

Assessing the impact of automatic dependency annotation on the measurement of phraseological complexity in L2 Dutch

Abstract

The extraction of phraseological units operationalized in phraseological complexity measures (Paquot, 2019) relies on automatic dependency annotations, yet the suitability of annotation tools for learner language is often overlooked. In the present article, two Dutch dependency parsers, Alpino (van Noord, 2006) and Frog (van den Bosch et al., 2007), are evaluated for their performance in automatically annotating three types of dependency relations (verb + direct object, adjectival modifier, and adverbial modifier relations) across three proficiency levels of L2 Dutch. These observations then serve as the basis for an investigation into the impact of automatic dependency annotation on phraseological sophistication measures. Results indicate that both learner proficiency and the type of dependency relation function as moderating factors in parser performance. Phraseological complexity measures computed on the basis of both automatic and manual dependency annotations demonstrate moderate to high correlations, reflecting a moderate to low impact of automatic annotation on subsequent analyses.

Keywords: Dependency parsing; L2 Dutch; phraseological complexity; proficiency

1. Introduction

The automatic annotation of dependency relations offers undeniable advantages for large-scale analyses of learner language production. Where learner corpora were previously analyzed primarily on the basis of lexical evidence, researchers now have access to Natural Language Processing (NLP) resources capable of providing multiple layers of linguistic annotation automatically with relatively high reported levels of accuracy for

learner productions. This interface between NLP and Learner Corpus Research (LCR) has been particularly advantageous to L2 phraseology research, which has traditionally relied on lexical information for the definition and extraction of phraseological units. Several implementations of NLP methods have been attested in L2 phraseology research to capture various lexical co-occurrence phenomena (Bestgen, 2017; Heid, 2008). Such methods, for example n-gram approaches to phraseological unit extraction, make use of lexical frequency evidence available in the learner production and typically require no additional linguistic annotation. Dependency parsers provide a layer of syntactic annotation to (learner) corpus data, allowing for the extraction of lexically and syntactically defined phraseological units. One such implementation is the use of dependency annotations in the operationalization and measurement of phraseological complexity (Paquot, 2018, 2019; Rubin, Housen, & Paquot, forthcoming). In this approach, dependency relations serve as the basis for the extraction of “relational co-occurrences”, i.e. two lexical items connected in an underlying structural relationship (e.g. verb-argument and modifier relations).

Phraseological complexity is defined within the broader construct of linguistic complexity, which is considered to be one of the central components of learner proficiency alongside accuracy and fluency (Housen & Kuiken, 2009). Linguistic complexity has been operationalized and measured across various language domains (e.g. syntax, lexis, morphology), and has proven to be a useful indicator of learner performance and development (Housen & Kuiken, 2009; Norris & Ortega, 2009). Following Ortega’s (2003) definition of linguistic complexity, diversity and sophistication have been proposed as the two core dimensions of phraseological complexity (Paquot 2018, 2019),

respectively measuring the degree of variation and the degree of sophistication of the phraseological units produced by a learner.

Phraseological complexity measures have been computed for three target dependencies for L2 English (Paquot, 2018, 2019): verb + direct object (dobj), adjectival modifier (amod), and adverbial modifier (advmod) dependencies. Modifiers have been demonstrated to increase in sophistication across proficiency levels (Durrant & Schmitt, 2009; Granger & Bestgen, 2014), and are therefore prominent in phraseology and learner corpus research studies. Verb-argument relations have played a central role in corpus-based SLA research, and can present specific challenges for automatic dependency annotation: (1) these often manifest in long-distance dependencies, and (2) these also exhibit a stronger reliance on morphosyntactic agreement. The inclusion of structural information in the measures of phraseological diversity and sophistication put forth by Paquot (2018, 2019) reflects a relatively recent shift in corpus-based L2 phraseology research away from purely lexically-defined units (Granger & Paquot, 2008; Gries, 2008). This particular approach to the quantification of complexity across structurally-defined phraseological units relies on accurate dependency annotations for the extraction of the target phraseological units. Inaccurate dependency annotation would therefore call into question the validity of these dependency-based complexity measures.

Whether these automatic annotation tools that have been developed and trained on native speaker language data are applicable for the annotation of learner language has therefore been a source of increased interest for both NLP and LCR (Meurers, 2009; Meurers & Dickinson, 2017). Large-scale learner corpora are only as valuable as the information that can be extracted from them, and automatic linguistic annotation has vastly expanded this potential (Díaz-Negrillo, Meurers, Valera, & Wunsch, 2010). Still,

these advantages cannot be accepted at face value alone, and it is the duty of the responsible (learner corpus) researcher to rigorously establish the validity of these methods for their specific purposes. The central aims of the present investigation are therefore to examine the impact of automatic dependency annotation on the measurement of phraseological complexity in learner productions, as well as to propose a suitable approach for assessing this impact.

2. Dependency parsing learner language

Despite the increasing use of automatic linguistic annotation for learner corpus data, the suitability of automatic dependency parsers for learner language annotation still needs to be empirically demonstrated. To that end, several analyses have evaluated the performance of dependency parsers on learner language input, reporting overall accuracy values above 80% (for L2 German: Krivanek & Meurers, 2012; Ott & Ziai, 2010; for L2 English: Huang, Murakami, Alexopoulou, & Korhonen, 2018). These findings have highlighted the role of the approach to parsing (e.g. rule-based, data-driven) as well as the variability of parser performance across dependency relations.

Attempts to accurately annotate learner language with dependency relations are faced with unique challenges stemming from the variability and idiosyncrasies characteristic of learner language. Moreover, variability in learner language is not consistent across all stages of learner development, and has been attributed to patterns of second language acquisition (Sharwood Smith & Truscott, 2005). Though research into learner language dependency annotation has produced promising results, the dimension of learner proficiency has not featured prominently. As L2 complexity research is primarily interested in the relationship between the complexity of the learner production

and learner proficiency, learner corpora investigated in these studies often include learner productions across a range of proficiency levels. Non-native-like learner language has been identified as a major source of parsing error (Huang et al., 2018), and it therefore stands to reason that less native-like language presents an increased challenge for automatic parsing. If parsing is systematically less accurate at the lower proficiency levels, this can introduce bias into subsequent analyses that factor in learner proficiency. By considering proficiency as a variable in the performance of automatic dependency annotation for learner language, the present analysis attempts to better align the aims of LCR with the evaluation of available NLP resources, in particular for L2 Dutch.

3. Research Setup

3.1 Research questions and objectives

The approach taken here to assess the impact of automatic dependency annotation on phraseological complexity measures is presented across two studies carried out to investigate: (1) the performance of two Dutch-language dependency parsers, Alpino (van Noord, 2006) and Frog (van den Bosch, Busser, Canisius, & Daelemans, 2007), on L2 Dutch texts, across CEFR exam levels and dependency relations and (2) the impact of the observed performance on the calculation of phraseological sophistication measures computed on the basis of the annotated dependency labels. As no previous study has (to the author's knowledge) carried out a systematic investigation of dependency parsing tools for L2 Dutch data, the first study constitutes an important contribution to research at the interface of NLP and LCR, in addition to serving as a foundation for the second study. The second study presented here takes on the task of investigating the real impact of the use of automatic annotation in LCR studies, in particular for the measurement of

phraseological sophistication. The two studies together correspond to the step-by-step procedure carried out in an analysis of phraseological complexity: from the identification of suitable annotation tools to the extraction of the target phraseological units with which measures of phraseological complexity are computed. These analyses are guided by the following research questions:

1. How accurate are the annotations produced by the parsers when provided with L2 Dutch input across CEFR exam levels and for a range of dependency relations?
2. How do the automatic dependency annotations impact measures of phraseological sophistication across CEFR exam levels and dependency relations?

In light of previous findings concerning automatic dependency annotation of learner language (e.g. Krivanek & Meurers, 2012; Ott & Ziai, 2010), some variation in the performance across dependency relations is expected. Additionally, accuracy is expected to increase with proficiency level, as higher proficiency learner language is thought to more closely resemble the native training data and the native annotation standards of the parsers. Given the hypotheses for parsing accuracy, the impact of the automatic dependency annotations on the measures of phraseological sophistication is expected to decrease at the higher proficiency levels.

3.2 Dutch learner data

The Dutch learner texts were sampled from an L2 Dutch corpus compiled from written extracts of the *Certificaat Nederlands als Vreemde Taal* (CNaVT; Certificate of Dutch as a Foreign Language) exams, administered by the *Centrum voor Taal en Onderwijs* (CTO; Center for Language and Education) at the *Katholieke Universiteit Leuven* in Belgium. The corpus includes written productions from exams across the B1, B2, and C1 levels of

the Common European Framework of Reference for Languages (CEFR; Council of Europe, 2001). The L1 backgrounds of the learners vary (though the majority consists of L1 French speakers from Belgium), as well as the learning contexts (L2/L3). The texts are written in response to written and/or oral prompts, and are primarily informative/argumentative. The texts are originally handwritten documents, which have been digitized (each text by one transcriber) as part of the corpus compilation process. The texts have been subjected to minimal pre-processing, including the removal of markup, titles, and greetings, and have not been corrected for orthographic (or other) errors. In this way, the variation and non-native-like features observed in the learner language, especially in relation to learner proficiency, factor into the evaluation carried out.

3.3 Sampling

Texts were randomly sampled whole from the corpus (as opposed to sampling sentences from texts at random), so that it was still possible to consider the degree of inter-speaker variation in parsing performance. From each of the three CNaVT exam levels included in the corpus, corresponding to the B1, B2, and C1 CEFR levels, 10 texts were sampled at random. In order to keep the manual dependency annotation task manageable, each text was reduced to 10 sentences. Because of differences in sentence segmentation between the two parsers, the final sample (see Table 1) included 307 sentences from 30 texts. The CEFR level of the CNaVT exam is taken here as a proxy for learner proficiency. Although not an ideal indicator of learner proficiency (cf. Carlsen, 2012), the prompts have been designed to elicit language at the target CEFR level. There is also sufficient variation in

the exam scores across the three levels, indicating variety in the proficiency levels of the learners.

Table 1. Sample size in number of texts, sentences, and tokens across proficiency level

	Texts	Sentences	Tokens
Overall	30	307	4,900
B1	10	100	1,582
B2	10	102	1,663
C1	10	105	1,655

4. Dutch Dependency parsers

The CNaVT texts were annotated using the only two dependency parsers developed for the Dutch language, Alpino (van Noord, 2006) and Frog (van den Bosch et al., 2007). Though Alpino has a higher reported accuracy than Frog on their respective test data, the parsers take different computational approaches to dependency annotation in Dutch, and it is not immediately apparent which approach is better suited to handling a greater degree of learner variability in addition to several different dependency relations. Both parsers are trained on the Alpino treebank (van der Beek, Bouma, Malouf, & van Noord, 2002), a collection of more than 7,000 sentences extracted from newspapers that have been manually annotated with dependency relations. The parsers share a dependency label set, and the output of the tools can therefore be compared with (relative) ease. The shared dependency label set can be consulted in Appendix I.

4.1 Alpino

The Alpino parser (van Noord, 2006) takes a hybrid approach to automatic annotation of linguistic information in Dutch, combining both a rule-driven approach, i.e. a head-driven phrase structure grammar (HPSG) and lexicon, and a data-driven disambiguation model. The Alpino HPSG consists of around 600 specific rules, and the lexicon around 100,000 entries (van Noord, 2006). While the rule-based grammar and lexicon propose linguistically-motivated dependency structures, the data-driven disambiguation component aids in selecting the most likely structure. Evaluated on the basis of the Alpino treebank, in addition to two additional test sets, i.e. questions from the CLEF question answering competition and sentences selected from the Twente News Corpus (Ordelman, de Jong, van Hessen, & Hondorp, 2007), the tool reaches an accuracy of between 87.9% and 90.9%, measured in concept accuracy (CA). CA is measured in van Noord (2006: 3) as follows:

$$CA = 1 - \frac{\sum_i D_f^i}{\max(\sum_i D_g^i, \sum_i D_p^i)}$$

Where:

D_p^i is the number of relations produced by the parser for sentence i ;

D_g^i is the number of relations in the treebank parse for sentence i ;

and D_f^i is the number of incorrect or missing parses produced by the parser for sentence i

The parser is available as a command-line tool and web-based service, and can process a text through the following stages: tokenization, lemmatization, POS-tagging, and dependency parsing. Output is provided in XML format, and allows for fully-specified dependency structures with, for example, multiple heads attached to a single dependent. XQuery scripts for querying dependency relations from the XML trees are incorporated

in the tool, and can be implemented through the command-line options (returning dependency relations in a more human-readable format: so-called dependency triples); these queries are outlined in detail by Bouma and Kloosterman (2007).

4.2 Frog

Frog, originally TADPOLE (van den Bosch et al., 2007), is a pipeline of Dutch NLP modules operating a memory-based learning engine, IGTREE (Daelemans, van den Bosch, & Weijters, 1997), common to all three core components of the tool: POS-tagger, morphological analyzer, and dependency parser. In this memory-based learning approach, the dependency relation information derived from the annotated training set is compressed into a decision-tree structure, which is then used to make predictions about novel dependency relations. The resulting IGTREE is therefore “able to reproduce the classifications of all examples in the original example database perfectly” (van den Bosch et al., 2007: 194). The Frog dependency parsing module reaches an accuracy of 67.3% for labeled dependencies based on a manually-annotated training set, and an accuracy of 74.3% with automatically-annotated data added to the training set (annotated with the Alpino parser), measured in terms of Labeled Attachment Score (LAS). Attachment scores, both labeled and unlabeled, are a common metric of dependency parsing accuracy, computed as the total number of correctly (labeled/unlabeled) attached dependency relations over the total number of identified dependency relations.

Frog is available as a command-line tool, and performs tokenization, lemmatization, POS-tagging, and dependency parsing. Output is provided in either XML or tab-delimited format. A token in this framework can only be attached to one head. When comparing reported parsing accuracy across Alpino and Frog for standard language

input, Alpino emerges as a clear front-runner. What is not entirely certain, however, is whether either of the approaches implemented in the tools is better suited to adapt to the challenges associated with learner language annotation across dependency relations and a range of proficiency levels.

4.3 Compatibility of parser output

A common difficulty in comparing the performance of different NLP tools is the lack of standardization across various aspects of the tools (Boyd & Meurers, 2008; Meurers & Wunsch, 2010; Tsarfaty, Nivre, & Andersson, 2011), including (but not limited to) tagsets, output formats, evaluation metrics, and, specific to dependency parsing, dependency grammar frameworks (de Marneffe & Nivre, 2019). As discussed above in Section 4, the parsers share a common set of dependency labels (see Appendix I), and implement similar annotation schemes. For example, while different dependency grammar frameworks deal with coordination differently, both Alpino and Frog annotate the coordinator as the head of its conjuncts, with exceptions in some more complex cases. Figure 1 presents the dependency tree for the following example of coordination (extracted from a C1 exam) in this annotation framework:

(1)	<i>tussen</i>	<i>de</i>	<i>presentator</i>	<i>en</i>	<i>een</i>	<i>luisteraar</i>
	between	DEF.SG	presenter	and	INDF.SG	listener
	‘between the presenter and a listener’					

The tools differ mainly in the explicitness of the dependency structure and in the format of the output. It was therefore necessary to implement a systematic conversion between the output of the parsers. The conversion was carried out by means of an in-house R script (R Core Team, 2017).

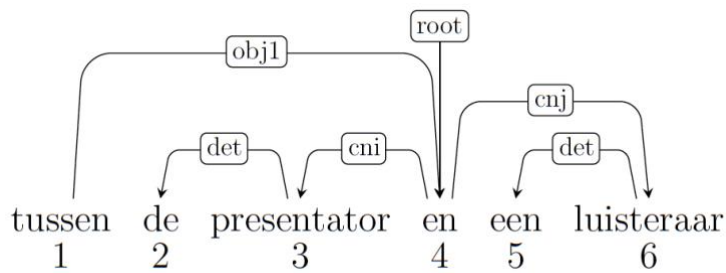


Figure 1. Example of dependency tree for word coordination

The first discrepancy between the output of the parsers is the inconsistency in the number of heads allowed to attach to a dependent. As Alpino allows for more explicit structures with multiple heads per dependent, the parser could theoretically (and does systematically) identify more dependency relations than Frog. As these missing dependencies are the result of design differences as opposed to errors on the part of Frog, the Alpino dependency structure was reduced to one head for each dependent. A concrete example is the treatment of passive constructions in Alpino as compared with Frog. Figures 2 and 3 present the dependency structure annotated by both Alpino and Frog, respectively, for the same passive structure in the following example adapted from a C1 exam:

(2)	<i>Mens-en</i>	<i>word-en</i>	<i>in</i>	<i>5</i>	<i>categorie-ën</i>	<i>ge-zet</i>
	Person-PL	PASS.AUX-3PL	in	5	category-PL	PTCP-place
	‘People are placed in 5 categories.’					

In this passive construction, “mens-en” (‘people’) is attached in a subject role to the passive auxiliary verb “word-en” (‘are’), and additionally to the main verb “zetten” (‘place’) in an object role by Alpino. Conversely, Frog does not represent the object role of “mens-en” (‘people’) in the main verb phrase, as only one head is allowed and Frog

therefore assigns the higher-order verb in the structure as the head: in this case, the auxiliary verb is higher up in the tree. To ensure compatibility between the two parsers, the verb-object relation between the main verb “zetten” and the noun “mensen” is removed from the Alpino parse tree.

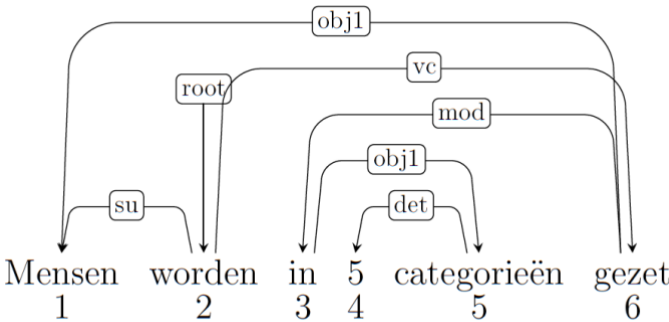


Figure 2. Example of dependency tree for passive sentence (Alpino annotation)

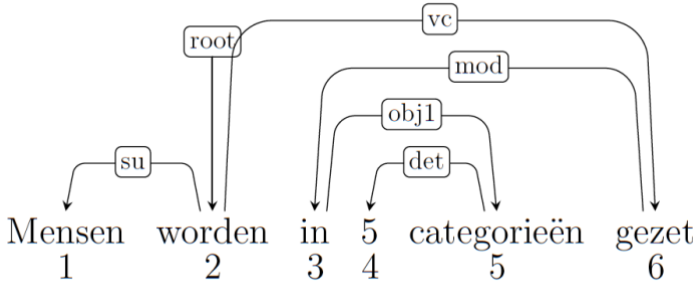


Figure 3. Example of dependency tree for passive sentence (Frog annotation)

Because the Alpino parses were returned in the dependency triples format mentioned above in Section 4.3, and the Frog parses in tab-delimited format, it was necessary to unify the formatting between the two. The tab-delimited format provided by Frog lends itself well to comparison, with each row representing a token and each token assigned an index, head, and dependency label. The Alpino dependency triples were therefore

converted to tab-delimited format to mirror the format of the Frog parses. The final and most time-consuming component of the conversion was dealing with discrepancies in word segmentation. Word segmentation differences present difficulties for comparing the output of the two parsers, as the parsers will identify a different number of tokens per sentence, and these tokens will receive offset indices. It is crucial that the indices of tokens correspond between the two parsers, as these are the basis for evaluating whether the same head-dependent attachments were made by both tools. While both parsers handle multi-word unit identification reasonably well, based upon fixed word lists, learner Dutch often presents cases that pose a challenge for identification, and which are often treated differently by the two parsers. When one parser identified a multi-word unit that was represented by the other parser as a noun phrase, the noun phrase was collapsed into a multi-word unit, effectively removing any determiner and/or modifier relations dependent on the head noun. In this way, neither parser was faulted, and no additional dependencies needed to be postulated by the annotator. This last step was carried out manually for each sentence that demonstrated discrepancies in token number and/or indexation.

5. Gold standard

Dependency annotations deal purely with surface forms of language, specifying structural relations determined on the basis of, among other sources of information, morphosyntactic marking, distributional information, and argument structures (i.e. information evidenced in the production itself) (Dickinson & Ragheb, 2009; Ragheb & Dickinson, 2012). As learner productions consist of realizations that often do not correspond to morphosyntactically well-formed structures (according to the native

standard), a multi-layered approach to syntactic annotation of learner language has been proposed to best capture conflicting sources of evidence in the learner production (Lüdeling, Walter, Kroymann, & Adolphs, 2005; Meurers & Wunsch, 2010; Ragheb & Dickinson, 2012). This multi-layered annotation approach can lead to several competing target hypotheses for the underlying structural relations of the learner production (Lüdeling et al., 2005), and the questions therefore remain (1) how to best translate this multi-layered annotation into a single automatically annotated dependency layer and (2) whether this can be achieved for learner language with the same level of accuracy demonstrated for native language (see the contribution of Weiss and Meurers to the present issue for a more detailed overview of target hypotheses).

The approach to gold standard dependency annotation adopted here is guided primarily by the argument structures licensed by a head node as evidenced in the learner production, irrespective of morphosyntactic agreement phenomena, in combination with distributional evidence. Whereas argument structures do not deal with an internal analysis of the learner language, a distributional approach takes into consideration the “position where a token with particular properties (e.g., singular noun) is predicted to occur, on the (syntactic) basis of its surrounding tokens” (Ragheb & Dickinson, 2012: 969). The dependency relations annotated here for L2 Dutch follow the LASSY syntactic annotation guide (van Noord, Schuurman, & Bouma, 2011).

The dependency tree in Figure 4 presents the target hypothesis of the dependency structure for the following non-native-like example from a B1 CNaVT text:

(3) Voor de eerste probleem heeft hij
 For DEF.SG first problem AUX.have.3SG 3SG.M.NOM
 voorgesteld de straat vernieuw-t
 proposed.PTCP DEF.SG street renovate-3SG

‘For the first problem, he proposed the street renovates’

The dependency arcs and labels in the tree represent the structure assumed in the learner language on the basis of the arguments licensed by the head nodes and the distributional context. In the prepositional phrase “Voor de eerste probleem” (‘For the first problem’), the non-neuter definite article “de” attaches to the neuter noun “probleem”, corresponding to the open slot for an article licensed by the noun, despite the lack of morphosyntactic agreement between article and noun. The article is then followed by a modifier “eerste” (‘first’). Context aids in the selection of arguments: for example, the ordinal numeral “eerste” (‘first’) can also serve as an argument of the verbal root as well as select for the article “de” (‘the’), which would not conflict in terms of morphosyntactic agreement, unlike the proposed structure in Figure 4 (where “probleem” is the argument of the verbal root). However, taking into consideration the distributional evidence, the argument structure adopted in the tree in Figure 4 accounts best for all components realized in the learner production.

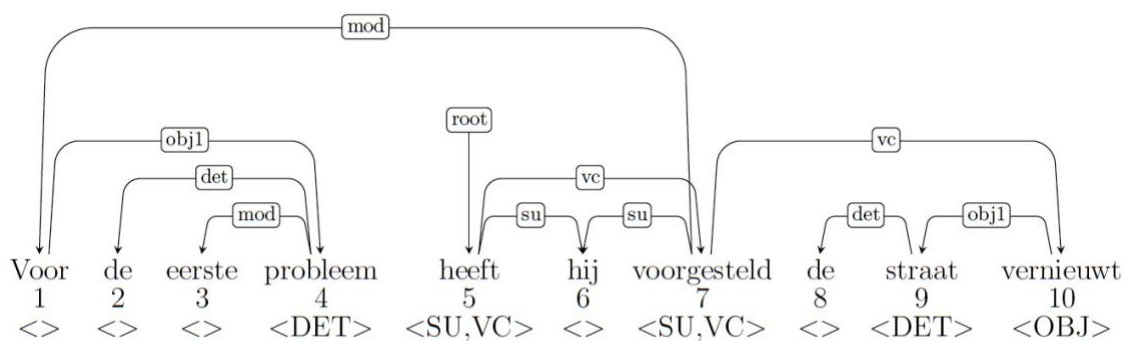


Figure 4. Dependency tree (with annotated argument structures) for the target hypothesis of Example 3

In the dependency framework adopted by Alpino and Frog, the auxiliary verb serves as the verbal root of the sentence, and main verbs are labeled as verbal complements of the auxiliary. Modifiers are attached to the main verb, and arguments are attached to the auxiliary verb (and additionally to the main verb by Alpino) (see Figure 4). The main verb “voorstellen” (surface form is the past participle “voorgesteld”; ‘propose’/‘proposed’) licenses the following syntactic slots in the selected context: a subject <SU> and a verbal complement <VC>. Two possibilities arise for the interpretation of the structure of the verbal complement, which is morphosyntactically ill-formed according to the native standard. In the learner production, the verbal head of the complement clause is finite “vernieuwt” (3SG.PRS, ‘renovates’), which licenses verbal arguments <SU> and <OBJ>, of which the former is not realized. In this interpretation, there is also an unrealized complementizer “dat” (‘that’) at the head of the verbal complement. Alternatively, the surface form of the verbal complement is more compatible with an infinitive complement headed by infinitive marker “te”, where the “te” is not realized and the verbal inflection of “vernieuwt” does not agree with the infinitive interpretation. Either way, the argument structure of “voorstellen” does not force an interpretation, still selecting “vernieuwt” as the head of the verbal complement. As a final note on the gold standard, annotation was carried out through the manual correction of the parser output (specifically Alpino output, as this is more explicit). This was determined to be the most time-effective solution, although previous investigations have found that this method can introduce some bias into the gold standard and inflate inter-annotator reliability measures (Huang et al., 2018).

5.1 Double coding the gold standard

A sample of 30 sentences was selected from the 307 manually annotated sentences (~10% of the total) for double coding. A graduate student¹ at the Vrije Universiteit Brussel was trained on the syntactic annotation guide used in the manual coding of the gold standard, and proceeded to carry out a manual annotation of the 30-sentence double coding sample. The double coded dependency annotations were only evaluated in terms of the dependencies identified by both annotators. Because the manual coding was carried out as explicitly as possible (and therefore allowed for multiple heads for each word), the double coding resulted in a slight difference (66) in the total number of dependencies annotated. In the end, the primary interest was whether or not the annotators agree upon the attachments and the labels assigned. The measure of inter-annotator reliability reported here, Cohen's Kappa, was computed on the basis of 541 dependencies jointly identified by the author and the second annotator. The unweighted kappa for inter-annotator reliability is .86 at $p < .01$.

6. Study 1: Parser accuracy for L2 Dutch

The parser evaluation results are presented first from a global perspective, afterwards going in for a closer look at the three target dependencies outlined in Section 1. The global parsing accuracy for the two parsers is presented in Table 2, averaged across all speakers, and then across CEFR exam levels. Accuracy is measured with the LAS, as labeled dependencies are required for the accurate identification of the target relations, and is calculated as the number of correctly attached and labeled dependencies over the total number of identified dependencies. These global indicators of parsing accuracy should be interpreted in the context of the original reported accuracy levels for each parser

¹ We would like to thank Zoë Van Vooren for her participation in this endeavor.

(90.9% for Alpino; 74.3% for Frog). Overall, both parsers perform on average with slightly less accuracy than their respective original reported accuracy levels. It should also be noted that the original reported accuracy for Alpino is measured differently than the LAS reported for Frog and in the present article. Original accuracy measures for Alpino were reported in concept accuracy (see Section 4.1), computed as the number of inaccurate/missing parses in the parser output averaged over either the total number of dependencies identified by the parser or by the gold standard (whichever number is larger). In general, a tentative conclusion can be drawn from these preliminary results that both parsers appear to lose some performance when presented with learner language input, with an overall LAS of 83.7% for Alpino and 66.5% for Frog.

Table 2. Labeled attachment score across parsers and proficiency levels

	LAS	
	Alpino	Frog
Overall	83.7%	66.5%
B1	76.4%	65.3%
B2	84.7%	66.7%
C1	89.7%	67.5%

6.1 Accuracy across proficiency

As discussed in Section 2, proficiency is one of the central interests in learner corpus research, and at the same time poses a challenge for the implementation of automatic processing (especially considering the native training data of the parsers). In this section, the performance of both parsers is evaluated across the three CEFR exam levels included among the sample (B1, B2, and C1). The LAS across parsers and proficiency level reported in Table 2 indicate a linear increase in accuracy for both Alpino and Frog, though

this pattern is more evident for Alpino. These results are represented visually in Figure 5. While the LAS remains stable with little variation across CEFR level in the Frog parses, the difference in LAS between the C1 sample and the B1 sample is around 13% in the Alpino annotations. While Alpino performs at 1.2% lower than the original reported accuracy on the highest band of proficiency, Frog performs at 6.8% lower than the original reported accuracy on the C1 productions.

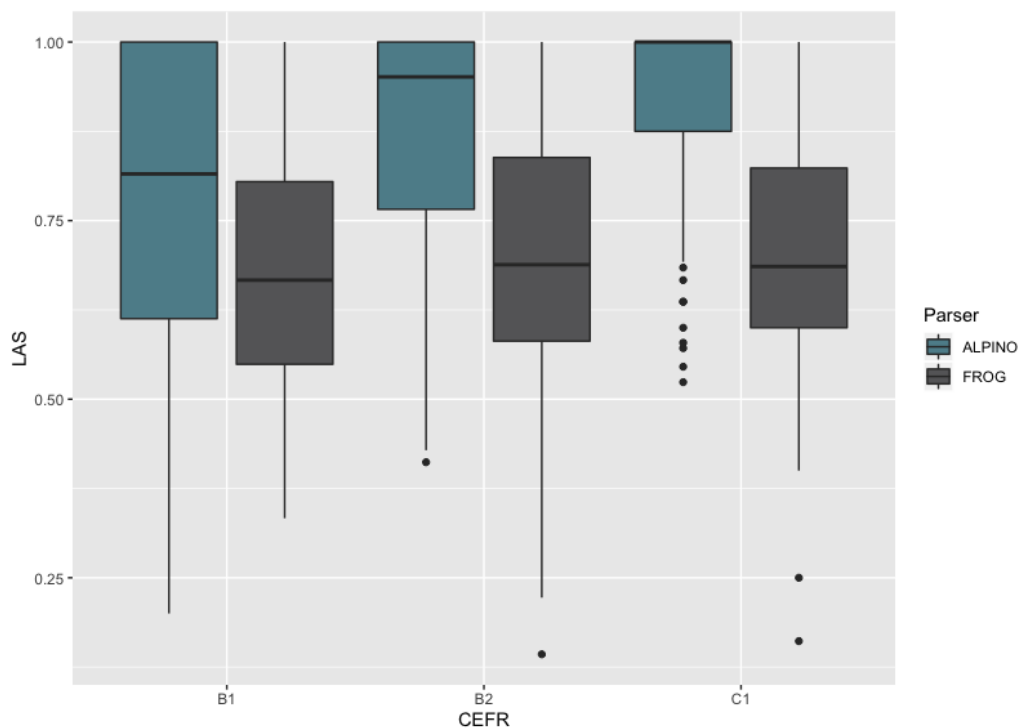


Figure 5. Boxplot of LAS across parsers and CEFR level

To test the significance of the observed differences in LAS between parsers and across CEFR levels, a logistic mixed-effects regression model was fit to the data. The model outcome variable is the binary success/failure of the attachment and label of the 4,900 words in the sample. A random intercept was entered to account for inter-speaker variation in parsing outcomes. For the fixed effects, the CEFR level of the exam

(B1/B2/C1) and the parser (Alpino/Frog) were entered as predictors, in addition to the interaction between the two. The model fit was not improved further by removing either of the predictors or their interaction, as determined by the AIC. The model fit is significantly better than the null model: LR $\chi^2(5) = 465.47$, $p < .01$. R^2 values for the model are quite low (marginal $R^2 = .096$; conditional $R^2 = .127$), though it was not expected that the parser and the CEFR level of the exam alone can account for the prediction of an accurate parsing outcome. The model output is presented in Table 3. The intercept in the model signifies the log odds of predicting a successful parse outcome when the parser is Alpino and the exam CEFR level is B1. The significance of the interaction between parser and CEFR level indicates that the change in the likelihood of a successful parse outcome across proficiency levels is significantly different between the two parsers, a pattern that was observed in the raw data visualization (Figure 5). For Alpino, the likelihood of a successful parse outcome significantly increases as CEFR level increases from B1 to B2 and from B1 to C1. An increase in CEFR level is associated with a significantly smaller increase in the likelihood of an accurate parse for Frog.

Table 3. Binary mixed-effects regression model summary

Random effects:				
<i>Groups</i>	<i>Name</i>		<i>Variance</i>	<i>Std. Dev.</i>
Speakers	Intercept		0.1147	0.3387
Fixed effects:				
<i>Predictors:</i>	<i>Estimate</i>	<i>Std. Error</i>	<i>z value</i>	<i>Pr(> z)</i>
Intercept**	1.2	0.12	9.76	< .01
CEFRB2**	0.61	0.18	3.45	< .01
CEFRC1**	1.07	0.18	5.85	< .01
FROG**	-0.55	0.08	-6.88	< .01
CEFRB2:FROG**	-0.49	0.12	-4.17	< .01
CEFRC1:FROG**	-0.91	0.13	-7.2	< .01

** Significant at $p < .01$

6.2 Accuracy across dependency relations

Performance was expected to fluctuate across specific dependency relations, as some structures can present greater challenges for automatic annotation than others. Of the three target dependency relations investigated here (direct objects, adjectival modifiers, and adverbial modifiers), direct objects were expected to exhibit the most erroneous annotation. The accurate extraction of the three dependency relations makes a more demanding request from the parser, as the attachment, label, and POS-tag of the head all need to be accurately annotated to identify each of the three relations. This is because the dependency labels are not specific enough for this level of granularity. For example, a MOD label applies to both adjectival and adverbial modifier relations; a distinction can therefore only be made on the basis of the POS annotation.

The accurate annotation of *dobj*, *amod*, and *advmod* relations is measured in terms of precision, recall, and F-score (harmonic mean of precision and recall). The results of this comparison across dependency relations, parsers, and CEFR levels indeed reveal some more nuanced performance patterns. Tables 4-6 present the accuracy metrics for the identification of *dobj*, *amod*, and *advmod* relations, respectively. Distinct patterns of accuracy are observed between parsers and across CEFR levels in the identification of direct object dependencies. Compared with the overall accuracy reported in Table 2, the accurate attachment and labeling of direct object dependencies, with the additional criterion of an accurate POS-tag for the verbal head, presents a challenge for both parsers, but especially for Frog. Across CEFR levels, the F-scores increase from B1 to B2 and then decrease from B2 to C1 for both parsers.

Table 4. Extraction accuracy of direct object relations across parsers and CEFR level

	Alpino			Frog		
	F-score	Precision	Recall	F-score	Precision	Recall
Overall	78.4%	78.2%	78.5%	36.5%	40.7%	33.2%
B1	67.1%	66.2%	68.1%	35.2%	36.1%	34.4%
B2	81.0%	82.6%	79.2%	38.9%	46.0%	33.7%
C1	67.1%	66.2%	68.1%	35.1%	39.7%	31.5%

Table 5. Extraction accuracy of adjectival modifier relations across parsers and CEFR level

	Alpino			Frog		
	F-score	Precision	Recall	F-score	Precision	Recall
Overall	89.3%	93.4%	85.5%	80.8%	78.4%	83.4%
B1	80.9%	92.7%	71.7%	86.4%	92.7%	81.0%
B2	91.1%	95.1%	87.5%	74.6%	67.8%	83.0%
C1	92.5%	91.9%	93.2%	84.8%	83.8%	86.0%

Table 6. Extraction accuracy of adverbial modifier relations across parsers and CEFR level

	Alpino			Frog		
	F-score	Precision	Recall	F-score	Precision	Recall
Overall	77.1%	75.8%	78.5%	48.6%	58.4%	41.7%
B1	71.3%	64.3%	80.0%	49.4%	58.1%	43.0%
B2	72.2%	74.3%	70.3%	51.7%	63.5%	43.5%
C1	85.0%	87.9%	82.3%	44.2%	52.8%	38.0%

Of the three target dependency relations, the adjectival modifiers were annotated with the highest degree of success by both parsers. However, whereas F-scores for adjectival modifiers increase linearly across CEFR levels for Alpino, the F-score decreases from B1 to B2 for Frog, with an increase again from B2 to C1. The difference between the two parsers in the identification of adjectival modifier relations is only marginal in comparison with the direct object relations. Adverbial modifier relations also appear to present a challenge here for automatic identification, with an overall F-score of 77.1% for Alpino and 48.6% for Frog. The difference in performance between parsers and across

CEFR level reflects the patterns observed for adjectival modifier relations: higher performance of Alpino compared to Frog, a linear increase in F-scores across CEFR level for Alpino, but then an increase in F-score from B1 to B2 followed by a decrease from B2 to C1 for Frog.

6.3 A qualitative look at parsing error

To understand the quantitative results presented above on a more fine-grained level, the present section provides some qualitative insight into the observed differences between parsers. Although Alpino outperforms Frog in the majority of the comparisons presented here, the dependency trees in Figures 6 and 7 demonstrate the annotation of Example 3 (shown above in Section 5) by both parsers, where the annotation provided by Alpino appears to have been much more affected by the lack of morphosyntactic agreement than Frog.

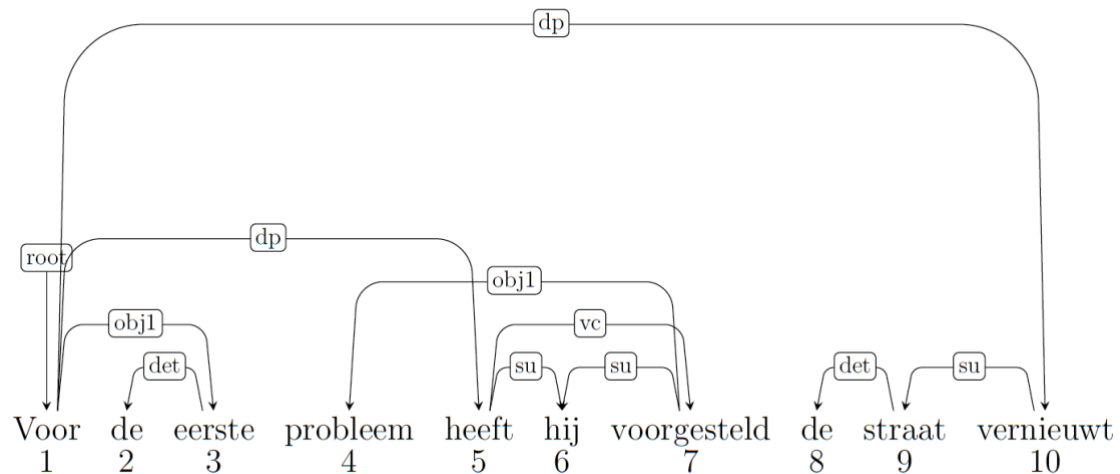


Figure 6. Dependency tree for Alpino annotation of Example 3 from B1 CNaVT text

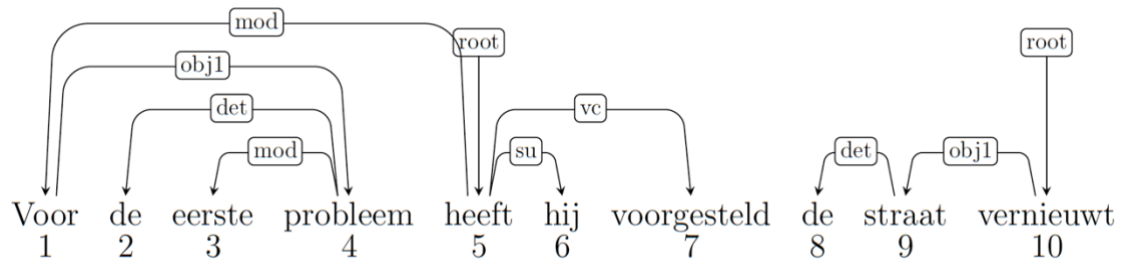


Figure 7. Dependency tree for Frog annotation of Example 3 from B1 CNaVT text

In Figure 6, the dependency tree for the Alpino annotation of the sentence shows an intransitive argument structure for the verb “vernieuwen”, a verb that is almost always used transitively in Dutch. This interpretation from Alpino likely has to do with the inflectional marking on the verb “vernieuwen” in what is expected to be an infinitive clause, which seems to preclude the accurate attachment and labeling of the two clauses in the sentence. Also interesting is the identification of the preposition “Voor” as the root of the sentence, despite the presence of two finite verbs. Figure 7 shows the dependency tree for the Frog annotation of the same sentence where “straat” is accurately attached to the verb “vernieuwen” in a direct object dependency relation, though this subclause is also not attached to the main clause in this dependency structure. In Section 5, the lack of morphosyntactic agreement between the neuter noun “probleem” and the non-neuter definite article “de” was highlighted. The argument structure allowed by the head nodes in the surrounding context informed the decision to attach the non-neuter article to the neuter noun in the target hypothesis. While the same attachment is made in the annotation provided by Frog, the non-neuter definite article in the Alpino annotation is attached instead to “eerste”, which can serve as both modifier and noun in Dutch, and in the latter case takes a non-neuter definite article. This leaves the noun “probleem” to attach as an

argument elsewhere in the sentence. As a consequence, the Alpino annotation is far removed from the target hypothesis proposed in Section 5.

The sentence presented below in Example 4, sampled from a B1 CNaVT exam, presents a challenge for both parsers due to the use of the plural morpheme “-en” in the word “verkeersdempelen” (correct form: “verkeersdempels”) and the use of third person singular inflectional suffix “-t” with the noun “begint” (correct form: “begin”). Figures 8 and 9 correspond to the annotations provided by Alpino and Frog for Example 4.

(4) <i>Ze</i>	<i>kunn-en</i>	<i>verkeersdempel-en</i>	<i>in</i>	<i>het</i>
3PL.NOM	AUX.can-3PL	speed.bump-PL	in	DEF.SG.N
<i>begin-t</i>	<i>van</i>	<i>de</i>	<i>straat</i>	<i>plaats-en.</i>
begin-3SG	of	DEF.SG	street	place-INF
‘They can place speed bumps at the start of the street.’				

The parsers handle the plural marker “-en” on the noun “verkeersdempel” in rather different ways, and neither approach aligns with the target hypothesis. The dependency annotation provided by Alpino identifies “verkeersdempelen” as a noun, but attaches it to the verb “kunnen” as a locative complement. This seems to be a result of the placement of the noun between the verb and the preposition “in”, which the parser identified as the particle of a separable complex verb “inkunnen”, though this is not one of the many separable complex verbs attested in Dutch. In the dependency tree produced by Frog, the plural morpheme “-en” is interpreted as the infinitive marker “-en”, and “verkeersdempelen” is thus identified as the head of an infinitive clause. The inflected ending “-t” on the noun “begin” is interpreted by both parsers as the verb “beginnen” (English: ‘begin’), though still resulting in two very different dependency structures. Both parsers then identify “het” as a subject pronoun attached to verbal “begin”, as opposed to the neuter definite article of the neuter noun “begin”, as in the target hypothesis. While

the impact of these non-native-like features of learner language, specifically a lack of morphosyntactic agreement, on parser performance is not altogether surprising, the way the parsers handle these features can help to shed light on their suitability for learner corpus research. With an error-tagged learner corpus, a systematic evaluation of parser performance can be carried out across types of learner errors, along the lines of the analysis carried out in Huang et al. (2018), for a better overview of these trends.

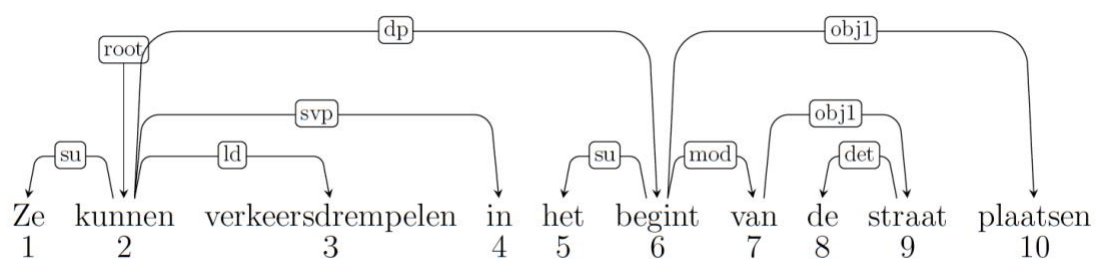


Figure 8. Dependency tree for Alpino annotation of Example 4 from B2 CNaVT text

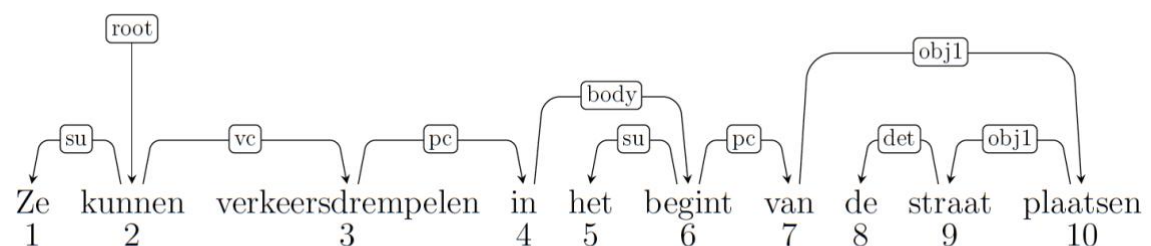


Figure 9. Dependency tree for Frog annotation of Example 4 from B2 CNaVT text

7. Study 2: Impact of automatic annotation on measures of phraseological sophistication

Despite the increasing use of automatic dependency annotation for learner corpus data, the impact of these annotations on subsequent research findings is not often investigated,

nor problematized. This second study therefore aims to quantify the impact of the automatically generated annotations on phraseological complexity research. Following the operationalization of the statistical association measure pointwise mutual information (MI) as an indicator of phraseological sophistication in learner corpus research (Durrant & Schmitt, 2009; Granger & Bestgen, 2014; Paquot, 2018, 2019), this analysis will focus on the effect of the dependency annotations on the MI scores computed for the three target phraseological units. MI measures the degree of attraction between two lexical units given their joint and respective distributions in a corpus, where a higher value for MI indicates a stronger degree of attraction and is interpreted here as increased sophistication. After having assessed the suitability of the two parsers for the L2 data at hand, the measures of phraseological sophistication were computed for the phraseological units extracted on the basis of both the gold standard annotations and the Alpino annotations, which were determined to be more accurate than the Frog annotations, especially for the direct object dependencies. The variation in mean MI scores per text across the three dependency relations is first explored visually, and then quantified by means of Pearson's correlations.

7.1 Computing phraseological sophistication

MI scores are computed on the basis of a larger reference corpus in order to increase generalizability. For the present analysis, MI was computed on the basis of frequency of occurrence information of the target dependency relations in the LASSY Groot corpus (van Noord, Schuurman, & Vandeghinste, 2006). Direct object, adjectival modifier, and adverbial modifier relations extracted from the Dutch learner texts on the basis of both the gold standard and automatic dependency annotations were searched in the extensive lists of the relations extracted from the reference corpus (> 8.5 million types of the three

target relations), and were assigned the MI score computed for the same relation in the reference corpus. MI was computed for all three dependency types extracted from the reference corpus using the Ngram Statistics Package (Banerjee & Pedersen, 2003). Due to the size of the reference corpus (close to 350 million words were used here), most relations extracted from the learner corpus could be identified in the reference corpus. Relations extracted from the learner data that were not able to be identified in the reference corpus were not assigned an MI score and did not factor into the calculation of phraseological sophistication. Mean MI scores were then calculated for each of the three dependency relations for each text.

7.2 Impact of automatic annotation on the measurement of phraseological complexity

The extent of the impact of automatic dependency annotation of learner language, as compared with manual annotation, is evaluated here in terms of the mean MI score for three target dependency relations (*doobj*, *amod*, and *advmod*) across the 30 texts included in the sample. The mean MI scores computed for both the manual and automatically annotated dependencies are presented visually in Figure 10.

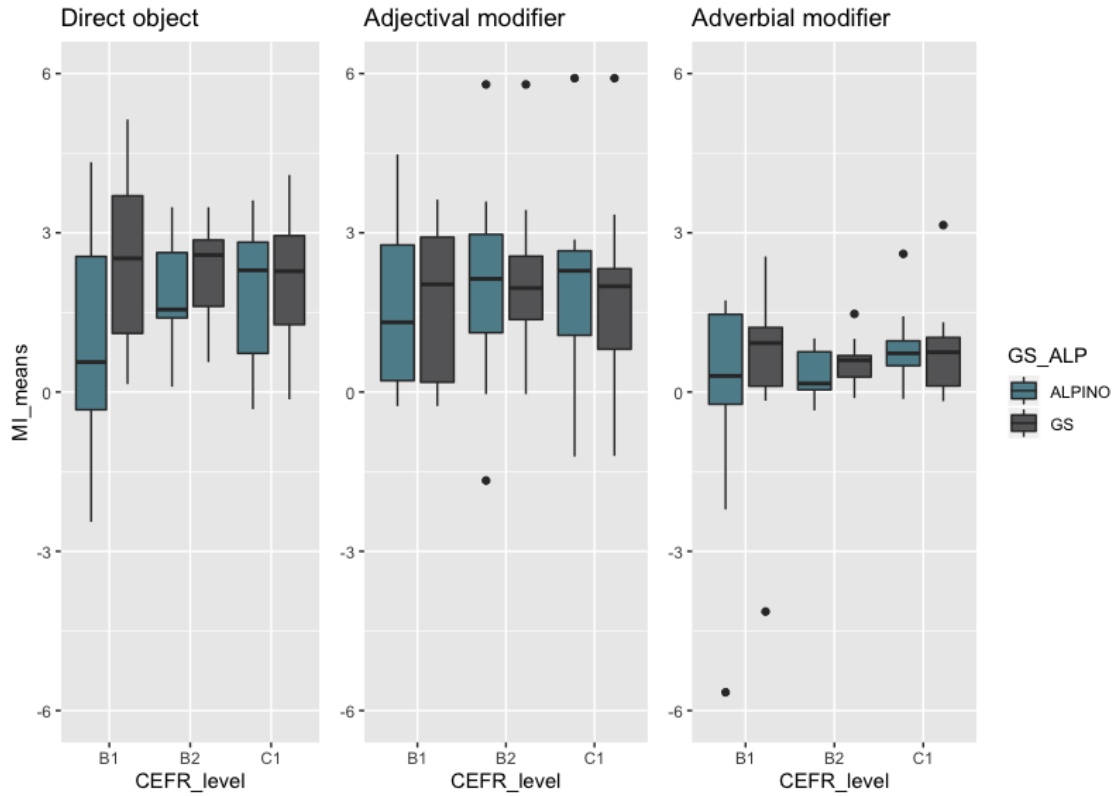


Figure 10. Mean MI score for dobj, amod, and advmod relations across CEFR levels and manual/automatic annotation

The impact of CEFR level on the measures computed for the automatically annotated dependencies is seemingly minimal, with the exception of the direct object dependencies at the B1 level. However, the small number of texts per proficiency level (10 per level) limits the possibility for inferential statistics at this level of granularity. The Pearson correlation coefficient for the correlation between the MI means computed on the basis of the two annotations (for all dependencies) is moderate to high ($R = .69$, $p < .01$), indicating a moderate to low overall impact of the automatic annotations on the measures. The next section considers the results on a more fine-grained level by looking at the impact of the annotation on MI means across the three dependency relations.

7.3 Effect of dependency relations

The correlation between the MI means computed on the basis of the gold standard annotation and the automatic annotation also differs moderately between dependency relations, in a pattern that reflects the extraction accuracy across dependency relations (see Tables 4-6). The MI means computed on the two annotation sets are plotted against each other for each of the three target dependency relations in Figure 11, where the y-axis indicates the MI means computed on the basis of the parser annotation and the x-axis the MI means computed on the basis of the gold standard annotation. The Pearson correlation coefficient is largest between the adjectival modifier annotations, followed by the adverbial modifiers, and finally the direct objects.

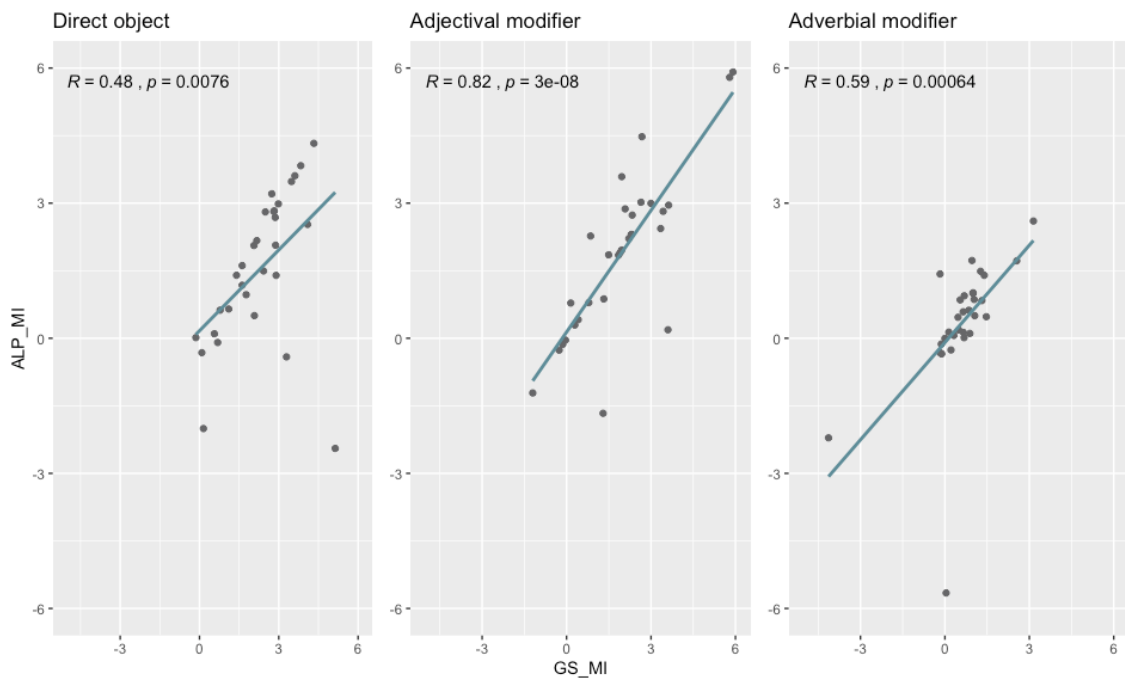


Figure 11. Correlation between MI means based on manual and automatic annotation across dependency relations

7.4 A qualitative look at the impact of parsing error

The discrepancies between the gold standard annotation and the automatically generated annotations were reviewed qualitatively in Section 6.3. Here, the impact of those differing dependency annotations are considered in terms of the phraseological units extracted using both annotation sets and the MI scores assigned to them. In the sentence presented in Example 3, reproduced below as Example 5, Alpino identified one direct object dependency (stel_voor+verb, probleem+noun; English: propose+verb, problem+noun) and no modifier dependencies (see Figure 6).

(5) *Voor de eerste probleem heeft hij*
 For DEF.SG first problem AUX.have.3SG 3SG.M.NOM
voorgesteld de straat vernieuw-t
 proposed.PTCP DEF.SG street renovate-3SG
 ‘For the first problem, he proposed the street renovates’

In the gold standard annotation (see Figure 4), one direct object dependency (vernieuw+verb, straat+noun; English: renovate+verb, street+noun) and one adjectival modifier dependency (eerst+adj, probleem+noun, English: first+adj, problem+noun) were identified. The MI scores assigned to these three dependencies on the basis of the reference corpus can be consulted in Table 7.

Table 7. MI scores assigned to dependencies extracted from Example 5

Annotation	Dependency relation	MI score
Alpino	stel_voor+verb, probleem+noun	-3.0625
Gold standard	vernieuw+verb, straat+noun	5.0516
Gold standard	eerst+adj, probleem+noun	-6.2951

This example helps to provide context for the correlations reported in Figure 11 between the MI means computed on the basis of phraseological units identified using both manual

and automatic annotations. The phraseological units extracted using both annotation sets delivered quite different MI scores, which are then added into the calculation of MI means as a sophistication measure. The annotations provided by Alpino for this sentence will contribute to a lower value for direct object sophistication, with no effect on the value for adjectival modifier sophistication. On the other hand, the gold standard annotations will contribute to a higher value for direct object sophistication and a lower value for adjectival modifier sophistication for this speaker.

8. Discussion and conclusions

The present investigation into dependency parsing Dutch learner language set out to answer some central questions surrounding the impact of automatic dependency annotation on measures of phraseological sophistication computed on the basis of such annotations. Given the substantial advantages to be gained from automatic linguistic annotation, and the increasingly frequent use of dependency parsers in LCR, the core concern is whether these tools are applicable for learner language, which can be characterized in part by varying degrees of individual and group variation. The first of the two studies presented in this article addressed the performance of the two Dutch dependency parsers Alpino and Frog on L2 Dutch texts, and the stability of the performance across CEFR levels and target dependency relations.

On a general level, both parsers perform lower than their original reported accuracy when presented with learner language input (original reported accuracy levels: 90.9% for Alpino, 74.3% for Frog; overall accuracy reported here: 83.7% for Alpino, 66.5% for Frog). These accuracy values are in line with the results of similar investigations of parser accuracy for L2 English (Huang et al., 2018) and L2 German

(Krivanek & Meurers, 2012; Ott & Ziai, 2010). When factoring in proficiency level, different patterns emerge between the two parsers. In Section 6.1, the results of a binary mixed-effects regression model confirm a significant interaction between CEFR level and parser in the prediction of an accurate parse outcome for each dependency. Put more directly, as the CEFR level of the exam increases from B1 to B2 and from B1 to C1, the likelihood of an accurate parse outcome from Alpino (B1: 76.4%; B2: 84.7%; C1: 89.7%) increases significantly more than from Frog (B1: 65.3%; B2: 66.7%; C1: 67.5%), where the performance remains more stable across CEFR level. These findings also highlight the potential for a linear increase in parser accuracy across proficiency level to introduce bias into subsequent research mapping complexity onto learner proficiency. Specifically, less accurate annotations at the lower proficiency levels can lead to less accurate measurement at the lower proficiency levels in subsequent analyses, potentially resulting in the observation of a non-existent effect of proficiency. While research into dependency parsing of learner language has focused on the impact of features of learner language as a homogeneous variety, the results of the present analysis suggest that the dimension of learner proficiency receive more attention in the evaluation of annotation tools in LCR moving forward.

As reported in previous findings (e.g. Ott & Ziai, 2010), parser accuracy also varies across dependency relations. Accuracy metrics for the extraction of the three target dependency relations (dobj, amod, advmod) presented in Tables 4-6 indicate an overall lower performance for direct object relations than for the two adjunct relations. This could be partly linked to the frequent occurrence of long-distance verb-object dependencies with (potentially non-native-like) intervening material, and the more local nature of adjunct relations. Whereas Alpino demonstrates a linear increase in accuracy (measured

in F-score) across CEFR levels for the two adjunct relations, Frog exhibits the same pattern of an initial increase from B1 to B2 followed by a decrease from B2 to C1 across direct object and adverbial modifier relations, and a reversed trend for adjectival modifier relations. In a comparison of parser performance between learner and native data, Huang et al. (2018) found that while accuracy was generally higher for native data, sentence length was found to significantly negatively correlate with parsing accuracy, offering somewhat of an advantage to learner data, which on average consists of shorter sentences. This potential competition between accuracy (more native-like language) and complexity (increased sentence length) may help to account for some of the non-linear trends in annotation accuracy across dependency relations and proficiency levels in the present analyses.

The second study presented here set out to assess the impact of automatic annotation on phraseological complexity measures computed on the basis of the extracted phraseological units, specifically highlighting the operationalization of the association measure MI as an indicator of phraseological sophistication. The overall higher accuracy of the Alpino annotations across the three dependency relations and proficiency levels was preferred over the Frog annotations for the second study. In a comparison between phraseological sophistication measures computed on the basis of the manual and automatic dependency parses, dependency relations with lower annotation accuracy (i.e. direct object relations) demonstrate a lower degree of correlation between the MI means computed from the two annotations. The use of automatic annotation tools does therefore appear to come at the cost of lower annotation accuracy, which in turn has an impact on the complexity measures computed on the basis of the phraseological units extracted.

The correlations between the manual and automatic annotations range from $R = .48$ (direct object dependencies) to $R = .82$ (adjectival modifier dependencies). The differences in the impact of automatic annotation on the sophistication measures across the three dependency relations suggests that phraseological complexity measures computed for some dependency relations may be less precise than for other dependency relations. The example presented in Section 7.4 demonstrates concretely how annotation errors can introduce noise into the complexity measures computed in subsequent analyses. The findings reported from both studies indicate that both proficiency level and dependency relations can function as moderating factors in the automatic annotation of dependency relations for learner language, with the potential to negatively impact the interpretation of subsequent analyses.

8.1 Implications for automatic annotation in LCR

The evaluation of performance across parsers, proficiency level, and dependency relations presented here explores some of the various factors that contribute to the performance of parsers when processing learner language. While Frog exhibits an overall lower accuracy level, the parser performance remains relatively stable across the three CEFR exam levels. On the other hand, the accuracy of the annotations provided by Alpino remain more stable across dependency relations, and increase linearly across proficiency levels. An important conclusion that can be drawn from these findings is the importance of rigorously evaluating NLP tools for the specific context in which they will be used, as various factors can influence performance (e.g. learner proficiency and target dependencies). Furthermore, the evaluation of automatic dependency parsers is rarely extended to assess the impact on subsequent analyses making use of the annotations.

Many tools have been developed for (mainly English) L2 complexity research that make use of the raw output of (dependency) parsers to compute a battery of complexity measures, without due consideration for the validity of the underlying annotations. Understanding the impact of automatic annotation on phraseological unit extraction is a crucial step to contextualize findings in L2 complexity research. The present analysis emphasizes the importance of taking this additional step.

Large-scale corpus analyses of learner language are made possible in part by automatic linguistic annotation, and research into the use of automatic annotation tools on learner productions has produced some promising results. Though non-native-like features of learner language have been shown to be a major source of parsing error, parsers have also been demonstrated to perform at high levels of accuracy on learner data (Huang et al., 2018). The reduced accuracy observed for automatically annotated dependency relations should not deter the use of these tools in large-scale learner corpus annotation projects, as the alternative would be to restrict learner corpus research to small-scale analyses. Rather, the focus going forward should be placed on incorporating NLP tools responsibly into LCR, by taking steps to assess the impact of the tools on research carried out using the annotations (or other output) that they provide.

8.2 Limitations and future directions

A more systematic analysis of the influence of non-native-like features on parsing performance is required in order to provide deeper insight into the differences between the way the parsers deal with learner language. Despite the substantial time required to annotate target hypotheses manually, this would also allow for more concrete conclusions concerning the behavior of the parsers across learner proficiency levels and dependency

relations. Moving forward, working with a larger manually-annotated learner corpus would also allow for statistical analyses that control for additional moderating factors, which will become increasingly important as the needs of the field grow more specific. One such variable that the present analysis does not account for is domain differences between the data used to train the parsers and the texts that are parsed. Although the Alpino parser has been demonstrated to perform at similar levels across domains (van Noord, 2006), the effect of domain and the potential interaction between the domain and the other moderating factors have not been measured in the present analyses.

The tools considered here are specific to the Dutch language, which was thought to be preferable when dealing with learner language, as these generally have high accuracy rates on their intended language samples, and available annotated training data for Dutch is sparse (both parsers considered here made use of the same treebank of around 7,000 sentences). However, an increasingly popular trend in dependency parsing is the application of the universal dependency (UD) annotation scheme (de Marneffe, Dozat, Silveira, Haverinen, Ginter, Nivre, & Manning, 2014), capable of encoding dependency relations across multiple languages. The benefit of these approaches is the standardization across dependency annotation and the ability to implement any gold standard training set available. Such tools may prove to provide an advantage to learner language annotation, being less sensitive to typological differences (de Marneffe & Nivre, 2019), and are therefore worth exploring.

References

- Banerjee, S., & Pedersen, T. (2003). The design, implementation, and use of the Ngram Statistics Package. In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics.
- Bestgen, Y. (2017). Beyond single-word measures: L2 writing assessment, lexical richness and formulaic competence. *System*, 69, 65-78.
- Bouma, G., & Kloosterman, G. (2007). Mining syntactically annotated corpora with XQuery. In Proceedings of the linguistic annotation workshop (pp. 17-24).
- Boyd, A., & Meurers, D. (2008). Revisiting the impact of different annotation schemes on PCFG parsing: a grammatical dependency evaluation. In Proceedings of the workshop on parsing German (pp. 24-32).
- Carlsen, C. (2012). Proficiency level—a fuzzy variable in computer learner corpora. *Applied Linguistics*, 33(2), 161-183.
- Council of Europe. (2001). Common European framework of reference for languages: Learning, teaching, assessment. Cambridge, UK: Cambridge University Press.
- Daelemans, W., van Den Bosch, A., & Weijters, T. (1997). IGTrees: using trees for compression and classification in lazy learning algorithms. In *Lazy learning* (pp. 407-423). Springer.
- de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal Stanford Dependencies: A cross-linguistic typology. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). Reykjavik, Iceland: European Language Resources Association (ELRA).
- de Marneffe, M.-C., & Nivre, J. (2019). Dependency grammar. *Annual Review of Linguistics*, 5, 197-218.

- Díaz-Negrillo, A., Meurers, D., Valera, S., & Wunsch, H. (2010). Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1-2), 139-154. (Special Issue on Corpus Linguistics for Teaching and Learning. In Honour of John Sinclair)
- Dickinson, M., & Ragheb, M. (2009). Dependency annotation for learner corpora. In M. Passarotti, A. Przepiórkowski, S. Raynaud, & F. Van Eynde (Eds.), *Proceedings of the eighth international workshop on treebanks and linguistic theories* (pp. 59-70).
- Durrant, P., & Schmitt, N. (2009). To what extent do native and non-native writers make use of collocations? *International Review of Applied Linguistics in Language Teaching*, 47(2), 157-177.
- Granger, S., & Bestgen, Y. (2014). The use of collocations by intermediate vs. advanced non-native writers: A bigram-based study. *International Review of Applied Linguistics in Language Teaching*, 52(3), 229-252.
- Granger, S., & Paquot, M. (2008). Disentangling the phraseological web. In S. Granger & F. Meunier (Eds.), *Phraseology: An interdisciplinary perspective* (pp. 27-49). Amsterdam, Philadelphia: John Benjamins.
- Gries, S. T. (2008). Phraseology and linguistic theory: a brief survey. In S. Granger & F. Meunier (Eds.), *Phraseology: An interdisciplinary perspective* (pp. 3-26). Amsterdam, Philadelphia: John Benjamins.
- Heid, U. (2008). Computational phraseology: an overview. In S. Granger & F. Meunier (Eds.), *Phraseology: An interdisciplinary perspective* (pp. 337-360). Amsterdam, Philadelphia: John Benjamins.

- Housen, A., & Kuiken, F. (2009). Complexity, accuracy, and fluency in second language acquisition. *Applied linguistics*, 30(4), 461-473.
- Huang, Y., Murakami, A., Alexopoulou, T., & Korhonen, A. (2018). Dependency parsing of learner English. *International Journal of Corpus Linguistics*, 23(1), 28-54.
- Krivanek, J., & Meurers, D. (2013). Comparing rule-based and data-driven dependency parsing of learner language. *Computational Dependency Theory*, 258, 207.
- Lüdeling, A., Walter, M., Kroymann, E., & Adolphs, P. (2005). Multi-level error annotation in learner corpora. *Proceedings of corpus linguistics 2005*.
- Meurers, D. (2009). On the automatic analysis of learner language: Introduction to the special issue. *CALICO Journal*, 26(3), 469-473.
- Meurers, D., & Dickinson, M. (2017). Evidence and interpretation in language learning research: Opportunities for collaboration with computational linguistics. *Language Learning*, 67(S1), 66-95.
- Meurers, D., & Wunsch, H. (2010). Linguistically annotated learner corpora: Aspects of a layered linguistic encoding and standardized representation. *Proceedings of Linguistic Evidence*.
- Norris, J. M., & Ortega, L. (2009). Towards an organic approach to investigating CAF in instructed SLA: The case of complexity. *Applied Linguistics*, 30(4), 555-578.
- Ordeman, R. J. F., de Jong, F. M. G., van Hessen, A. J., & Hondorp, G. H. W. (2007). TwNC: a Multifaceted Dutch News Corpus. *ELRA Newsletter*, 12(3-4).
- Ortega, L. (2003). Syntactic complexity measures and their relationship to L2 proficiency: A research synthesis of college-level L2 writing. *Applied Linguistics* 24 (4), 492-518.

- Ott, N., & Ziai, R. (2010). Evaluating dependency parsing performance on German learner language. In Proceedings of the ninth international workshop on treebanks and linguistic theories (Vol. 9, pp. 175-186).
- Paquot, M. (2018). Phraseological competence: A missing component in university entrance language tests? insights from a study of EFL learners' use of statistical collocations. *Language Assessment Quarterly*, 15(1), 29-43.
- Paquot, M. (2019). The phraseological dimension in interlanguage complexity research. *Second Language Research*, 35(1), 121-145.
- R Core Team. (2017). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.
- Ragheb, M., & Dickinson, M. (2012). Defining syntax for learner language annotation. In Proceedings of COLING 2012 (pp. 965-974).
- Reznicek, M., Lüdeling, A., & Hirschmann, H. (2013). Competing target hypotheses in the FALKO corpus. In A. Díaz-Negrillo, N. Ballier & P. Thompson (Eds.), *Automatic Treatment and Analysis of Learner Corpus Data*, John Benjamins 101–123.
- Rubin, Housen, & Paquot (forthcoming). Phraseological complexity as an index of L2 Dutch writing proficiency: A partial replication study. In S. Granger (Ed.), *Perspectives on the Second Language Phrasicon: The View from Learner Corpora*. Bristol: Multilingual Matters.
- Sharwood Smith, M. & Truscott, J. (2005). Stages or Continua in Second Language Acquisition: A MOGUL Solution. *Applied Linguistics*, 26(2), 219-240.

- Tsarfaty, R., Nivre, J., & Andersson, E. (2011). Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. In Proceedings of the 2011 conference on empirical methods in natural language processing (pp. 385–396).
- van den Bosch, A., Busser, B., Canisius, S., & Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In Proceedings of the 17th meeting of Computational Linguistics in the Netherlands.
- van der Beek, L., Bouma, G., Malouf, R., & van Noord, G. (2002). The Alpino dependency treebank. In Computational linguistics in the Netherlands 2001 (pp. 8-22).
- van Noord, G. (2006). At last parsing is now operational. In TALN 2006 (pp. 20-42).
- van Noord, G., Schuurman, I., & Vandeghinste, V. (2006). Syntactic annotation of large corpora in STEVIN. In Proceedings of the fifth international conference on language resources and evaluation (LREC'06). Genoa, Italy: European Language Resources Association (ELRA).
- van Noord, G., Schuurman, I., & Bouma, G. (2011). Lassy Syntactische Annotatie, Revision 19455, 208p.
- Weiss, Z. & Meurers, D. (this issue). Analyzing the linguistic complexity of German learner language in a reading comprehension task: Using proficiency classification to investigate short answer data, the impact of linguistic analysis quality, and cross-data generalizability. International Journal of Learner Corpus Research, Special Issue on NLP.

Appendix I. Shared dependency label set

Dependency label	Description: Dutch	Description: English
APP	appositie, bijstelling	apposition

BODY	romp (bij complementizer)	body of complementizer phrase
CMP	complementizer	complementizer
CNJ	lid van nevenschikking	conjunct
CRD	nevenschikker (als hoofd van conjunctie)	coordinator
DET	determinator	determiner
DLINK	discourse link	discourse link
DP	discourse particle	discourse particle
HD	hoofd	head
HDF	afsluitend element van circumpositie	closing element of circumposition
LD	locatief of directioneel complement	locative or directional complement
ME	maat (duur, gewicht, ...) complement	measure complement
MOD	bijwoordelijke bepaling	adverbial
MWP	deel van een multi-word unit	part of a multi-word unit
NUCL	kernzin	nucleus
OBCOMP	vergelijkingscomplement	comparative
OBJ1	direct object, lijdend voorwerp	direct object
OBJ2	secundair object (meewerkend, belanghebbend, ondervindend)	indirect object
PC	voorzetselvoorwerp	prepositional object
POBJ1	voorlopig direct object	temporary direct object
PREDC	predicatief complement	predicative complement
PREDM	bepaling van gesteldheid 'tijdens de handeling'	predicative modifier
RHD	hoofd van een relatieve zin	head of a relative clause
SAT	satelliet; aan- of uitloop	satellite
SE	verplicht reflexief object	mandatory reflexive pronoun
SU	subject, onderwerp	subject
SUP	voorlopig subject	temporary subject
SVP	scheidbaar deel van werkwoord	separable verb particle
TAG	aanhangsel, tussenvoegsel	affix
VC	verbal complement	verbal complement
WHD	hoofd van een vraagzin	head of interrogative
