Discontinuous Constituency and BERT: A Case Study of Dutch

Konstantinos Kogkalidis[*] and **Gijs Wijnholds**[*]
Utrecht Institute of Linguistics OTS, Utrecht University
k.kogkalidis,g.j.wijnholds@uu.nl

Abstract

In this paper, we set out to quantify the syntactic capacity of BERT in the evaluation regime of non-context free patterns, as occurring in Dutch. We devise a test suite based on a mildly context-sensitive formalism, from which we derive grammars that capture the linguistic phenomena of control verb nesting and verb raising. The grammars, paired with a small lexicon, provide us with a large collection of naturalistic utterances, annotated with verb-subject pairings, that serve as the evaluation test bed for an attention-based span selection probe. Our results, backed by extensive analysis, suggest that the models investigated fail in the implicit acquisition of the dependencies examined.

1 Introduction

Assessing the ability of large-scale language models to automatically acquire aspects of linguistic theory has become a prominent theme in the literature ever since the inception of BERT (Devlin et al., 2019) and its many variants, largely due to their unanticipated performance. Standard practice involves attaching BERT to a shallow neural model of low parametric complexity, and training the latter at detecting various linguistic patterns of interest, revealing in the process the amount to which they are encoded within BERT's representations. The consensus points to BERT-like models having some capacity for syntactic understanding (Rogers et al., 2020). Their contextualized representations encode structural hierarchies (Lin et al., 2019) that can be projected into parse structures, using linear (Hewitt and Manning, 2019) or hyperbolic transformations (Chen et al., 2021), from which one can even obtain an accurate reconstruction of the underlying constituency tree (Vilares et al., 2020).

Despite their broadening scope, a latent bias persists in the insights provided by the probing literature, due to its focus being, by default, on English. English, albeit boasting a rich collection of evaluation resources, is characterized by a simple grammar with relatively few complications over the syntactic and morphological axes. Specifically when it comes to syntax, English lies in close proximity to a context-free language, a class characterized by its low rank in terms of formal complexity and expressive power (Chomsky, 1956). Perhaps more importantly, several commonly used evaluation test beds, including the Penn Treebank (Klein and Manning, 2001), are in themselves context-free, muddying the territory between probing for acquired syntactic generalization and arbitrating pattern extraction. As such, claims about the syntactic skills of language models should not be assumed to freely transfer between languages (and, in some cases, even datasets).

In this paper, we seek to evaluate BERT in the face of patterns that go beyond context-freeness. We employ a *mildly context-sensitive* grammar formalism to generate complex patterns that do not naturally occur in English. We choose instead to experiment on Dutch, a language long-argued to be non-context free, due it its capacity for exhibiting an arbitrary number of *cross-serial dependencies*. In Dutch, cross-serial dependencies arise in sentences where verbs form clusters, causing their respective dependencies with their arguments to intersect when drawn on a plane: Figure 1 portrays an adaptation of the example of Bresnan et al. (1982).

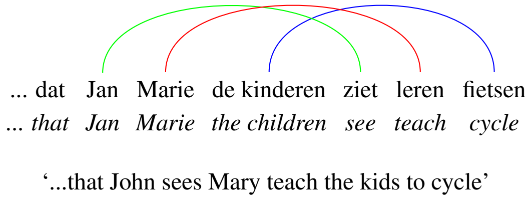
... dat Jan Marie de kinderen ziet leren fietsen
... that Jan Marie the children see teach cycle
'...that John sees Mary teach the kids to cycle'

Figure 1: Illustration of crossing dependencies in Dutch.

To that end, we first identify two well-studied constructions in Dutch that commonly involve cross-

[*] Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3776 - 3785
May 22-27, 2022 ©2022 Association for Computational Linguistics

serial dependencies: control verb nesting and verb raising. We produce an artificial but naturalistic dataset of annotated samples for each construction; each sample contains span annotations for the verb- and noun-phrases occurring within, as well as a mapping that associates each verb to its corresponding subject. We then implement a probing model intended to select a verb's subject from a number of candidate phrases, train it on a gold-standard resource of Dutch, and employ it on our data. Our experimental results convey a rapidly declining performance in the presence of discontinuous syntax, suggesting that the Dutch models investigated do not automatically learn to resolve the complex dependencies occurring in the language. To facilitate further research on the topic, our code is publicly available online.[1]

2 Background

2.1 Context freeness of natural languages

There has been a long debate, since the introduction of the Chomsky hierarchy (Chomsky, 1956), on whether all string patterns in natural language can be encompassed by the class of context-free grammars. The dispute often makes a distinction between *weak* and *strong* context-freeness, whereby the question shifts between generating all strings or all constituent expressions of a language. In Dutch specifically, patterns involving *cross-serial dependencies* have been commonly brought up by linguists in arguing that at least fragments of Dutch are context-sensitive, in turn designating the language not strongly context-free (Huybregts, 1984; Pullum and Gazdar, 1982; Bresnan et al., 1982; Shieber, 1985).

To capture such patterns without employing unnecessary computational expressiveness (and corresponding complexity), one can resort to the more pragmatic alternative of *mildly* context-sensitive grammars (Joshi, 1985): systems that can capture certain types of crossing dependencies, while remaining computationally tractable.[2]

2.2 Multiple Context-Free Grammars

One of the more general classes of mildly context-sensitive systems are multiple context-free grammars (MCFGs), which essentially generalizes the notion of a context-free grammars to operations on *tuples* of strings. We defer the reader to Seki et al. (1991) for a full definition and discussion of the properties of MCFGs. Instead we provide a simplified, computationally-oriented description that is more in line with our purposes and implementation. An m-multiple MCFG can be thought of as a tuple $\langle \mathcal{A}, \mathcal{N}, d, \mathcal{C}, \mathcal{R}, \textsc{s}_0 \rangle$, where:

- $\mathcal{A}$ is the terminal *alphabet*
- $\mathcal{N}$is a set of *non-terminals* and $d : \mathcal{N} \to \mathbb{N}$ a function from non-terminals to natural numbers; each non-terminal $\textsc{n}$ is encoding a tuple of strings of fixed arity $d(\textsc{n})$ and the maximal arity of $\mathcal{N}$ decides the grammar's multiplicity
- $\mathcal{C}$ is a mapping that associates each non-terminal $\textsc{n}$ to a (possibly empty) set of elements from the $d(\textsc{n})$-ary cartesian product $(\mathcal{A}^*)^{d(\textsc{n})}$; put simply, the set of constants $\mathcal{C}_\textsc{n}$ prescribes all the possible ways of initializing the non-terminal $\textsc{n}$
- $\mathcal{R}$ a set of *rewriting rules*; rules are functions $\mathcal{N} \times \cdots \times \mathcal{N} \to \mathcal{N}$ that provide recipes on how to combine a number of non-terminals into a single non-terminal by rearranging and concetenating their contents; we will write: $\textsc{c}(z_1, \ldots z_k) \leftarrow \textsc{a}(x_1, \ldots x_m)\ \textsc{b}(y_1, \ldots y_n)$ to denote a rule that combines non-terminals $\textsc{a}$ and $\textsc{b}$ of arities m and n into a non-terminal $\textsc{c}$ of arity k, where each of the left-hand side coordinates $x_1, \ldots y_n$ is used exactly once in the right-hand side coordinates $z_1, \ldots z_k$
- $\textsc{s}_0$ the *start* symbol, a distinguished element of $\mathcal{N}$ satisfying $d(\textsc{s}_0) = 1$

The choice of MCFGs as our formal backbone comes due to their many advantages. Being a subtle but powerful generalization of CFGs, MCFGs have a familiar presentation that makes them easy to reason about, while remaining computationally tractable (Ljunglöf, 2012; Kallmeyer, 2010). At the same time, they offer an appealing dissociation between abstract and surface syntax and lexical choice. A derivation inspected purely on the level of rule type signatures takes the form of an abstract syntax tree that is reminiscent of a traditional CFG parse. Normalizing an MCFG so as to disallow rules from freely inserting constant strings (i.e. wrapping all constants under a non-terminal)

[1] https://github.com/gijswijnholds/discontinuous-probing.

[2] Theoretical analyses of cross-serial dependencies can be found in various mildly context-sensitive frameworks, including CCG (Steedman, 1985), Multimodal Typelogical Grammar (Moortgat, 1999), the Discontinuous Lambek Calculus (Morrill et al., 2007) and others (Muskens, 2007; Koopman and Szabolcsi, 2000).

allows us to (i) trace back all substrings of the final yield to a single non-terminal and (ii) provide a clear computational interpretation that casts an MCFG as a linear type system, and its derivation as a functional program (De Groote and Pogodalla, 2003).

3 Methodology

3.1 Linguistic background

We focus on two patterns in Dutch: control verb nesting and verb raising.

Control Verb Nesting Control verbs select a (referential) noun phrase and an infinitival complement which lacks an overt subject. This missing dependent (a so-called *understood* subject) can be traced back to a higher level of the syntax tree, materialising as a dependent of the matrix clause; from a semantic standpoint, it is implicitly carried over to the subordinate clause by the control verb. The choice of *which* of the (possibly many) dependents is carried over is purely lexical, and essentially determined by the choice of verb (Augustinus, 2015)[3]:

(1) a. de student belooft de docent te studeren
the student promises the teacher to study
'the student promises the teacher to study'

b. de docent vraagt de student te studeren
the teacher asks the student to study
'the teacher asks the student to study'

The two sentences of example (1) agree in their surface form, but differ in how the agent understood as 'studying' is selected; in (1a) it is the main clause subject ('promise' being a *subject control* verb), whereas in (1b) it is the main clause object ('ask' being an *object control* verb).

The basic constructions above can quickly become more nuanced in a variety of ways:

(2) a. de hond vraagt de student de oefeningen te
the dog asks the student the exercises to
eten
eat
'the dog asks the student to eat the exercises'

b. de docent vraagt de hond de student
the teacher asks the dog the student
de oefeningen te laten doen
the exercises to let do
'the teacher asks the dog to let the student do the exercises'

c. de docent vraagt de hond de student te
the teacher asks the dog the student to
beloven de oefeningen niet te eten
promise the exercises not to eat
'the teacher asks the dog to promise the student not to eat the exercises'

To begin with, if the head of the subordinate clause is a transitive infinitive, its object is positioned immediately after the main clause; this has the effect of creating a sequence of noun phrases that precede the verbal complement (2a). Further, in the case of the infinitive being a causative which selects for another infinitive, subject selection is preserved for the former, but flipped for the latter (2b).

Finally, things get interesting when realizing that the above patterns can recurse, as a verbal complement may act as the object of another verbal complement (2c).

The nesting of control verbs makes for a challenging probing task, as the dependency between a verb and its subject may span multiple depths of the syntax tree, while at the same time requiring subtle lexical distinctions to resolve correctly.

Verb Raising Dutch verb raising is the phenomenon whereby the head of an infinitival complement attaches to the verb governing it, creating a cluster in the process (Evers et al., 1976). Verbs allowing this construction select for bare complements (i.e. do not require the complementizer *te*). Unlike the previous case, the verbal complement does now contain a material subject; the complication is this time due to each nested verbal complement adding yet another set of crossing dependencies.

(3) a. de docent ziet de student de hond
the teacher sees the student the dog
de oefeningen leren eten
the exercises teach eat
'the teacher sees the student teach the dog to eat the exercises'

b. de docent ziet de hond de student de eend
the teacher sees the dog the student the duck
de oefeningen helpen leren eten
the exercises help teach eat
'the teacher sees the dog help the student teach the duck to eat the exercises'

By construction, the verb raising grammar isolates the problem of resolving verb-subject dependencies in a purely syntactic setting, as no lexical variation will change the choice of dependent for a given verb. As such, it allows us to probe for a model's potential at syntactic generalization that does no longer rely on lexical cues.

[3] Some of the verbs that we select are optional clustering verbs, but we use them only in the control setting.

$$
\begin{array}{rcll}
\textsc{s}(xyzu_1u_2) & \longleftarrow & \textsc{np}(x)\ \textsc{tv}(y)\ \textsc{np}(z)\ \textsc{vc}(u_1,u_2) & (A_1) \\
\textsc{s}(xyzuw_1vw_2) & \longleftarrow & \textsc{np}(x)\ \textsc{tv}(y)\ \textsc{np}(z)\ \textsc{np}(u)\ \textsc{cv}(v)\ \textsc{vc}(w_1,w_2) & (A_2) \\
\textsc{vc}(x,y) & \longleftarrow & \textsc{te}(x)\ \textsc{inf}_{iv}(y) & (A_3) \\
\textsc{vc}(zx,y) & \longleftarrow & \textsc{te}(x)\ \textsc{inf}_{tv}(y)\ \textsc{np}(z) & (A_4) \\
\textsc{vc}(xy,zu_0u_1) & \longleftarrow & \textsc{np}(x)\ \textsc{te}(y)\ \textsc{inf}_{c}(z)\ \textsc{vc}(u_0,u_1) & (A_5) \\
\textsc{vc}(xyu,zv_1v_2) & \longleftarrow & \textsc{np}(x)\ \textsc{te}(y)\ \textsc{inf}_{c}(z)\ \textsc{cv}(u)\ \textsc{vc}(v_1,v_2) & (A_6) \\
\hline
\textsc{s}(xyzvu_1u_2) & \longleftarrow & \textsc{np}(x)\ \textsc{tv}(y)\ \textsc{np}(z)\ \textsc{vc}(u_1,u_2)\ \textsc{adv}(v) & (A_1^m) \\
\textsc{s}(vyxzu_1u_2) & \longleftarrow & \textsc{np}(x)\ \textsc{tv}(y)\ \textsc{np}(z)\ \textsc{vc}(u_1,u_2)\ \textsc{adv}(v) & (A_1^i) \\
 & \vdots & &
\end{array}
$$

(a) 2-MCFG for control verbs.

$$
\begin{array}{rcll}
\textsc{s}(xy_1y_2) & \longleftarrow & \textsc{pref}(x)\ \textsc{sub}(y_1,y_2) & (B_1) \\
\textsc{sub}(x,y) & \longleftarrow & \textsc{np}(x)\ \textsc{inf}_{iv}(y) & (B_2) \\
\textsc{sub}(xy,z) & \longleftarrow & \textsc{np}(x)\ \textsc{np}(y)\ \textsc{inf}_{tv}(z) & (B_3) \\
\textsc{sub}(xz,yu) & \longleftarrow & \textsc{np}(x)\ \textsc{rv}(y)\ \textsc{sub}(z,u) & (B_4)
\end{array}
$$

(b) 2-MCFG for verb raising.

Figure 2: 2-MCFGs capturing the phenomena of Section 3.1.

3.2 Data generation

For our data generation needs, we design a custom implementation of an MCFG enriched with two added functionalities. First, we define two sets $\mathcal{N}_v$, $\mathcal{N}_n \subset \mathcal{N}$ that specify which non-terminals correspond to verb- and noun phrases respectively. Every occurrence of a marked non-terminal indicates a unique phrase in the final yield, which we can trace by traversing the derivation tree. This, in turn, gives us the possibility of assigning one or more labels to the constituent substrings that make up a sentence, according to which phrase(s) they were part of, even in the case of discontinuous and/or overlapping substrings. Additionally, we decorate MCFG rules with subject inheritance schemes. In the simplest case, a scheme may directly specify the subject noun of a verb, if the non-terminals of both occur on the same rule, i.e. they inhabit the same depth of the generation tree. Alternatively, when the two occur at different depths, a scheme may defer the decision by propagating verb indices down through non-nominal constituents that will contain the matching subject, but at an arbitrary nesting depth (see Figure 3 for an example). Lexical constants for primitive categories are populated by means of an automatically compiled but manually verified lexicon.

3.3 Grammars

We use the above framework to instantiate distinct grammars for both syntactic phenomena of interest. Note that the grammars are not purposed for the construction of exhaustive or accurate analyses of the phrase structures considered, but rather for the controlled generation and annotation of suitable samples.

Control Verb grammar Our first grammar, given in Figure 2a, models control verb nesting. The grammar accounts for the mobility of verbal complements by encoding them as non-terminals of multiplicity 2, making the grammar a 2-MCFG. We have two constructors for sentences that combine two noun phrases and a transitive verb with a verbal complement (A_1), optionally under the context of a causative verb and its direct object (A_2). In the base case, verbal complements are constructed with *te* and either an intransitive infinitive (A_3) or a transitive infinitive and its object (A_4). In the inductive case, a verbal complement can contain a control verb in infinitival form together with a noun phrase and another verbal complement, either alone or with a causative (A_5 and A_6). To increase the variance of generated samples, we also consider two variations for each of the first two rules

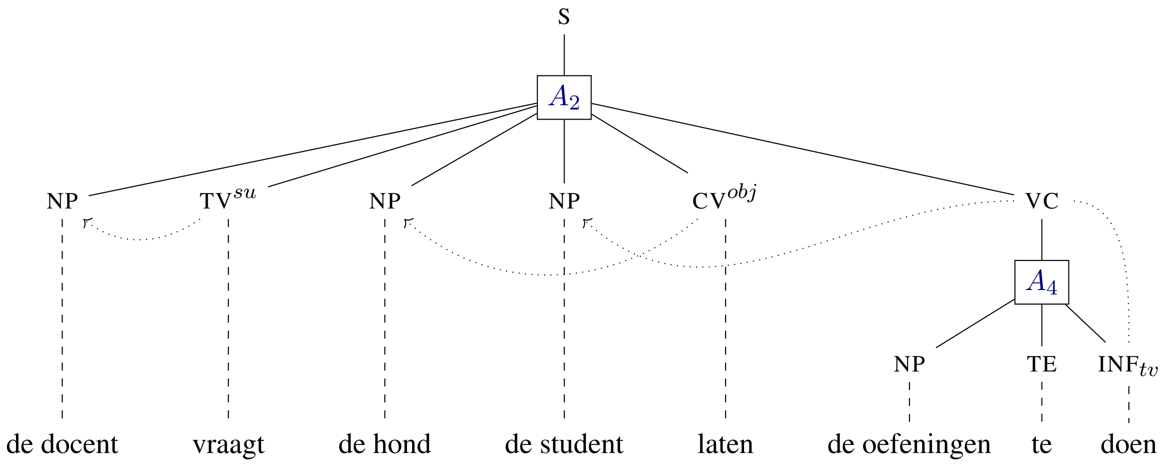

Figure 3: Generation tree for example (2b). Boxed nodes correspond to rule applications. Non-terminal superscripts denote verbal subtype (subject- or object control). Dashed lines assign lexical constants to non-terminals. Dotted lines demonstrate how verbs select for their subjects: TV^{su} and CV^{obj} both find their subjects at the same depth of the tree, but the presence of the latter signifies that the main clause object will be propagated to the verbal complement, to be there selected by INF_{tv}. Note that the tree presented should not be confused for a constituency parse – a more fitting paradigm would be an abstract syntax tree, that prescribes the program $A_2\left(\text{NP}(\text{de docent}), \text{TV}(\text{vraagt}), \text{NP}(\text{de hond}), \text{NP}(\text{de student}), \text{CV}(\text{laten}), A_4\left(\text{NP}(\text{de oefeningen}), \text{TE}(\text{te}), \text{INF}_{tv}(\text{doen})\right)\right) \mapsto$ (2b).

that incorporate adverbial modifiers: one where the adverb is inserted after the verb (A_1^m) and, more interestingly, one where the adverb is inserted before the verb (A_1^i); Dutch being a V2 language, this has the effect of inverting the position of the verb and subject of the main clause.

We set $\mathcal{N}_v := \{\text{TV}, \text{MV}, \text{INF}_x\}$ and $\mathcal{N}_n := \{\text{NP}\}$. We divide each of TV MV and INF_c into two subtypes, specifying whether they are subject- or object-selecting; each subtype has a distinct set of lexical entries. Finally we decorate each rule with subject propagation schemes, dependent on the subtypes of the participating verbal non-terminals; rather than explicitly enumerate these schemes here, we provide a visual example in Figure 3.

Verb Raising grammar For the second grammar we can do with just four rules (Figure 2b). The grammar is centered around a single non-terminal of multiplicity 2 that encodes subordinate clauses. In the base case, such a clause can be constructed with the aid of either a noun-phrase and an intransitive infinitive (B_2), or two noun phrases and a transitive (B_3). In the inductive case, a subordinate clause is embedded within a broader subordinate clause, where it occupies the object position of a raising verb (B_4). Finally, a sentence is generated by joining a subordinate clause to a matrix clause missing its verbal complement – we avoid deconstructing the matrix clause and denote it as a fixed prefix string (B_1). We set $\mathcal{N}_v := \{\text{INF}_{iv}, \text{INF}_{tv}, \text{RV}\}$ and $\mathcal{N}_n := \{\text{NP}\}$. Unlike in the case of control verb nesting, there is no subject inheritance necessary; rules B_2, B_3, and B_4 all add a verb and their subject simultaneously.

3.4 Probing Model

Our probing model first aggregates the contextualized token representations for each verb- and noun-phrase, before computing a verb-to-noun cross-attention matrix.

The aggregation process is essentially an attentive pooling over (two types of) variably sized, potentially overlapping clusters (Li et al., 2015). We start by representing each distinct verb- and noun-phrase as a binary mask over the tokenized input sentence; each sentence is then associated with a variable number of both types of masks. Using a pair of learned projections, we map the BERT-contextualized token representations into scalar values denoting attention scores. For each phrase, attention weights for participating (potentially discontinuous) tokens are computed by softmaxing their corresponding attention scores; summing the attention-weighted BERT representations yields a single vector for each phrase. We use the implementation of Fey and Lenssen (2019) to efficiently compute batch-wide representations leveraging the sparsity of the phrasal masks.

The pair-wise agreement between verb and noun representations is computed using standard dot-product attention, restricted to pairs occurring in the same sentence via dynamic masking. Prior to computing this attention matrix, we map the verb and noun representations to a lower dimension using another pair of learned projections; this serves to add a hint of expressive capacity to the probe, while also reducing the memory footprint of the matrix multiplication. Finally, softmaxing the attention weights over the noun-dimension allows us to retrieve a trainable subject selection for each occurrence of a verb.

4 Experiments & Results

4.1 Experimental setup

The experiments with our grammars consist of several parts. We first carry out an automatic filtering and annotation process on a gold standard corpus to gather a collection of suitable sentences, with which we train our probe on a natural, "real-world" dataset. To obtain our datasets, we start by fixing a maximal recursion depth for each grammar, and exhaustively generate the corresponding sets of derivable abstract trees. We then semi-automatically assemble a lexicon, with which we populate the various primitive categories employed by our grammars. From each tree, we obtain a set number of unique sentences by randomly sampling the constants behind leaf non-terminals with a preset seed. Finally, we apply the trained probe on the artificial samples and measure its performance across various generation parameters.

Probe Training An inevitable downside of using a rule-based system for generation purposes is low variance in several aspects of the output data. In our case, the limited number of rules employed, in combination with their relative simplicity, would mean a fair amount of repeating patterns that are easy to decipher and memoize. Albeit an advantage for interpretability and analysis purposes, this can potentially backfire if we are to use our grammars' yield for training: one can assume that BERT's contextualization preserves, at least in part, the relative position information contained within its input, thus providing the probe with a workaround (or confound) to the actual problem. To avoid overfitting, we consequently choose to train the probe on an external data source derived from Lassy-Small, the gold standard corpus of written Dutch (Van Noord et al., 2013). Lassy makes for an excellent data source for our task, as it provides analyses in the form of graphs, rather than trees, so as to explicitly account for several non-local phenomena (crucially, this includes the semantic subjects of verbal complements). We traverse the Lassy graphs to annotate noun phrases (all leaf nodes that descent from a noun phrase or are otherwise marked as a noun or pronoun) and verbs of interest (phrasal heads within a dependency frame that contains a subject previously identified as a noun phrase). From the 65 000 samples of Lassy, we extract about 12 000 that contain at least two *distinct* subjects without exceeding a word length of 30. We split the latter into two mutually exclusive sets of 10 000 and 2 000 samples: we train with the first and use the second for model selection.

We experiment with two Dutch language models: BERTje (de Vries et al., 2019) and RobBERT (Delobelle et al., 2020), based on BERT and RoBERTa (Liu et al., 2019) respectively. BERTje and RobBERT have shown to perform highly on a variety of Dutch NLP tasks, such as Named Entity Recognition (Sang and De Meulder, 2003), Sentiment Analysis (van der Burgh and Verberne, 2019), and Natural Language Inference (Wijnholds and Moortgat, 2021). For each model, we train 3 probes that differ in their initialization seeds, using AdamW (Loshchilov and Hutter, 2018) with a learning rate of 10^{-4}, a batch size of 32 and a dropout rate of 15%, applied at BERT's output. We perform model selection using accuracy over the validation set as our metric, measured over individual verb predictions; validation accuracy converges after ca. 80 training epochs.

Controlled data generation Despite remaining grammatical, sentences start looking odd and unnatural when allowing recursion to arbitrary depth – we impose an upper limit that leads to complex but still human-parsable data: 4 and 6 for the verbal control and raising grammars, respectively. To cast the generated trees into sentences, we populate primitive categories (that is, categories that can be instantiated lexically rather than – or in addition to – by rule) with sets of semi-automatically assembled constants. For simplicity, we consider only the case of verbs accepting a person as their indirect object; we filter 40 such verbs from a larger collection of ditransitives crawled from Lassy, and manually gather 30 temporal, locative and manner adverbs that can modify them. All verbs are drawn from Lassy (Van Noord et al., 2013) and the lists

	# Nouns				Tree Depth			Rule					
Model	2	3	4	5	2	3	4	A_1^X	A_2^X	A_3	A_4	A_5	A_6
BERTje	81.1	58.8	50.5	42.9	61.8	52.7	46.8	100	67	43.1	34.6	36.1	27.1
RobBERT	73	52.8	42.4	35.9	58.3	47.2	38.8	93.6	58.1	41.2	19.6	21	17

(a) Control Verb Grammar

	# Nouns				Tree Depth					Rule		
Model	2	3	4	5	2	3	4	5	6	B_2	B_3	B_4
BERTje	75.6	52.4	33.5	25.5	92.2	66.4	40.5	29	23	53.4	53.8	36.7
RobBERT	46.3	37.2	24.5	11.4	65.6	36.9	33.6	19.4	12.6	89.1	24.3	12.9

(b) Verb Raising Grammar

Table 1: Seed-averaged accuracy scores for the two grammars of Section 3.3, grouped by various parameters. The X superscript denotes inclusion and aggregation of the adverbial modifier variants for the corresponding rules.

of Augustinus (2015): we gather 9 subject- and 33 object-control verbs, 2 causatives and 7 raising verbs. A comprehensive set of around 10 000 gendered nouns (the ones that have *de* as their article) is finally obtained from the *Algemeen Nederlands Woordenboek*.[4] In the verb raising grammar, we trigger subordination by prefixing generated expressions with the string *Iemand ziet* ('somebody sees').

From each generated abstract tree, we obtain 10 syntactically identical sentences that vary only in their meaning by performing controlled sampling over the lexicon; the very large product space of constants guarantees sample uniqueness. This parameterization means we can inspect and group samples on the basis of either their surface form or their underlying tree, a property that will come when analyzing model performance. To ensure naturality and consistency in the model's input, we capitalize and punctuate generated sentences in a final post-processing step.

4.2 Results

The trained probes are tested on our generated data, yielding a prediction for every verb occurrence. For each model, we report the seed-averaged accuracy on each experiment in Table 2: test performance is substantially lower than in the validation benchmarks.

Model	Lassy	Control	Raising
BERTje	97.6	48	43.1
RobBERT	92.5	40.6	29.2

Table 2: Model accuracy on the validation data (Lassy) versus the test data (Control, Raising).

To facilitate analysis, we group predictions according to their context, namely (i) number of noun phrases in the sentence (classification targets), (ii) maximal depth of the underlying abstract syntax tree and (iii) production rule, and aggregate them into accuracy scores, presented in Table 1. This breakdown suggests that model performance remains passable for the easier portion of the dataset, but degrades quickly as the difficulty of the task increases; models have a harder time associating a verb to its subject as sentences get longer and more complicated. The over-representation of harder-samples due to the dominance of deeper abstract syntax trees then serves to explain the striking performance decline.

Control Verbs Focusing on the control grammar first, we remark that both models consistently score above the random baseline (i.e. 1 divided by the number of classification targets), seemingly indicating that some notion of semantic comprehension perseveres in the presence of control verb nestings. Grouping scores by rule is revealing: the main clause subject is (almost) always correctly detected, regardless of nestedness of the co-occurring complement and unperturbed by the presence of word-order variations due to modifiers (A_1^X). Verbal complements and causatives, on the other hand, are more often than not incorrectly analyzed, even in the simplest cases of a bare infinitive in isolation (A_3), or a causative occurring at the topmost branch of the tree (A_2^X).

To procure an explanation for this discrepancy, we start by measuring accuracy in verbs occurring under subject- and object control scopes separately. The remarkably low results hint that models struggle with both kinds of control, while indicating the presence of an implicit bias slightly favouring

[4] https://anw.ivdnt.org

Model	Control Scope		Consistency
	subject	object	
BERTje	34.4	36.1	68.4
RobBERT	18.7	37.8	63

Table 3: Metrics specific to the Control Verb grammar.

the more common object control reading (especially so in the case of RobBERT). Next, we investigate whether the low performance is due to models simply misreading certain constructions, assigning subjecthood to the (same) wrong noun phrase. To quantify how consistent the models are, we gather all predictions occurring in the same context (i.e. same part of the same tree under the same scope, object or subject) and varying only in terms of lexical realization. The consistency of a model in a specific context is calculated as the frequency of the most common prediction (correct or otherwise); the model's overall consistency is then the average consistency over all contexts. Models generally fail at producing the same prediction given the same syntactic template, instead being susceptible to distraction from word variations.

Verb Raising The story is no different when it comes to the second grammar: both models fail to draw close to their validation benchmarks. Surprisingly, RobBERT's metrics lie below the random baseline, positing that it encodes a wrong syntactic structure in verb cluster formations, rather than simply not acquiring the correct one. The disproportionately high accuracy of rule B_2 readily provides an explanation: the noun-phrase directly preceding an infinitive is assumed to be its subject. BERTje, on the other hand, is more trustworthy, maintaining comparable performance in both intransitive (B_2) and transitive (B_3) infinitival phrases. The degradation associated with deeper trees coincides with the drop in performance for the recursive rule B_4.

4.3 One-Shot Learning

Given the purported inadequacy of both models at correctly or consistently predicting subjecthood in our datasets' cross-serial constructions, we resort to one final experiment that serves as a sanity check for the quality of our data. Using a different lexical sampling seed, we generate a single sentence from each abstract syntax tree, resulting in datasets of 307 and 30 samples for the control verb and verb raising grammars, respectively. These compact datasets are then used for fine-tuning the two models (combined with probes) in a one-shot learning fashion; after a few epochs of training, we test the resulting models on the corresponding original datasets.

Model	Control	Raising
BERTje	92.4	68.5
RobBERT	61.6	36.7

Table 4: Model results for the one-shot setup.

The results, presented in Table 4, show that minimal supervision does improve model performance, indicating that the learned parameter updates generalize beyond the lexical choices of the fine-tuning data, thereby verifying the generation pipeline's internal consistency. Improvement is lower in the case of the verb raising grammar; we posit that the task is harder to acquire due to its predominantly syntactic nature but also the smaller number of training samples.

5 Conclusion

We implemented a test suite based on multiple context-free grammars to generate a large collection of sentences containing complicated syntactic phenomena specific to Dutch. We trained a probing model on extracting verb-to-subject pairings from the contextualized representations of state-of-the-art pretrained Dutch language models using an external resource of generic text accompanied by gold standard annotations. We then tested the probe on our generated data, and found it to perform substantially below its own validation benchmarks. After conducting extensive analysis aimed at identifying the source of this discrepancy, we showed that the probe's predictions are inconsistent and its accuracy quickly diminishes as the complexity of the syntactic patterns increases. Based on the above, we conclude that neither of the BERT models investigated has learned to internalize syntactic and semantic subjecthood in nested constructions involving cross-serial dependencies. Our findings serve as empirical evidence hinting at unsupervised language models having difficulty in the automatic acquisition of discontinuous syntactic patterns.

We leave several directions open for future work. To begin with, one could mirror the patterns analyzed to other languages and compare model performance cross-linguistically, juxtaposed by the corresponding grammar complexity. Alternatively, one could render more elaborate grammars intended to capture other syntactic or semantic phenomena of interest. Finally, it is worth investigating the extent to which the "real-world" validation samples incorrectly classified are exemplars of the types of discontinuity captured by our grammars.

Acknowledgments

The authors are grateful to the anonymous reviewers for their feedback. The authors moreover thank Michael Moortgat for review of the paper before submission, and for valuable discussions. Both authors acknowledge support from the Dutch Research Council (NWO) under the scope of the project "A composition calculus for vector-based semantic modelling with a localization for Dutch" (360-89-070).

References

Liesbeth Augustinus. 2015. *Complement raising and cluster formation in Dutch*. Netherlands Graduate School of Linguistics.

Joan Bresnan, Ronald M. Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13(4):613–635.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing {bert} in hyperbolic spaces. In *International Conference on Learning Representations*.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Philippe De Groote and Sylvain Pogodalla. 2003. m-linear context-free rewriting systems as abstract categorial grammars. In *Proceedings of Eighth Meeting on Mathematics of Language (MOL 8)*, pages 71–80.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT model. *arXiv preprint arXiv:1912.09582*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch roBERTa-based language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3255–3265.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Arnold Evers et al. 1976. The transformational cycle in Dutch and German. *Nieuwe (De) Taalgids*, 69(2):156–160.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Riny Huybregts. 1984. The weak inadequacy of context-free phrase structure grammars. *Van periferie naar kern*, pages 81–99.

Aravind Krishna Joshi. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?

Laura Kallmeyer. 2010. *Parsing beyond context-free grammars*. Springer Science & Business Media.

Dan Klein and Christopher D Manning. 2001. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 338–345.

Hilda Judith Koopman and Anna Szabolcsi. 2000. *Verbal complexes*. 34. MIT Press.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Yongjie Lin Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Peter Ljunglöf. 2012. Practical parsing of parallel multiple context-free grammars. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, pages 144–152, Paris, France.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michael Moortgat. 1999. Meaningful patterns. *Linguistics*, 210:4.

Glyn Morrill, Oriol Valentín, and Mario Fadda. 2007. Dutch grammar and processing: A case study in TLG. In *International Tbilisi Symposium on Logic, Language, and Computation*, pages 272–286. Springer.

Reinhard Muskens. 2007. Separating syntax and combinatorics in categorial grammar. *Research on language and computation*, 5(3):267–285.

Geoffrey K Pullum and Gerald Gazdar. 1982. Natural languages and context-free languages. *Linguistics and Philosophy*, 4(4):471–504.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229.

Stuart M Shieber. 1985. Evidence against the context-freeness of natural language. In *Philosophy, language, and artificial intelligence*, pages 79–89. Springer.

Mark Steedman. 1985. Dependency and coördination in the grammar of dutch and english. *Language*, pages 523–568.

Benjamin van der Burgh and Suzan Verberne. 2019. The merits of universal language model fine-tuning for small datasets – a case with dutch book reviews.

Gertjan Van Noord, Gosse Bouma, Frank Van Eynde, Daniel 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 *Essential speech and language technology for Dutch*, pages 147–164. Springer, Berlin, Heidelberg.

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. 2020. Parsing as pretraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9114–9121.

Gijs Wijnholds and Michael Moortgat. 2021. SICK-NL: A dataset for Dutch natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1474–1479, Online. Association for Computational Linguistics.