from the location given at the end of the paper.

- Changes to the data:
  - Parentheticals were overrepresented in the parse failures, and not typically contain content relevant to our intended application. We therefore remove material within parentheses, almost 2% of the original material in terms of alphanumerical tokens.[5]

We were able to improve Alpino's coverage on Dutch C-CLAMP from an initial 78.1% (without meta-annotation and before any improvements elsewhere in the pipeline) to 84.2%. This is a considerable improvement, even though the resulting overall coverage is still low compared to what Alpino achieves on in-domain material. For instance, for present-day Netherlandic Dutch newspaper articles, Alpino's coverage was already reported to be 95%–96 in van Noord (2004), and currently lies above 98%.

3.3 Effect of meta-annotation on the parse trees

Although the meta-annotations only add information and the original text is recoverable from both the annotated string and the resulting parse, it is still the case that these meta-annotations work

5. This is a simplistic and rather draconic measure. A more involved, but much better solution would have been to remove the parenthetical material from the pipeline, but to reinsert it into the parses afterwards, either in analysed or unanalysed form (cf. the handling of punctuation marks). This would for instance ensure that all tokens in the material are represented in the XML containing the parses. We reserve this improvement for any future versions of the parsed corpus.

through in the parses, sometimes in unexpected ways. A future user of the parsed corpus must be aware of these effects, which is why we describe them in this subsection by looking at some XML snippets taken from parses.

`@alt` (6 520 809 occurences[6]) The majority of `@alt` meta-annotations concern simple spelling variants such as *zoo* → *zo* or the removal of case marking from adjectives, these do not lead to any real discrepancies in the parse tree.[7] However, some applications can affect the parse tree. For instance, the string *[hij acht het] zijner niet waardig* '[he deems it] not worthy of him' (lit. 'him.GEN not worthy') is presented to the parser as `[ @alt hem zijner ] niet waardig`. This solves the problem that present-day Dutch uses the object form of the pronoun rather than the possessive/genitive used in the historical example. Below is a simplified[8] excerpt of the XML produced for this sentence by the parser. We see that the word is given as *zijner*, but the (morphological) analysis corresponds to *hem*. This leads to a succesful and correct parse (*zijner* is an `obj2` complement to *waardig* 'worthy'), but it also means `zijner` is incorrectly lemmatized as `hem` and classified as the `obl`(ique) form of the pronoun.

```
<node begin="4" cat="ap" end="7" id="8" rel="predc">
    <node begin="4" end="5" id="9" lemma="hem"
        postag="VNW(pers,pron,obl,vol,3,ev,masc)" rel="obj2" word="zijner"/>
    <node begin="5" end="6" id="10" lemma="niet" postag="BW()" rel="mod" word="niet"/>
    <node begin="6" end="7" id="11" lemma="waardig" postag="ADJ(vrij,basis,zonder)"
        rel="hd" word="waardig"/>
</node>
```

`@mwu`, `@mwu_alt` (1 592 and 36 091 occurences, respectively) The meta-annotations `mwu` and `mwu_alt` are used for a range of multiword expressions, often for cases that historically were "words with spaces" (Sag et al. 2002), but which Alpino in accordance with present-day convention expects as single graphic words, for instance *op nieuw* → *opnieuw*.

The annotation is also employed for certain syntactic constructions, for instance for the prenominal genitive NP, which was historically more frequent and productive than it is now. For example, *[in] 's waters macht* '[at] the mercy of the water' (lit. 'the.GEN water.GEN power') receives meta-annotation `[ @mwu_alt zijn 's waters ] macht`, which tells the parser to treat the whole genitive NP *'s waters* as if it were a possessive pronoun (*zijn* 'his'), facilitating the intended attributive parse.

```
<node begin="8" cat="np" end="11" id="15" rel="obj1">
  <node begin="8" cat="mwu" end="10" id="16" rel="det">
    <node begin="8" end="9" id="17" lemma="de" postag="LID(bep,gen,evmo)" rel="mwp"
        word="&apos;s"/>
    <node begin="9" end="10"id="18" lemma="waters"
        postag="VNW(bez,det,stan,vol,3,ev,prenom,zonder,agr)" rel="mwp" word="waters"/>
  </node>
  <node begin="10" end="11" id="19" lemma="macht" postag="N(soort,ev,basis,zijd,stan)"
      word="macht"/>
</node>
```

As the XML fragment shows, the prenominal genitive NP is correctly integrated in its parent NP, but it occurs at the cost of remaining effectively unanalysed (each of its parts have the `mwp` – multiword part – function, and in addition the analysis of *waters* as a possessive pronoun `VNW bez` with lemma `waters` is nonsensical).

6. The numbers refer to the total number of meta-annotations of this type in the corpus. Some rules may trigger multiple meta-annotations, such as the rule that applies to the genitive in example (5).
7. We do not quantifiy this further here, to avoid distracting from the discussion. For a clearer view of the proportions of the different causes for meta-annotation, we refer to the discussion and graph in Section 4.1.
8. We silently leave out XML attributes not relevant to the discussion.

@phantom (159 610 occurences) Insertion of a token is for instance used in combination with other meta-annotation to split tokens. This offers a way to handle fused complementizer-subject clitic forms: *da'k ['t zeg]* 'that I [say it]' becomes `[ @alt dat da'k ] [ @phantom ik ]`, rewriting the complementizer without the clitic and inserting an 'invisible' pronoun to stand in for the subject. This subject will not show up in the parse tree, but it will allow Alpino to construct a finite clause as if the subject were realized.

A (much) more frequent application in our material again concerns genitives. It allows us to rewrite a genitival determiner into a sequence containing a phantom preposition 'of'. The example *[de zwakte] zijns gemoeds* '[the weakness] of his character' (lit. 'his.GEN character.GEN'), is given to the parser as `[ @phantom van ] [ @alt zijn zijns ] [ @alt gemoed gemoeds ]`, that is, we insert the preposition *van* 'of' and remove the genitive suffixes from the determiner and noun.

```
<node begin="39" cat="pp" end="41" id="66" rel="mod">
  <node begin="39" cat="np" end="41" id="67" rel="obj1">
    <node begin="39" end="40" lemma="zijn"
        postag="VNW(bez,det,stan,vol,3,ev,prenom,zonder,agr)" word="zijns"/>
    <node begin="40" end="41" lemma="gemoed" postag="N(soort,ev,basis,onz,stan)"
        word="gemoeds"/>
  </node>
</node>
```

In the XML, we see a unary branching, headless PP above the NP. There is no information in the parse that the inserted preposition is the cause of this PP. The syntactic relations within the NP are as intended, but note that morphologically the meta-annotation cause the parser to incorrectly analyse the possessive pronoun and the noun as having unmarked case (**stan**[dard]), and not **gen**(itive).

As the examples above show, the use of meta-annotations may introduce discrepancies and unexpected configurations in the parse. However, each of the examples above is also an example of a succesful parse of a sentence that would not have been parsable without the meta-annotations. Overall, the usefulness of meta-annotations therefore easily makes up for the discrepancies. Since all information about the meta-annotation is preserved in the final XML, an interesting future project would be to see if it is possible to fix them in a post-processing step.

4. Changes in spelling conventions and syntactic complexity

The pre-processing of Dutch C-CLAMP does not only contribute to improving the parsing quality of the corpus but can indirectly also offer insights into its language. In this section, we repurpose meta-annotations to offer a first glimpse into the implementation of spelling reforms, and relate parser coverage to changes in syntactic complexity in Dutch C-CLAMP. These brief explorations point to promising directions for future research.

4.1 Meta-annotations and implementation of spelling reforms

The meta-annotations mostly address orthographic changes or changes close to orthography, such as accusative case marking in NPs. An interesting question is therefore if we can see the rate of adoption of the 19th and 20th century spelling reforms in Dutch C-CLAMP material reflected in the number of edits made by our preprocessing tool chain. The presentation of spelling reforms in this section is based upon Nunn (1998, Appendix H) and *Genootschap Onze Taal*'s website on this topic.[9]

Figure 2 plots the number of meta-annotations per 100 tokens through time for the whole corpus. The shaded area covers the 19th century, for which De Gids, published in The Netherlands, is the

9. `https://onzetaal.nl/schatkamer/lezen/taal-en-maatschappij/spelling-geschiedenis`, cons. January 2026.

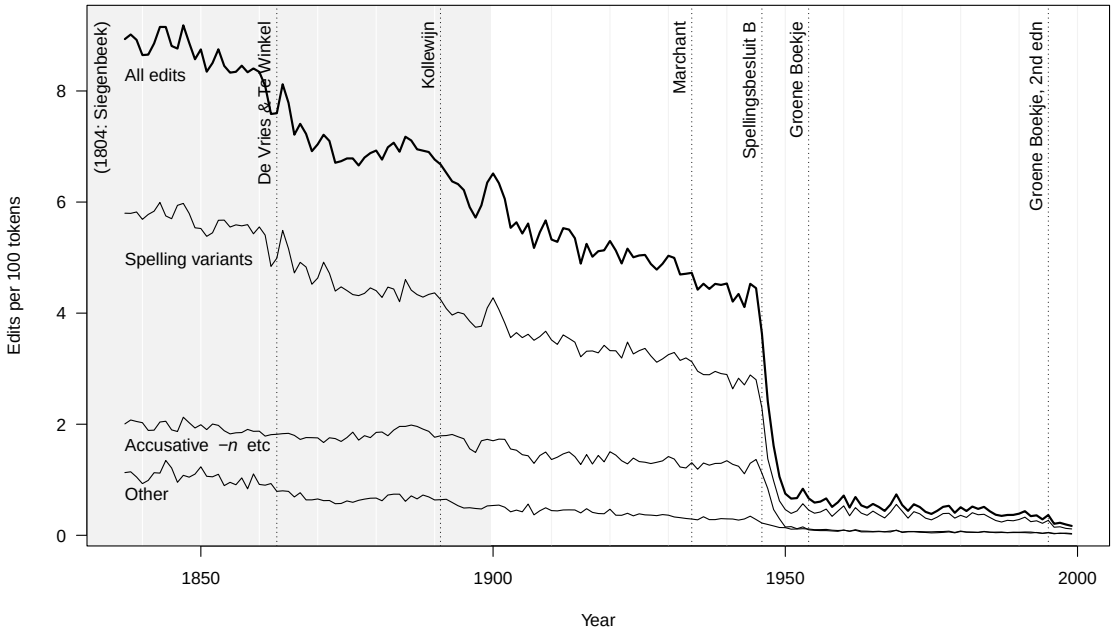

Figure 2: The number of meta-annotations per 100 tokens over time. The upper line "All edits" is the total number of edits, the lower lines show the contributions of the major components. The dotted vertical lines indicate the introduction/publication of spelling standards. The shaded area before 1900 consists only of material from De Gids (NL).

only source in the corpus. This part of the plot is thus a characterization of spelling particular to De Gids. This early period is marked by three spelling standards / spelling-related publications: the government-sanctioned Siegenbeek orthography (1804), which predates the earliest material in the corpus; the orthography of De Vries and Te Winkel (1863), connected to the historical Dictionary of the Dutch Language, officially adopted in Belgium in 1864 and in The Netherlands in 1883; and the comments on the complexity of Dutch orthography by Kollewijn (1891). The latter marks the beginning of a long debate about spelling simplification, which is only settled well into the 20th century with the 1934 Marchant orthography in the Netherlands and the post-WWII orthography laws of 1946 (Belgium) / 1947 (The Netherlands). The latter were solidified in 1954 in the glossary known as *het Groene Boekje*. This official glossary has been updated several times since, but the changes are minor when compared to the pre-WWII changes in orthography (and, moreover, the variants are known to Alpino). One of Kollewijn's eventually successful simplification proposals was dropping the marking of the accusative *-n* from the official standard.

Against this background of spelling reforms, we start by observing the glaringly obvious: the post-war reform basically brings us to the present-day spelling and from that time hardly any edits are needed. In the period between the end of the WWII and the publication of the Groene Boekje, the overall edits fall from about 5 per 100 tokens to fewer than 1. The plot also shows a decomposition of the total number of edits into three "sources" as seperate lines: 1) the general list of spelling variants, 2) the targeted handling of accusative marking, and 3) all the other sources. Edits from these latter sources dissappear almost completely.

Earlier orthographic reforms do not leave clear marks in the graph. The overall trend in the combined edits is a gradually falling number of edits. Looking at just the edits from the list of spelling variants, we might be tempted to see slight effects of the orthographic reforms: After the publication of De Vries and Te Winkel, the number of edits first falls a bit quicker, indicating material that quickly becomes more like the present-day standard, but stablizes a bit until the publication of Kollewijns commentary, after which the number of edits falls at a slightly faster pace again. The orthography introduced by Marchant in the the Netherlands marks a slight increase in modernization pace, just before the cliff marked by the end of WWII. Looking at the use of the accusative *-n*, we conclude that there are two relatively stable periods pre-WWII, around 2 edits per 100 tokens during the 19th century and a fractionally lower level during the 20th century.

There are also peaks in the number of edits that invite further study. For instance, the gradual increase and then decrease in total edits during the 1880s, which seems to mostly originate in an increase in the use of accusative marking. We can only guess at the reason for this temporary bout of conservatism in De Gids, and will therefore not discuss this further. The sharp peak around the turn of the century on the other hand occurs mainly in the edits coming from the list of spelling variants. It can be related to a very brief, 3-year, increase of archaic pronominal forms with *-e*, like *ene* 'a/one' → *een*, *mijne* 'my/mine' → *mijn*, *zelve* 'self' → *zelf*, etc. Again, we are not certain what the reason for this might be. Interestingly, it cannot be attributed to the inclusion of Belgian periodicals, as the peak begins in 1899, one year before the publication date of the first Belgian articles in the corpus.

4.2 Parser coverage and changing syntactic complexity

The concept of coverage can not only be used in error mining and the parsing improvement iterations, we can also leverage it to gain insight into parser performance across some of our variables of interest. In particular, we will look at coverage across time to see how much harder the older material is for the parser, and at coverage per periodical (and thus, approximately, region) to see if any periodicals stand out and if there is a difference between the publications from Belgium and The Netherlands.

Alpino's coverage per year is given in the top plot in Figure 3. As we can see, there is a very tight correlation between coverage and time (Spearman's $\rho = .985$), with the coverage of the material of the 1990s crossing the 90% mark. At the lower end, however, the parser struggles with the oldest material. Until 1860, coverage hoovers around 70%, which is very poor.

The second plot in Figure 3 shows the development of (graphical) sentence length in terms of non-punctuation tokens. Here, too, we see a clear correlation between sentence length and time, topping out at an averages just above 30 tokens per sentence in the oldest material to less than 20 tokens per sentence in the newest. The drawn Poisson regression line assumes a log-linear relationship between length and time, but as we can see this is a mischaracterization. In fact in seems that generally sentence length falls during the 19th century, but is relatively stable during the 20th century, up until the last decade where it appears to fall a bit further. A decrease in sentence length around the same time was also noted for Dutch novels (van Cranenburgh 2022a), and is found in neighboring languages, such as English, German, and French (see Rudnicka 2018 for an overview). It is conceivable that a closer look at the data will show that the observed general picture with regards to sentence length can be explained from underlying factors like author properties or genre. This is an interesting venue for further study that C-CLAMP facilitates.

Long sentences are a clear risk factor for Alpino failing to find a full parse (van Noord 2006), and part of the reason for the coverage increase should be sought in the decrease in average sentence length. However, considering the difference in shape between the coverage and average sentence length curves, this is unlikely to be the whole explanation. Indeed, a logistic regression shows both

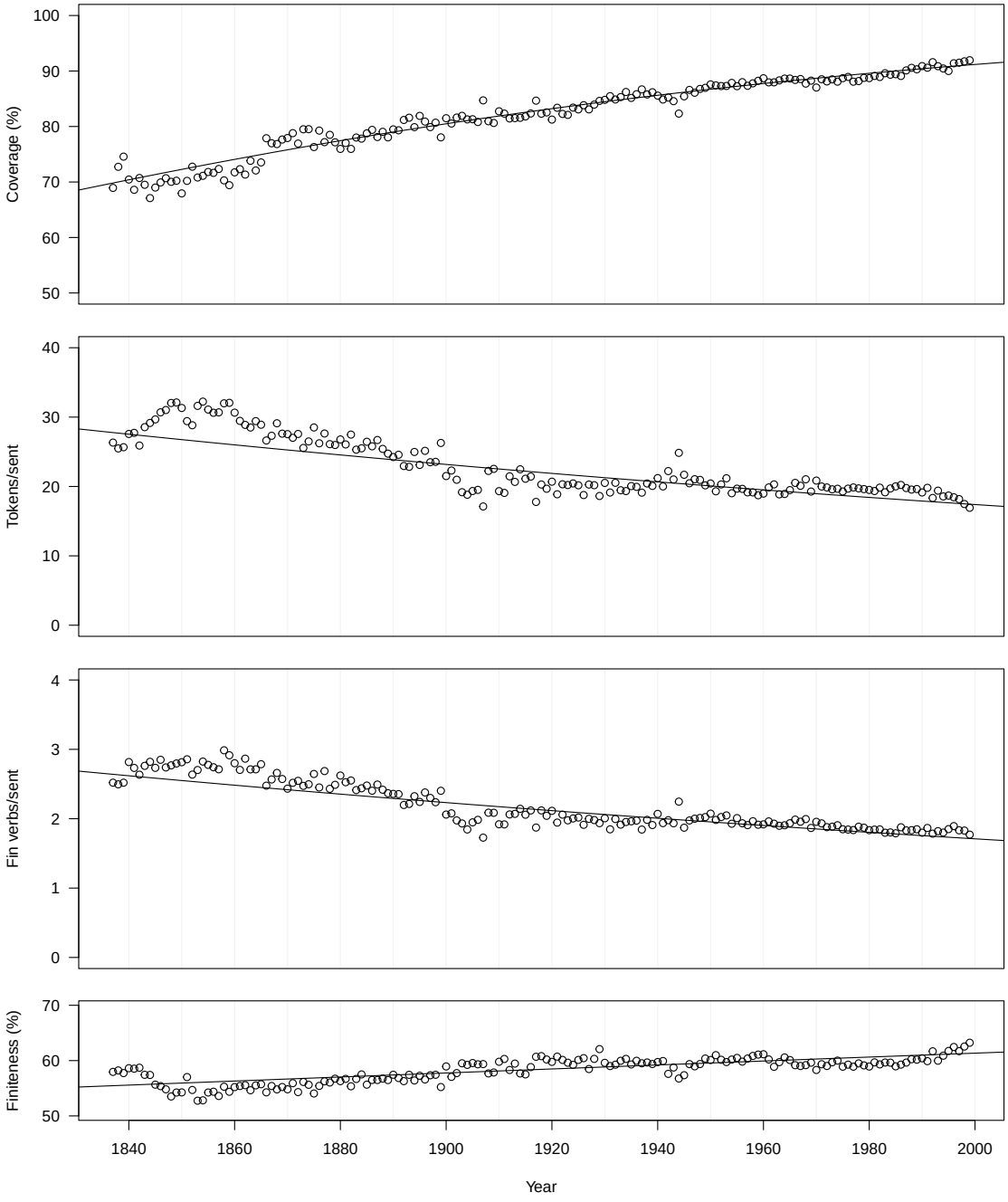

Figure 3: Coverage (top), average sentence length (second from top), average number of finite verbs per sentence (second to bottom) and proportion of verbs that are finite (bottom) over time. The regression lines (Poisson for counts, logistic for proportions) summarize the trends the plotted variables against time.

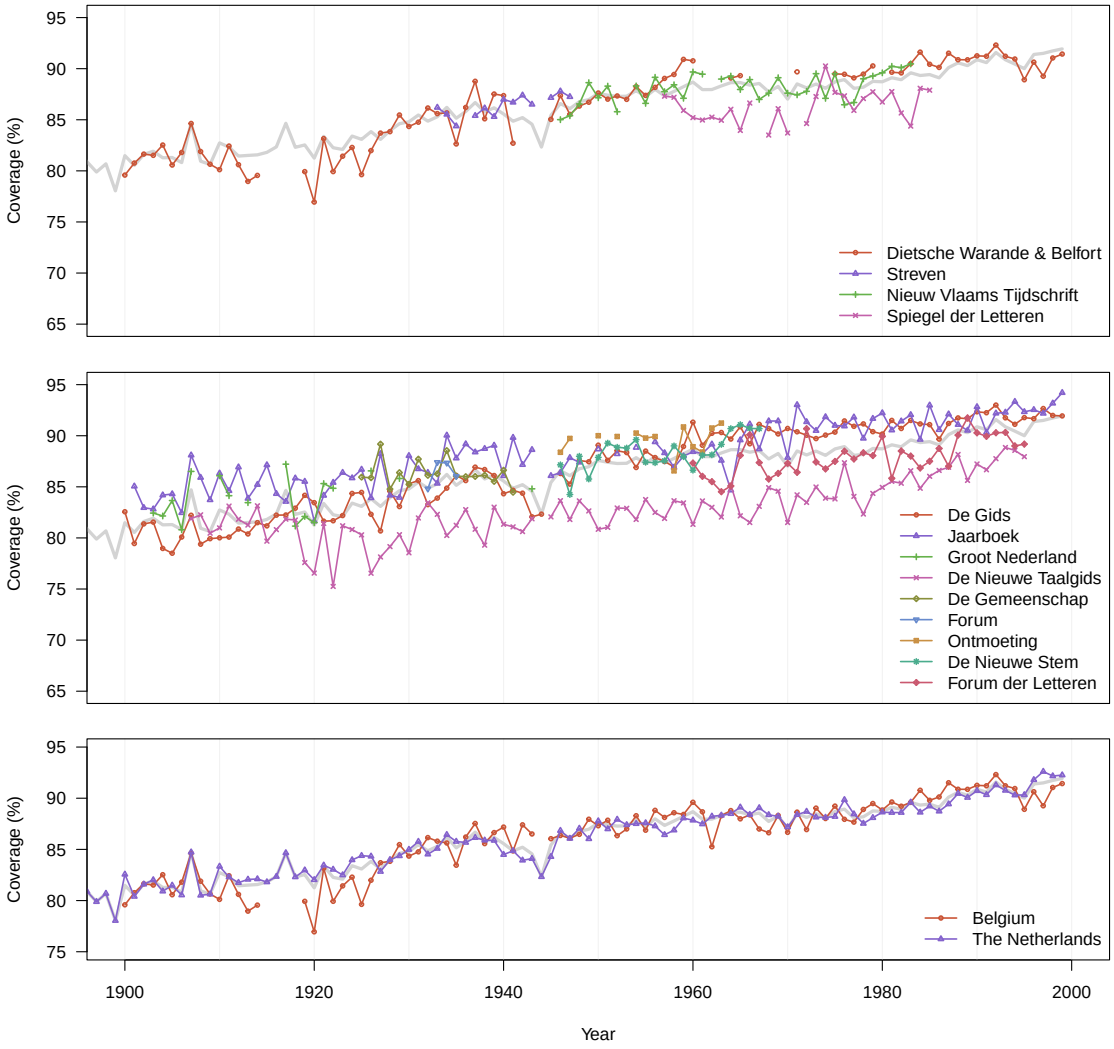

Figure 4: Coverage over time per periodical and region (periodicals from Belgium – top, from The Netherlands – mid, regionwise averages – bottom). The average over all periodicals from both regions is given in grey

average sentence length and time to be significant in modelling coverage, although we are not able to offer a causal explanation of the additional effect of time.[10]

In Figure 4, we split the coverage data into trends over time per publication and per region. The top graph contains the Belgian periodicals and the middle graph those published in the Netherlands. In both graphs, overall coverage (that is, combined from both countries) has been plotted to provide

10. We fitted a logistic regression model with *single tree?* (yes/no) as the dependent variable and *average sentence length* and *year* (residualized on average sentence length) as independent variables. Both factors are significant $p \ll .001$ (Wald tests). The model shows coverage decreases with average sentence length and additionally increases with time.

a point of comparison. Coverage for most journals fluctuates around the overall coverage. Amongst the Belgian publications, *Spiegel der Letteren* stands out as its coverage falls almost completely below the average. In the Dutch data, the same holds for *De Nieuwe Taalgids*, albeit under a considerably longer stretch of time and staying further away from the average trend. Apparently, these periodicals are particularly challenging for Alpino. The bottom graph shows coverage over times aggregated by publication region. Reassuringly, there does not seem to be a systematic difference between the two regions. If Alpino has a harder time with Belgian Dutch, it does not show in these graphs.

Finally, we look at complexity in terms slightly closer to our use case of studying verb constructions. The two lower panels in Figure 3 show the average number of finite verbs per graphical sentence over time, and the proportion of all verbs that is finite. The former can be seen as an alternative measure of sentence complexity, as it can be taken as a proxy for the number of finite clauses (coordinated or subordinated) in a sentence. Over time, this falls from roughly 3 to 2 finite verbs per sentence on average.[11] Visually, the development of finite verbs per sentence is very similar to that of tokens per sentence. As a verb construction requires at least one non-finite verb, the proportion of verbs that are finite can tell us something about the occurance of verb constructions. Although the effect is only small, the bottom panel shows that proportion of finite verbs goes up over time: verb constructions become less common. This combination, fewer finite verbs to form co- or subordinate finite clauses, and (relatively) fewer non-finite verbs for non-finite constructions, means that sentences over time become simpler in terms of their verbal syntax in our material.

5. Compiling the Dutch Verb Construction Database

We now move from the parsed Dutch C-CLAMP, to the creation of the Dutch Verb Construction Database from this resource. Below, we describe the extraction procedure of verb constructions from the Alpino parses, provide a short overview of the extracted construtions, and evaluate the automatic extraction method against a manually annotated dataset. Our manual evaluation targets the dataset instead of the parsed corpus as a whole, because our prime interest is how accurately verb constructions were parsed in the corpus (cf. Bloem 2016).

5.1 Extraction

By verb construction, we mean the combination of two verbs, such as the combination of an auxiliary with a non-finite main verb in examples (1) and (2). Verb constructions can be stacked into longer verb chains, as illustrated in Figure 5. We extract such chains using the method developed for Coussé and Bouma (2022), which uses the following criteria:

1. A chain consists of m hierarchically connected verbs, $\mathrm{v}_1 \ldots \mathrm{v}_m$, that each have exactly one verb above them in the chain (*head*) and one verb below them in the chain (*dependent*), except for the highest verb, v_1, which has no head, and the lowest verb, v_m, which has no dependent.
2. The highest verb v_1 is finite, the rest, $\mathrm{v}_2 \ldots \mathrm{v}_m$, are infinitives or past participles.
3. Infinitival lower verbs may be bare, accompanied by the infinitival marker *te* or by the combination *aan het*.
4. When accompanied by *te*, verbs must not be in an infinitival clause introduced by complementizer *om*.
5. The chains are made as long as possible under requirements 1–4.

The verb chains described by these points are filtered by a further requirement:

11. An average of 2 finite verbs per sentence may seem high, but it should be kept in mind that this distribution is right-skewed. The modal number of finite verbs per sentence is 1.

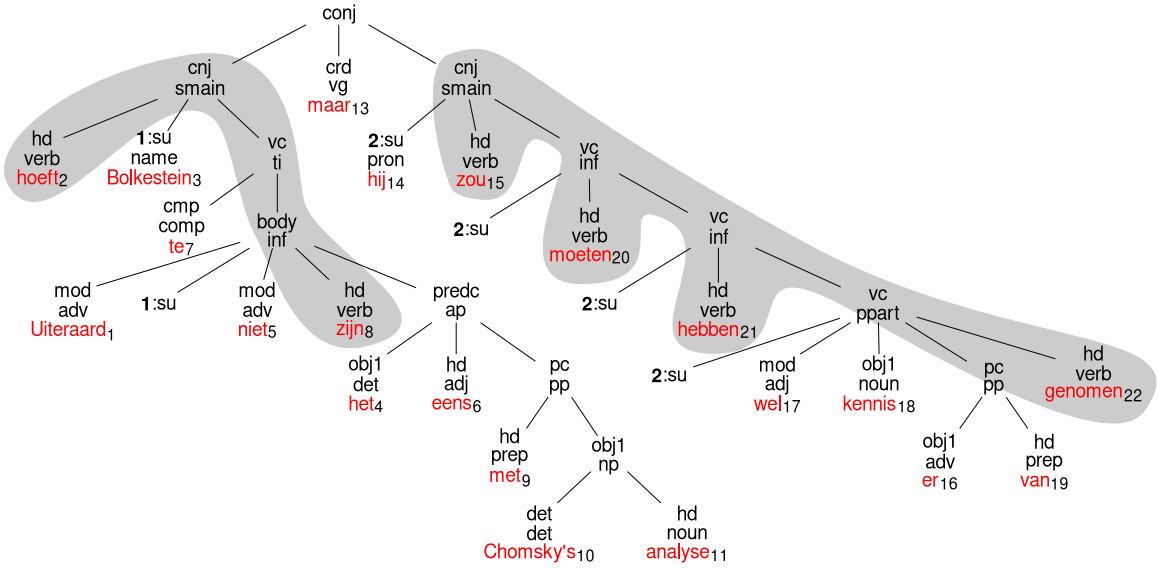

Figure 5: The tree for [S *Uiteraard* ***hoeft***$_1$ *Bolkestein het niet eens te* ***zijn***$_2$ *met Chomsky's analyse*], *maar* [S *hij* ***zou***$_1$ *er wel kennis van* ***moeten***$_2$ ***hebben***$_3$ ***genomen***$_4$]. 'Of course, Bolkestein doesn't have to agree with Chomsky's analysis, but he should have informed himself about it.', which contains a two-verb chain in the left conjunct and a four-verb chain in the right conjunct.

6. All verbs but the lowest, that is, $v_1 \ldots v_{m-1}$, must be listed in Haesereyn (1997, Section 18.5.8) as potentially clustering.[12]

Because we extract whole verb chains, we include cases in which multiple verb constructions are combined into more complex units, which come with their own grammatical traits. The database thus not only supports studying verb constructions, but also investigating the complex units themselves (see for instance the word order in three-verb chains in Section 6) or aspects of the constituting verb constructions sensitive to their position in the chain.

Figure 5 illustrates what verb chains as defined above look like in the annotations produced by Alpino, which follow guidelines based upon the guidelines of the Spoken Dutch Corpus (van Noord et al. 2013). The figure contains two chains, both highlighted by a gray background. In both, v_1 is the head of a finite clause type (*smain*, main clause), and the lower verbs are inside complements marked with the VC dependency (verbal complement). The precise position of the verb inside this complement depends on the verb form and the presence of marking like the infinitival marker *te*. Criteria 1–6 were implemented as XQuery queries over Alpino XML representations of structures like those in Figure 5. The queries were excecuted using BaseX (BaseX GmbH 2023).

We extract just over 16 million verb chains, or just over 5 million if we only consider chains of length 2 and longer, from Dutch C-CLAMP. For the longer chain lengths, the number of cases drops rapidly. The number of extracted examples per chain length are tabulated in Table 2.

12. A head verb *clusters* when it and its dependent verb are not divided by a clause boundary. For a *non-clustering* head verb, the dependent verb occurs in a non-finite subordinate clause. A *potentially clustering* head verb can occur in either configuration. The relevant categories in Haesereyn (1997) are *verplicht groepsvormend* 'obligatorily clustering' and *niet-verplicht groepsvormend* 'optionally clustering'. We refer to Haesereyn (1997) for a more elaborate discussion of the phenomenon of clustering.

Chain length	Count	%
1	10 907 845	67.4
2	4 557 738	28.2
3	682 525	4.2
4	35 480	0.2
5	475	<0.1
6	14	≪0.1
Total	16 184 077	100.0

Table 2: Size of the extracted verb chain dataset.

As mentioned, all heads in the extracted verb chains are lexically specified as *potentially* clustering. The data therefore contain verb chains where all verbs appear without any intervening clause boundaries (6) as well as chains where the verbs are distributed over multiple clauses (7):

(6) *het tweetal, dat ik als doopgetuigen* ***heb meenen*** *te* ***mogen laten*** ***optreden***
the couple that I as baptism witnesses have believe TE may let act
'the couple about which I was convinced I could let them act as witnesses of the baptism'

(7) *En alleen de nooddrang* ***heeft***$_1$ *mij* ***kunnen***$_2$ ***doen***$_3$ ***besluiten***$_4$ [*dit zwijgen te* ***doen***$_5$ ***eindigen***$_6$ *bij sommigen*].
and only the urgency has me can do decide this silence TE do end with some
'And only the urgency of it could make me decide to force some of them to end their silence.'

The 5 million verb chains of lengths 2–6 are built from a total of 102 verb constructions. As is typical for lexical counts, we see a very wide frequency span. The five most common constructions – *hebben* PTCP/INF-PRO-PTCP 'have [done]' (perfect), *worden* PTCP 'be [done]' (passive), *zullen* INF 'shall [do]', *kunnen* INF 'can [do]', *zijn* PTCP/INF-PRO-PTCP 'be [done]' (temporal) – each occur 400 thousand–1 million times. The five least common constructions – *kunnen* PTCP 'can [be done]', *zetten* TE-INF 'put away [to do]', *houden* AAN-HET-INF 'keep [smth doing]', *lijken* AAN-HET-INF 'seem [to be doing]', *blijken* AAN-HET-INF 'turns out [to be doing]' – each only occur 10 times or less in the database. All 103 constituting verb constructions and their frequency in the corpus are listed in Appendix A.

5.2 Evaluation

We manually annotated a selection of sentences and compared the automatically extracted data against the resulting data set. To estimate the *precision* of our method, we annotated 100 randomly selected datapoints per 20-year period from the extracted two-verb chains and likewise for the extracted three-verb chains. To estimate *recall*, we randomly selected finite verbs from Dutch C-CLAMP and manually checked whether they were v_1 in a verb chain. From each 20-year period, we annotated as many sentences as needed to get a set of 100 positive instances. This was done separately for the two- and three-verb chains. In positive cases, the other verbs (v_2, v_3) were also marked up. To reduce the amount of sentences that needed to be inspected to collect the recall testset, we prefiltered the random selection using the lexical annotation alone. For the two-verb chain data, we selected sentences that contained a finite verb that meets requirement 6 and one or more additional past participle or infinitival verb, per requirement 2. For the three verb data, we picked sentences that contain a finite verb, requirement 6, one or more past participles or infinitives,

Chain length	Precision	Recall	Prevalence	Corr. prevalence
two verbs	97.0	88.5	52.8	29.5
three verbs	99.4	87.3	15.9	4.3

Table 3: Overall precision and recall of the extraction of two- and three-verb chains, in percent. Prevalence refers to the prevelance in the evaluation data, corr[ected] prevalence gives an estimate of prevelance in the overall data by correcting for data set pre-filtering.

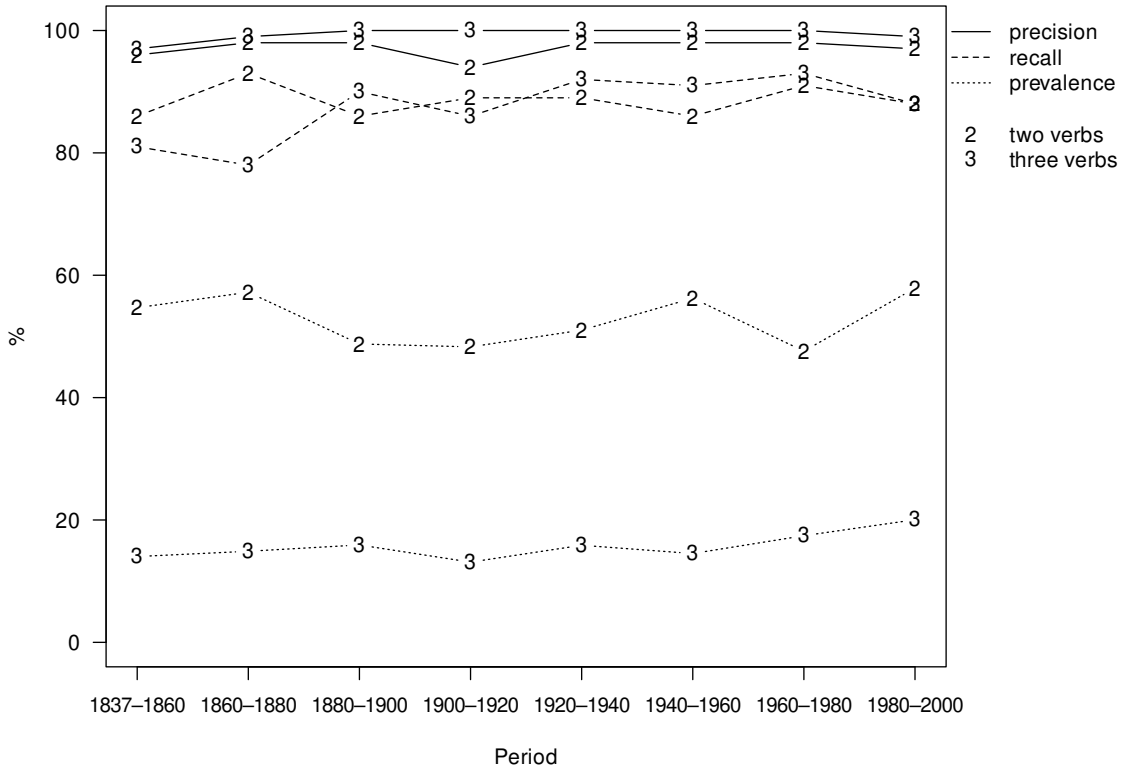

Figure 6: Result of the evaluation of our extracted two- and three-verb chains against manually annotated data. "Prevalence" refers to the prevelance in the evaluation data.

requirement 6, and one or more additional participles or infinitives. Prefiltering removed 44% of sentences in the two-verb selection and 73% in the three-verb selection, thus greatly reducing our annotation effort. As the prevalence numbers in Table 3 show, one out of two inspected items contained a two-verb chain, but only one out of six inspected items a three-verb chain, even after filtering. A systematic manual inspection of the automatically extracted data is a necessary step constructing a dataset for quantitative analysis, but it constitutes a non-trivial effort.

The results of our evaluation are given in Table 3 and Figure 6. Both two-verb and three-verb chains are recognized with very high precision, 97% and above 99%, respectively. Recall is also good,

we estimate we find just over 88% of the two-verb chains and just over 86% of the three-verb ones. A relevant question in diachronic research is whether this quality is constant over time. As the graph in Figure 6 shows, recall and precision are mostly stable over time, or at least do not show a clear trend. The exception is recall of the three-verb chains, which falls somewhat as we move towards older material.[13]

Further inspection of the recall data also shows other systematic tendencies. For instance, recall drops unsurprisingly as the distance between the verbs in the chain increases, for both chain lengths. Chains that involve coordination[14] are also an expected problem point, as our extraction method does not handle coordinated structures at all. There are more cases of coordinated chains in the three-verb data than in the two-verb data (37 and 20 cases, respectively) because of the extra conjoinable level in the former. Most, but far from all, of the coordinated chains are missed, and the correctly retrieved cases are accidental.

In the evaluation results in Table 3 and Figure 6, prevalence refers to the proportion of two- or three-verb chains in the annotated data sets. For the two-verb data, we see variation around the average 53% over time, but there is no clear trend. On the other hand, the three-verb data shows a slight increase of the three-verb chains over time.[15] We can project these figures to the whole data, by correcting for the bias introduced by our selection of the data to be annotated. We then see that we estimate that almost 30% of the finite verbs are part of a two-verb chain, and just of 4% are part of a three-verb chain. The picture we get from the automatically extracted chains in Table 2 aligns well with the estimate from manual annotation.

Finally, to get a sense of the downstream effect of the meta-annotations added in the parsing process, we reparsed our recall testset without meta-annotations and re-extract verb chains.[16] Recall drops to 84.2% (two verbs) and 83.4% (three verbs). As can be expected, the differences occur in the earlier periods. Meta-annotation appears to be beneficial to the quality of the final database.

6. A case study with the verb construction data

To illustrate one type of study facilitated by the database, we will look at verb order in three-verb constructions. We zoom more specifically in on the type below, in which a modal v_1 has scope over a perfect auxiliary v_2. Augustinus (2015, 129) gives three attested orders for such constructions in present-day Dutch: a dominant ascending order v_1–v_2–v_3 and two mixed orders v_3–v_1–v_2 and v_1–v_3–v_2, of which the latter is relatively infrequent. Example (8) is attested in the corpus, the two alternatives (9)–(10) are constructed.

(8) *Waterloo! Hasselt! Leuven!* *zeg* *mij,* *wat* *ik* *er* ***zou***$_1$ ***hebben***$_2$ ***uitgevoerd***$_3$*[!]*
tell me what I there would have done
'Waterloo! Hasselt! Leuven! Tell me what I would have accomplished there[!]'

(9) wat ik er **zou**$_1$ **uitgevoerd**$_3$ **hebben**$_2$ ⟨constructed⟩

13. We ran a series of simple logistic regressions with time (in years) as the independent variable to quickly confirm the visual impression. Modelling the precision data, the dependent variable was the correct classification of a extracted chain; modelling the recall data, the dependent variable is whether the manually labelled chain was also extracted. Of the four regressions, only recall of three-verb chains showed a significant effect of time (N=800 for each model, Wald test with $\alpha = 0.05$). The model predicts an increase in recall of 8.5%pts over the central 100-year period of 1860–1960.
14. In our manual annotation, we annotated chains that involve coordination as well as sharing of part of the chain between conjuncts by only considering the left-most branch such a structure as an instance of a verb chain. This ensures a straightforward, consistent, and unique annotation target.
15. Similar to precision and recall (footnote 13), we ran simple logistic regressions with time as dependent variable. The indepedent variable was whether the finite verb in the annotation data was at the start of a two-/three-verb chain. Time was not significant (Wald test, $\alpha = 0.05$) for the two-verb chains (N=1515), but it was for the three-verb chains (N=5015), with the model predicting just under 3%pts increase over the central 100 years.
16. We thank an anonymous reviewer for this suggestion.

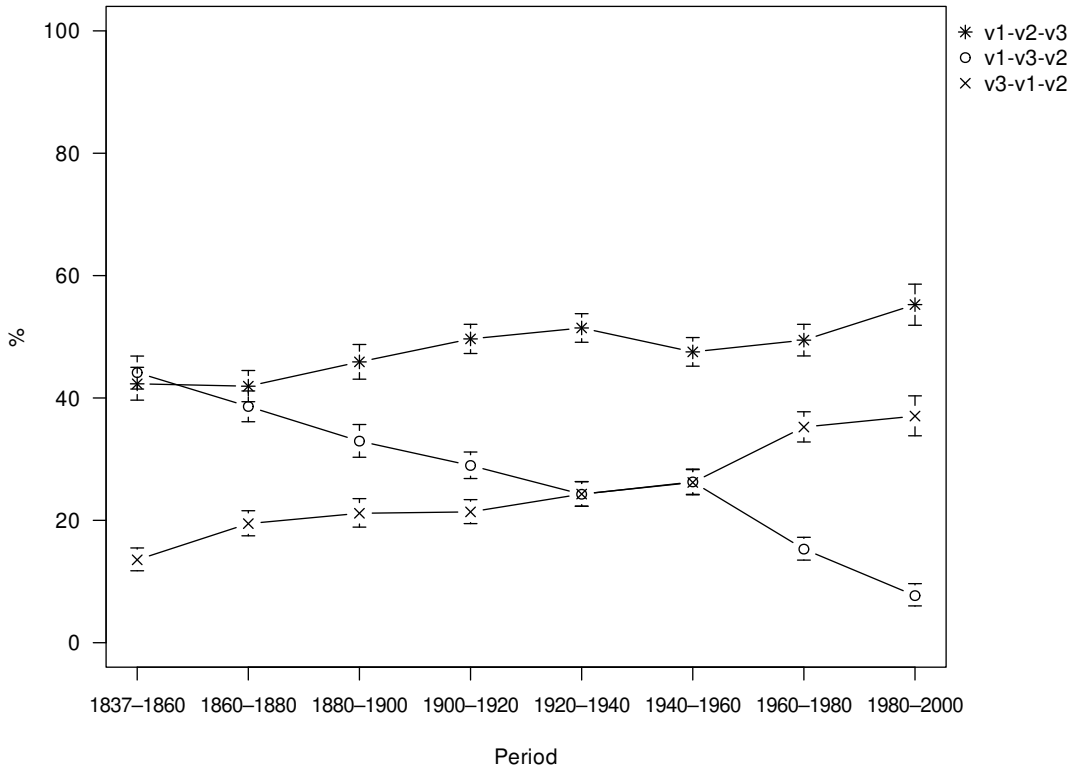

Figure 7: The order of three-verb combinations where v_2 is the infinitival perfect auxiliary *hebben* 'have', in subordinate clauses. The confidence intervals are at the 99.375% level (Bonferroni correction, familywise 95% confidence interval for 8 comparisons)

(10) wat ik er **uitgevoerd**$_3$ **zou**$_1$ **hebben**$_2$ ⟨constructed⟩

To study verb order in this context, we select from our dataset all examples of three-verb chains that meet the following requirements: the finite verb occurs in a subordinate clause, the second verb is the infinitival of *hebben* 'have', the verbs are directly next to each other, and the third verb does not have a seperated prefix.[17] The resulting selection consists of 22k5 examples. The dominant order is the rightward increasing order v_1–v_2–v_3, with almost 11k cases.

When we consider the data over time (Figure 7) we see that the increasing order is the dominant one for all but the first 40 years of the dataset, when it tied for first place with v_1–v_3–v_2, as in 9, with the order of the perfect auxiliary and the past participle flipped. Over time, this once frequent order steadily looses ground (goes from 44% to 8%) to the variant with an initial participle v_3–v_1–v_2, as in 10 (from 14% to 37%), and to v_1–v_2–v_3, which becomes more firmly established as the default (from 42% to 55%).

17. Note that these constraints exclude some word order variation to keep the case simple. For instance, the verb in example (8) has a seperable prefix *uit-*, that can also occur leftward of its stem: *uit zou hebben gevoerd*, etc. Such examples will not show up in the selection.

The finite verb v_1 is a form of *zullen* 'shall' in 75% of the cases, with forms of *kunnen* 'can', *moeten* 'must', *mogen* 'may', *willen* 'want' filling up the remaining 25%. Interestingly, there are clear diachronic trends here, too: *zullen*'s share falls from 77% to 68%, and *mogen* and *willen* fall from 6% to below 2%. The winners are *kunnen*, from 5% to 10% and *moeten*, from 6% to 19%. An explanation for these trends, and if they can be correlated with the changes in word order preferences will have to await further study.

7. Conclusion

In this article, we described the creation of the Gothenburg Parsed Version of Dutch C-CLAMP and the extraction of the Dutch Verb Construction Database from this parsed corpus. Automatic parsing of Dutch C-CLAMP with Alpino posed a substantial challenge, as the material is partly off-domain for the parser with respect to genre, region, and time. We addressed this challenge primarily by bringing the text to the parser, adopting the approach developed by van Cranenburgh and van Noord (2022), which adds meta-annotations to the text so that changes in spelling and other orthographic conventions are smoothed out. To a lesser extent, we also adapted the parser itself to better handle the specific characteristics of the material. The iterative application of these adjustments improved Alpino's coverage from 78% to 84%, demonstrating that this combined strategy is an effective way of dealing with historical, out-of-domain data.

Beyond improving parsing quality, corpus processing also offered valuable insight into the language of the corpus and opens up avenues for further research. The spelling edits made to adjust the corpus to the parser offer a window onto how successive spelling reforms were integrated into the language. We observe a steady decrease in the number of edits over time, pointing to a gradual modernizing in spelling which reaches completion after the Second World War. Closer to our project focus on verb constructions, parser coverage improves over time, which seems to relate with an overall tendency towards shorter sentences, fewer finite verbs per sentence, and a lower proportion of finite verbs relative to all verbs. Taken together, these trends suggest an ongoing process of simplification in verbal syntax. In addition to documenting the challenges involved in parsing Dutch C-CLAMP, we demonstrated how the resulting parsed corpus supports large-scale investigations of syntactic structure, more specifically, of changing verb constructions in the recent history of Dutch. We defined the search queries used to extract all verb constructions from the corpus into the Dutch Verb Construction Database, and showed, on the basis of a manual evaluation of two and three-verb chains, that the automatic extraction achieves good precision and recall. Finally, we presented a small case study illustrates how the database supports the diachronic study of changing word order in three-verb clusters.

The resources presented in this paper and associated code can be downloaded from the following locations:

- The Parsed Gothenburg Edition of Dutch C-CLAMP can be downloaded from the Language resource catalogue of the Dutch Language Institute `http://hdl.handle.net/10032/tm-a3-e3`. Access to the materials requires an agreement with the National Library of the Netherlands. Instructions are available at the download site.
- The code to add the meta-annotation used in the corpus presented here can be downloaded from `https://github.com/gerlofbouma/gbg-edn-cclamp-metaannotate`
- The Dutch Verb Construction Database can be downloaded from `https://doi.org/10.5281/zenodo.20058111`.
- XQuery code to extract verb chains from Alpino annotations is available from `https://github.com/gerlofbouma/dvcdb-extraction`.

Acknowledgments

We thank three anonymous reviewers for their feedback and suggestions, which have improved the paper. The research presented in this article is part of the project *Verb Constructions in the Recent History of Dutch. A Constructional Network Perspective* funded by the Swedish Research Council (nr. 2023-0087). The first author has also been supported by Språkbanken – jointly funded by its 10 partner institutions and the Swedish Research Council (2025–2028; project id 2023-00161).

References

Augustinus, Liesbeth (2015), *Complement raising and cluster formation in Dutch*, PhD thesis, KU Leuven. LOT Dissertation Series 413, https://www.lotpublications.nl/complement-raising-and-cluster-formation-in-dutch.

BaseX GmbH (2023), BaseX (version 10.5). [Software] https:://www.basex.org.

Bloem, Jelke (2021), *Processing verb clusters*, PhD thesis, Universiteit van Amsterdam. LOT Dissertation Series 586, https://www.lotpublications.nl/processing-verb-clusters.

Burchell, Laurie, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield (2023), An open dataset and model for language identification, *in* Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Toronto, Canada, pp. 865–879. http://doi.org/10.18653/v1/2023.acl-short.75.

Byloo, Pieter and Jan Nuyts (2014), Meaning change in the Dutch core modals: (Inter)subjectification in a grammatical paradigm, *Acta Linguistica Hafniensia* **46** (1), pp. 85–116, Routledge. http://doi.org/10.1080/03740463.2014.955978.

Coupé, Griet (2015), *Syntactic extension - The historical development of Dutch verb clusters*, PhD thesis, Radboud University. LOT Dissertation Series 395, https://www.lotpublications.nl/syntactic-extension-the-historical-development-of-dutch-verb-clusters.

Coussé, Evie (2008), *Motivaties voor volgordevariatie : een diachrone studie van werkwoordvolgorde in het Nederlands*, PhD thesis, Universiteit Gent.

Coussé, Evie (2014), Lexical expansion in the HAVE and BE perfect in Dutch: A constructionist prototype account, *Diachronica* **31** (2), pp. 159–191, John Benjamins. http://doi.org/10.1075/dia.31.2.01cou.

Coussé, Evie and Gerlof Bouma (2022), Semantic scope restrictions in complex verb constructions in Dutch, *Linguistics* **60** (1), pp. 123–176. http://doi.org/10.1515/ling-2021-0172.

de Kok, Daniël, Jianqiang Ma, and Gertjan van Noord (2009), A generalized method for iterative error mining in parsing results, *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, Association for Computational Linguistics, Suntec, Singapore, pp. 71–79. http://www.aclweb.org/anthology/W/W09/W09-2609.

de Kok, Daniël and Gertjan van Noord (2017), Mining for parsing failures, *in* Wieling, Martijn, Martin Kroon, Gertjan van Noord, and Gosse Bouma, editors, *From semantics to dialectometry. Festschrift for John Nerbonne*, number 32 in *Tributes*, College Publications, pp. 81–90. http://www.let.rug.nl/vannoord/30years/festschrift/.

De Troij, Robbert and Freek Van de Velde (2020), Beyond mere text frequency: Assessing subtle grammaticalization by fifferent quantitative measures. a case study on the Dutch soort construction, *Languages* **5** (4), 55. http://doi.org/10.3390/languages5040055.

Eckhoff, Hanne, Kristin Bech, Gerlof Bouma, Kristine Eide, Dag Haug, Odd Einar Haugen, and Marius Jøhndal (2018), The PROIEL treebank family: a standard for early attestations of Indo-European languages, *Language Resources and Evaluation* **52**, pp. 29–65. http://doi.org/10.1007/s10579-017-9388-5.

Haesereyn, Walter, editor (1997), *Algemene Nederlandse Spraakkunst 2*, Martinus Nijhoff, Groningen. Consulted as e-ANS v1.3, https://e-ans.ivdnt.org/data/archief/ans2/e-ans/body.htm.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz (1993), Building a large annotated corpus of English: The Penn Treebank, *Computational Linguistics* **19** (2), pp. 313–330, MIT Press. https://aclanthology.org/J93-2004/.

Narayan, Shashi and Claire Gardent (2012), Error mining with suspicion trees: Seeing the forest for the trees, *in* Kay, Martin and Christian Boitet, editors, *Proceedings of COLING 2012*, The COLING 2012 Organizing Committee, Mumbai, India, pp. 2011–2026. https://aclanthology.org/C12-1123/.

Nijs, Julie, Freek Van de Velde, and Hubert Cuyckens (2025), Is word order responsive to morphology? Disentangling cause and effect in morphosyntactic change in five Western European languages, *Entropy* **27** (1), 53. http://doi.org/10.3390/e27010053.

Nunn, Anneke (1998), *Dutch orthography*, PhD thesis, Katholieke Universiteit Nijmegen. LOT Dissertation Series 6, https://www.lotpublications.nl/dutch-orthography.

Nuyts, Jan, Wim Caers, and Henri-Joseph Goelen (2022), The Dutch modals, a paradigm?, *Paradigms regained*, Language Science Press, pp. 245–265. http://doi.org/10.5281/zenodo.5675853.

Pettersson, Eva, Beáta Megyesi, and Joakim Nivre (2012), Parsing the past - identification of verb constructions in historical text, *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, Association for Computational Linguistics, pp. 65–74. https://aclanthology.org/W12-1010/.

Piersoul, Jozefien and Freek Van de Velde (2023), Men use more complex language than women, but the difference has decreased over time: a study on 120 years of written Dutch, *Linguistics* **61** (3), pp. 725–747. http://doi.org/10.1515/ling-2021-0022.

Piersoul, Jozefien, Robbert De Troij, and Freek Van de Velde (2021), 150 years of written Dutch, *Nederlandse Taalkunde* **26** (3), pp. 339–362, Amsterdam University Press. http://doi.org/10.5117/NEDTAA2021.3.002.PIER.

Piotrowski, Michael (2012), *Natural Language Processing for Historical Texts*, Synthesis Lectures on Human Language Technologies, Springer. http://doi.org/10.1007/978-3-031-02146-6.

Rudnicka, Karolina (2018), Variation of sentence length across time and genre, *in* Whitt, Richard Jason, editor, *Diachronic Corpora, Genre, and Language Change*, number 85 in *Studies in Corpus Linguistics*, John Benjamins, Amsterdam, pp. 220–240. http://doi.org/10.1075/scl.85.10rud.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger (2002), Multiword expressions: A pain in the neck for NLP, *in* Gelbukh, Alexander, editor, *Computational Linguistics and Intelligent Text Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–15. http://doi.org/10.1007/3-540-45715-1_1.

Sagot, Benoît and Éric de la Clergerie (2006), Error mining in parsing results, *in* Calzolari, Nicoletta, Claire Cardie, and Pierre Isabelle, editors, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Sydney, Australia, pp. 329–336. https://aclanthology.org/P06-1042/.

Taylor, Ann (2020), Treebanks in historical syntax, *Annual Review of Linguistics* **6** (Volume 6, 2020), pp. 195–212, Annual Reviews. http://doi.org/10.1146/annurev-linguistics-011619-030515.

van Cranenburgh, Andreas (2022a), A dataset of Dutch novels 1800-2000. https://lab.kb.nl/about-us/blog/dataset-dutch-novels-1800-2000.

van Cranenburgh, Andreas (2022b), Machine learning canonicity in Dutch novels 1800-2000. https://lab.kb.nl/about-us/blog/machine-learning-canonicity-dutch-novels-1800-2000.

van Cranenburgh, Andreas and Gertjan van Noord (2022), Openboek: A corpus of literary coreference and entities with an exploration of historical spelling normalization, *Computational Linguistics in the Netherlands Journal* **12**, pp. 235–251. https://clinjournal.org/clinj/article/view/157.

Van de Velde, Freek, Jozefien Piersoul, and Isabeau De Smet (2020), De wervelkolom van taalverandering, *Nederlandse Taalkunde* **25** (2-3), pp. 371–385, Amsterdam University Press. http://doi.org/10.5117/NEDTAA2020.2-3.020.VAND.

van Noord, Gertjan (2004), Error mining for wide-coverage grammar engineering, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, pp. 446–453. http://doi.org/10.3115/1218955.1219012.

van Noord, Gertjan (2006), At Last Parsing Is Now Operational, *in* Mertens, Piet, Cédrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pp. 20–42.

van Noord, Gertjan (2009), Learning efficient parsing, *in* Lascarides, Alex, Claire Gardent, and Joakim Nivre, editors, *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, Association for Computational Linguistics, Athens, Greece, pp. 817–825. https://aclanthology.org/E09-1093/.

van Noord, Gertjan, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste (2013), Large scale syntactic annotation of written Dutch: Lassy, *in* Spyns, Peter and Jan Odijk, editors, *Essential speech and language technology for Dutch: Results by the STEVIN programme*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 147–164. http://doi.org/10.1007/978-3-642-30910-6_9.

Appendix A. Verb constructions in the database