Computational Linguistics in the Netherlands Journal 12 (2022) 49-63 Submitted 12/2022; Published 12/2022

Noun Phrase and Verb Phrase Ellipsis in Dutch: Identifying Subject-Verb Dependencies with BERTje

Tessel Haagen* T.E.HAAGEN@STUDENTS.UU.NL
Loïs Dona* L.M.DONA@STUDENTS.UU.NL
Sarah Bosscha* S.A.M.BOSSCHA@STUDENTS.UU.NL
Beatriz Zamith* M.B.ZAMITHCASTRO@STUDENTS.UU.NL
Richard Koetschruyter* R.A.KOETSCHRUYTER@STUDENTS.UU.NL
Gijs Wijnholds* G.J.WIJNHOLDS@UU.NL

* *Utrecht University, Utrecht, Netherlands*

Abstract

Previous research has set out to quantify the syntactic capacity of BERTje (the Dutch equivalent of BERT) in the context of phenomena such as control verb nesting and verb raising in Dutch. Another complex language phenomenon is ellipsis, where a constituent is omitted from a sentence and can be recovered using context. Like verb raising and control verb nesting, ellipsis is suitable for evaluating BERTje's linguistic capacity since it requires the processing of syntactic and lexical cues to recover the elided phrases. This work outlines an approach to identify subject-verb dependencies in Dutch sentences with verb phrase and noun phrase ellipsis using BERTje. Results will inform about BERTje's capability of capturing syntactic information and its ability to capture ellipsis in particular. Understanding more about how computational models process ellipsis and how it can be improved is crucial for boosting the performance of language models, as natural language contains many instances of ellipsis. Using training data from Lassy, converted to contextualized embeddings using BERTje, a probe model is trained to identify subject-verb dependencies. The model is tested on sentences generated using a Context Free Grammar (CFG), which is designed to generate sentences containing ellipsis. These sentences are also converted to contextualized representations using BERTje. Results show that BERTje's syntactic abilities are lacking, shown by accuracy drops compared to baseline measures.

1. Introduction

Ellipsis entails that a constituent has been omitted from a clause or phrase and can be recovered using the remaining context. The task of understanding an elliptical construction can be phrased as identifying which elided constituent and which overt constituent should be connected. This research focuses on verb phrases and noun phrase ellipsis (explained in more detail in Section 2.1).

Resolving these elliptical constructions remains a task that is very challenging to define computationally, even though it is trivial for humans. Many researchers have been concerned with formalizing the task of ellipsis detection and resolution computationally, resulting in models aimed explicitly at ellipsis resolution (McShane and Babkin 2016, Lin et al. 2019, Hardt 1997, Aralikatte et al. 2019, Lappin 2005, Nielsen 2004). However, it would be more valuable if general language models could identify and resolve them, taking us a step closer to models with a human level of language understanding. This requires knowledge about how language models process ellipsis, in this case, BERTje (de Vries et al. 2019). This research aims at gaining more insight into the syntactic capacity of BERTje in the context of ellipsis.

Previous research has been done concerning evaluating the syntactic capacity of BERT in the context of control verb nesting, and verb raising in Dutch (Kogkalidis and Wijnholds 2022). It was found that BERT struggled with capturing subject-verb dependencies for these kinds of con-

structions. It is expected that elliptical constructions are also tricky for BERT to capture, as researchers agree it is a very complex phenomenon to detect and resolve computationally (McShane and Babkin 2016, Hardt 1997, Nielsen 2004).

The pipeline of our work contains several steps: first, a Context Free Grammar (CFG) is defined to generate (Dutch) sentences containing ellipsis, having separate grammars for verb and noun phrase ellipsis, respectively. Following, a BERTje model (de Vries et al. 2019) is used to generate contextualized representations for the sentences coming from the grammar. These form the input for a probing model that has been trained on sentences from the Lassy Small corpus (van Noord et al. 2013), which allows us to test the inherent capacity of the contextualized embeddings to recognize ellipsis.

This paper is structured as follows: in section 2 we discuss the background of our work, followed by section 3 detailing the implementation details. Section 4 contains the experimental results, analysis, and discussion. We conclude in section 5 and end this section with some directions for further work.

2. Theoretical Background

2.1 Ellipsis

One of the most common linguistic phenomena, while simultaneously one of the most complex linguistic occurrences to trace computationally, is formally known as ellipsis (McShane et al. 2005). One refers to a construction as being elliptical when a phrase or group of phrases is deliberately left out of a sentence without changing its original meaning. They are occasionally marked by "...". While there are many different types of ellipsis, this study focuses on verb phrase and noun phrase ellipsis, since those are the most common types of ellipsis. In this work, the noun phrase ellipsis cases we consider only involve the omission of a subject. There are crosslinguistic differences concerning ellipsis. For example, German (Merchant 2005), Spanish, and French do not demonstrate verb phrase ellipsis, while English and Dutch do (Cyrino and Matos 2005).

2.1.1 Verb Phrase Ellipsis

A verb phrase ellipsis is a linguistic phenomenon in which the verb phrase (main predicate) of a sentence (clause), often in combination with its internal argument, is omitted and can be retrieved from context (Van Craenenbroeck 2017).

(1) Sommige mensen gaan naar het park, andere mensen ... naar het strand en sommige
Some people go to the park, other people ... to the beach and some
mensen houden van de zon en andere mensen ... van de regen
people love of the sun, and other people ... of the rain
'Some people go to the park, other people ... to the beach and some people love the sun and other people ... the rain'

(2) James lachte en Marie ... ook, maar John huilde en Sarah ... ook.
James laughed and Marie ... too but John cried and Sarah ... too
'James laughed and Marie ... too, but John cried and Sarah ... too'

In the two examples above, the omitted phrases are "gaan", "houden", "lachte" and ''huilde" respectively. In both cases, it is possible to recover the elided constituents by looking at the antecedent verb phrases.

2.1.2 Noun Phrase Ellipsis

A noun phrase ellipsis is a similar phenomenon in which the noun phrase of a sentence is omitted and can be understood from the context.

(3) Emma zag drie vogels in de lucht en ... ving er twee
Emma saw three birds in the sky and ... caught of them two
'Emma saw three birds in the sky and caught two'

In the example above, the omitted noun phrase is "Emma". Again, it is possible to recover the semantics of the sentences and the elided phrases by looking at the antecedent noun phrases.

2.2 Context-Free Grammars

In order to be able to specifically target these cases of ellipsis as a phenomenon to probe a neural language model for, we generate elliptical sentences using a context free grammar (CFG). The decision has been made to generate elliptical sentences using a CFG and not extract a test set of elliptical sentences from the Lassy corpus because the latter would not leave enough data to train the probe model. A CFG is a formal grammar consisting of a set of rewrite rules that can describe context-free languages. A generalization of context free grammars is Multiple Context Free Grammars (MCFG) (Seki et al. 1991) in which production rules can range over tuples of strings rather than single strings in the case of a CFG. Such grammars can analyze non-context-free languages like $\{a^n b^n c^n \mid n \geq 1\}$. This paper is concerned with the Dutch language, which has been shown not to be a context-free language (Bresnan et al. 1987). This means that sentences generated by a CFG cannot encapsulate all phenomena in the Dutch language but instead contain a minimal subset that suits our purposes. As elaborated in Kogkalidis and Wijnholds (2022), the usage of CFGs proves advantageous. Firstly, it allows for reasoning to take place relatively easily while being simultaneously computationally manageable. Second, it allows for a clear distinction between abstract and surface syntax and lexical choice. Their approach used an MCFG (Multiple Context Free Grammar) which can model more phenomena, such as cross-serial dependencies, which is not needed for our purposes.

2.3 Probe Tasks

A probe model is used to target the generated ellipses as a phenomenon. Probing tasks are models for gaining more insight into the ability to capture certain kinds of linguistic information (Tenney et al. 2018). These tasks are designed to target specific linguistic phenomena; if a model successfully performs the task, it can be concluded that they have encoded the phenomenon of interest. Machine learning approaches for natural language processing tasks have gained much popularity over the last few years. Part of the research in this domain is to investigate what linguistic information these language models can encode. Due to the black-box nature of many machine learning approaches, gaining insight into this (Conneau et al. 2018) is challenging. In this research, the probing task consists of finding the correct subject-verb dependencies in a set of Dutch sentences to evaluate the syntactic capacity of BERTje.

2.4 BERT

The probe model uses encoded phrases by BERT. BERT (Devlin et al. 2018) is a unique machine learning framework for Natural Language Processing where phrases are defined by the surrounding phrases and not by pre-fixed identity. BERT was trained on 3.3 billion tokens of unlabeled text to predict identities of words that have been masked-oud of the input text. Next, the BERT model predicts if the second half of the input follows the first half in the corpus or is an incidental individual text segment. (Clark et al. 2019) BERT helps computers understand the semantics (meaning)

of ambiguous language in the text by using surrounding text to establish context. BERT is an abbreviation for Bidirectional Encoder Representations from Transformers; as the name tells, the model is based on the transformer architecture (Vaswani et al. 2017). Transformer architectures are deep learning models where every output element is connected to every input element, and the weightings are dynamically calculated based on their connection. This research uses BERTje; it has the same architecture and parameters as BERT but is trained in Dutch. BERTje is based on a large and diverse dataset of 2.4 billion tokens (de Vries et al. 2019). In this research, the language model will encode sentences passed on to it from our model into contextualized vector representations. The lexicon randomly generates sentences with our grammar, and all the phrases are turned into vectors by BERTje. These vector representations include encoded phrases, e.g. subjects, verbs, and noun phrases.

2.5 Language Models and Ellipsis

2.5.1 Evaluating the Syntactic Capacity of BERTje

Kogkalidis and Wijnholds (2022) constructed MCFGs to generate sentences in Dutch containing control verb nesting and verb raising, encoding subject-verb dependencies. This has been used to test a probe model trained to identify these subject-verb dependencies using Lassy Small, a gold standard natural corpus of written Dutch (van Noord et al. 2013). In addition, two Dutch versions of BERT, BERTje (de Vries et al. 2019), and RobBERT (Delobelle et al. 2020), were used to convert sentences to their corresponding contextualized representations before feeding them to the probe model.

Control verbs select a (referential) noun phrase, and an infinitival complement without an explicit subject in the surface form of the sentence (Augustinus 2015). This subject can be traced back to a higher level of the syntax tree. The choice of which of the dependents is the actual subject belonging to the infinitival complement is determined by the lexical choice of the verb. The nesting of control verbs is challenging to trace computationally, as the dependency between a verb and its subject may require traversing multiple depths of the syntax tree and depends on lexical information.

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). Unlike the previous case, the subject of the verbal complement now does show up in the surface form of the sentence. However, this time the complexity lies in the fact that each nested verbal complement adds another set of crossing dependency relationships, which is a purely syntactic challenge for the probe model.

Results of Kogkalidis and Wijnholds (2022) showed that the probe's predictions are inconsistent, and its accuracy quickly diminishes as the complexity of the syntactic patterns increases. BERTje had not learned to internalize syntactic and semantic cues defining subject-verb dependencies in the context of control verb nesting and verb raising.

2.5.2 The complex phenomenon ellipsis

As has been mentioned before, an ellipsis is a complex linguistic structure that can have varying levels of complexity (McShane and Babkin 2016).

This is illustrated in examples 4 and 5 below, where sentence 5 is more complex due to the more significant number of elliptical constituents. It is similar to the control verb nesting discussed above because the elided verb phrases can be traced back to a higher level of the syntax tree. The fact that subjects in different elliptical clauses inherit their verb from different non-elliptical clauses at different levels of the syntax tree also makes example 2 more complex.

(4) Jess kocht een fiets en Toby ... een auto en Jenny eet snoep
Jess bought a bike and Toby ... a car and Jenny eats candy
'Jess bought a bike and Toby ... a car and Jenny eats candy'

(5) Jess kocht een fiets en Toby ... een auto en Bill slaapt maar Harry ... niet en
Jess bought a bike and Toby ... a car and Bill sleeps but Harry ... not and
Jessica maakte een taart en Bea ... koekjes
Jessica made a cake and Bea ... cookies
'Jess bought a bike and Toby a car and Bill is sleeping but Harry isn't and Jessica made a cake and Bea cookies'

Although elliptical constructions are straightforward for humans to understand, this is not always the case for computational models, illustrated by various attempts at designing models that can detect and resolve ellipsis (McShane and Babkin 2016, Aralikatte et al. 2019, Lin et al. 2019, Lappin 2005). Although some cases of ellipsis are easily detected and resolved by computational models, the more complex cases still require more research to arrive at fully comprehensive models for ellipsis detection and resolution. McShane and Babkin (2016) designed ViPER (Verb Phrase Ellipsis Resolver), a language model aimed at detecting and resolving verb phrase ellipsis. They demonstrated that the verb phrase ellipsis has different difficulty levels, influencing the performance of ViPER. These specialized models for detecting and resolving ellipsis can be powerful in many cases. However, the disadvantage is the absence of a wide range of transferable tasks, unlike language models like BERT(je). Understanding more about how computational models, particularly BERTje, process ellipsis and how it can be improved is crucial for boosting the performance of language models, as natural language contains many instances of ellipsis. Wijnholds and Sadrzadeh (2019) have already demonstrated that including information about verb phrase ellipsis in sentence embeddings can outperform traditional embedding methods.

3. Method

This section outlines the methodology to evaluate BERTje's syntactic capacity. It starts with an explanation of the grammar and its implementation, followed by a short motivation of the lexicon. This is followed by an overview of the implementation of the probe model, which is trained to identify subject-verb dependencies.

3.1 Context Free Grammar

The context free grammars for generating sentences have been implemented separately for a noun phrase and verb phrase ellipsis. A separate generation of sentences with noun and verb phrase ellipsis is desired to ensure analysis of BERT in the context of noun phrase and verb phrase ellipsis can be done in isolation. This section starts by explaining the rules of the CFGs and then continues with an illustration of how the inheritance of nouns and verbs and subject-verb dependencies have been encoded. Tables 1 and 2 show the rules for the verb phrase and noun phrase ellipsis grammars. For clarity, we refer to items that make up a rule as phrases; for example, SUBJ, ES or DC in rule 1 in table 1 are all phrases. The term "terminal rule" will be used to refer to phrases that can only be filled by a lexical item (a terminal). For example, SUBJ (see table 1) is a terminal rule, because it can only be rewritten into a lexical item. We refer to other rules as non-terminal rules, these rules describe rewrites from non-terminal symbols to other non-terminal symbols (for example rule 1).

3.1.1 Verb Phrase Ellipsis

The top-level rule of the grammar (rules 1 to 6 in table 1) conjoins three clauses, with the first consisting of a subject, object, and verb. The second clause is an elliptical sentence, and the final clause is dependent. These three types of clauses are connected using conjunctions. The rules are designed in such a way that sentences with the same conjunction twice in a row are not possible, making the sentences appear more natural. Similar rules for constructions with intransitive verbs or constructions with auxiliary verbs and infinitives have also been implemented. It is noteworthy

that elliptical sentences containing an adverb (ADV) are only possible for intransitive verbs. When using an adverb in the case of other verbs, the object would also be elided, making it both a noun and verb phrase ellipsis. The decision has been made to separate noun and verb phrase ellipsis from each other to be able to test BERT on each of them in isolation; hence the ES_{ADV} is only used with intransitive verbs. As mentioned earlier, the subject of the elliptical sentence in the construction corresponds to the verb(s) of the preceding clause of this rule. How these connections have been encoded in this grammar is explained in section 3.1.3.

The next set of rules allows for recursively elongating sentences by adding dependent clauses. Three of these rules are terminal (rules 7 to 9 in table 1) and have a construction containing a subject and either a transitive verb with an object, an intransitive verb, or an auxiliary infinitive construction with an object. They form dependent clauses to fill the DC slot in the top-level rules mentioned in the previous paragraph. These rules have been added to the grammar to ensure that there are also verbs in a sentence that are not connected to any cases of ellipsis. This makes the task of connecting subjects to their corresponding verb a challenge for the probe model. Rule 8 introduces recursion by assigning the top level (S) rule discussed in the former paragraph to the dependent clause.

Rules 11 to 14 model the creation of verb phrase elliptical sentences. Rule 11 and 13 allow for recursively adding more elliptical sentences by defining an elliptical sentence as a conjunction of two elliptical sentences. Rule 12 consists of a subject and object, omitting the verb belonging to that subject. In rule 14, the elliptical sentence consists of a subject and an adverb, again omitting the verb. How the relationships between these elliptical sentences and their corresponding elided verb is encoded are explained in section 3.1.3.

Another grammar has been designed to generate baseline sentences, meaning sentences without any form of ellipsis. This has been done by taking the verb phrase ellipsis grammar as a starting point and adding the verb terminal markers back into the rules for making elliptical sentences. This means that the verbs that get substituted into the elliptical sentences do not always correspond to the verbs in the preceding main clause, since the lexical items chosen for the verb terminals are chosen randomly from the lexicon.

3.1.2 Noun Phrase Ellipsis

The top-level rules and the rules for making dependent clauses are the same for the grammar that generates sentences containing noun phrase ellipsis. The difference is in the rules that define elliptical sentences. The set of rules is listed in table 2.

As mentioned, there are rules allowing for recursively adding more elliptical sentences by defining an elliptical sentence as a conjunction of two elliptical sentences. However, the terminal rule for an elliptical sentence now elides the subject instead of the verb. This means separate rules for transitive, intransitive and auxiliary infinitive constructions are needed. For the transitive case, the elliptical sentences consist of the verb and object, whereas intransitive elliptical sentences consist of the verb only. Elliptical sentences with an auxiliary and infinitive have an order of auxiliary, object, and infinitive. How the relationships between these elliptical sentences and their corresponding elided subject are encoded is explained in section 3.1.3. Sentences have been constructed with a maximum tree depth of 3 for all grammars, not taking the terminal rules into consideration. To clarify, Figure 1 has a tree depth of two.

3.1.3 Encoding Subject-Verb Dependencies

Every rule contains a tuple encoding inheritance, which will help to infer the subject-verb dependencies. This mechanism is an annotation from which subject-verb dependencies can be inferred. Although more strategies for implementing inheritance relationships are possible, coupling this mechanism to the CFG as an annotation has been the chosen strategy, following the implementation of Kogkalidis and Wijnholds (2022). It signals which phrase should be passed on to another phrase in

	Rule
1	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ} \cdot \mathsf{CNJ} \cdot \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
2	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV} \cdot \mathsf{CNJ} \cdot \mathsf{ES}_{ADV} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
3	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF} \cdot \mathsf{CNJ} \cdot \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
4	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
5	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV} \cdot \mathsf{CNJ} \cdot \mathsf{ES}_{ADV}$
6	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
7	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ}$
8	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV}$
9	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF}$
10	$\mathsf{DC} \rightarrow \mathsf{S}$
11	$\mathsf{ES} \rightarrow \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
12	$\mathsf{ES} \rightarrow \mathsf{SUBJ} \cdot \mathsf{OBJ}$
13	$\mathsf{ES}_{ADV} \rightarrow \mathsf{ES}_{ADV} \cdot \mathsf{CNJ} \cdot \mathsf{ES}_{ADV}$
14	$\mathsf{ES}_{ADV} \rightarrow \mathsf{SUBJ} \cdot \mathsf{ADV}$

Table 1: CFG for verb phrase elliptical sentences

	Rule
1	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ} \cdot \mathsf{CNJ} \cdot \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
2	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV} \cdot \mathsf{CNJ} \cdot \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
3	$\mathsf{S} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF} \cdot \mathsf{CNJ} \cdot \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{DC}$
4	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
5	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
6	$\mathsf{S} \rightarrow \mathsf{DC} \cdot \mathsf{CNJ} \cdot \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
7	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{TV} \cdot \mathsf{OBJ}$
8	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{ITV}$
9	$\mathsf{DC} \rightarrow \mathsf{SUBJ} \cdot \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF}$
10	$\mathsf{DC} \rightarrow \mathsf{S}$
11	$\mathsf{ES} \rightarrow \mathsf{ES} \cdot \mathsf{CNJ} \cdot \mathsf{ES}$
12	$\mathsf{ES} \rightarrow \mathsf{VB}_{TV} \cdot \mathsf{OBJ}$
13	$\mathsf{ES} \rightarrow \mathsf{VB}_{ITV}$
14	$\mathsf{ES} \rightarrow \mathsf{VB}_{AUX} \cdot \mathsf{OBJ} \cdot \mathsf{INF}$

Table 2: CFG for noun phrase elliptical sentences

the same rule. We use this mechanism in the verb phrase ellipsis CFG to encode that an elliptical sentence should inherit a verb. In the case of the noun phrase ellipsis, the elliptical sentence should inherit a noun. The tuple has the length of the right-hand side of the rule, and each slot contains a list of indices of what to inherit. The value False is used if a specific phrase does not need to inherit anything. The value True is used in the inheritance tuple to indicate that the verb or subject that the elliptical sentences on the right-hand side need to inherit is not present in the current rule but should be derived from a higher-level rule. This is used in rule 9, where the elliptical sentences should inherit a verb or subject which are not present in the current rule but should be derived from one of the top-level rules 1 to 5. In the example in Figure 1, the elliptical sentence inherits the auxiliary verb and infinitive, indicated by the boldfaced text.

The last component of a rule is a dictionary to indicate the dependencies between subjects and verbs. For both the verb phrase and noun phrase ellipsis grammar, this dictionary has indices of verbs as keys and a list of subject indices as values, encoding which verbs belong to which subjects. The indices mean the position of the phrase on the right-hand side of a rule. If either the verb or subject is not present in a rule (it is elided), a component of the subject-verb relationship is missing. In that case, we use the value None to indicate the missing component in the dictionary. This missing component should be derived from the previously defined inheritance relationships. In Figure 1, the subject-verb dependencies are indicated by arrows. It can be seen that in this case, the ES_{ADV} rule does not contain the verb to form the subject-verb dependency so the value None will be used in the dictionary accordingly. Due to the inheritance encodings, the algorithm can find the correct verbs "wil" and "zien" corresponding to the subject "de spion".

3.2 Lexicon

The second part of the implementation relevant to mention is the lexicon. The lexicon used in this research contains a set of different phrases that can be assigned to specific phrasal types in the grammar [1]. During the construction of the lexicon, it has been made sure that only singular subjects and objects can occur, and transitive and intransitive verbs are all in third person singular form, so they match each other grammatically. Moreover, all subjects and objects are persons and verbs are all semantically plausible for persons in order to control for semantic implausibility being a potential cause for poor performance. It is noteworthy to mention that despite the subjects and objects in the lexicon being composed of two-word phrases, this does not pose an issue for the probe. This is due to the first part of the probe aggregating the embeddings of a noun or verb phrase into a single embedding, regardless of the number of words in the phrase.

1. https://osf.io/yuc9q/?view_only=fffb1719e9c7449bafc4e7cfbbe1c091

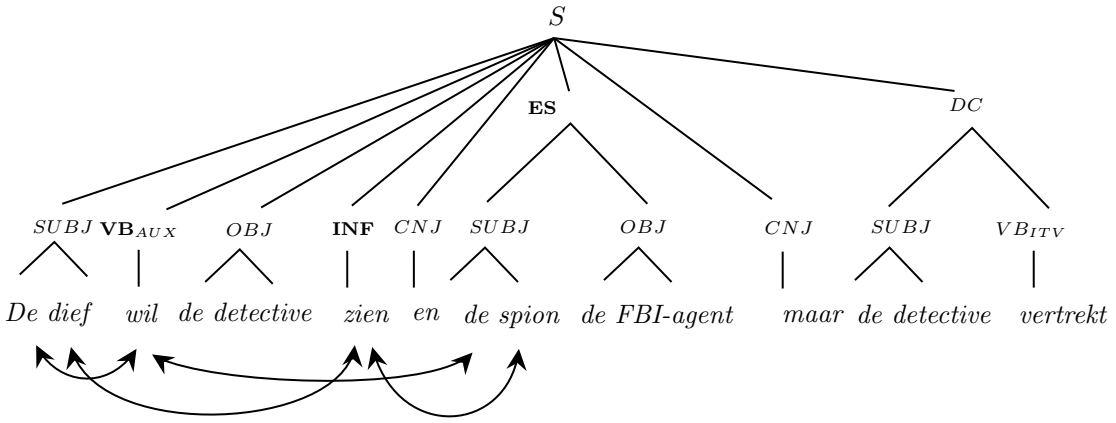

Figure 1: Generation tree example for verb phrase ellipsis. Subject-verb dependencies have been indicated by arrows and the inheritance relationships are bold.

3.3 BERTje and the Probing Model

For instantiating the probe model, we adapt the implementation of Kogkalidis and Wijnholds (2022). Similarly to the original probe model, sentences are vectorised sentences using BERTje, after which the pair-wise connections between verbs and nouns are computed to be optimised against the correct subject-verb matches. The difference in architecture is that the current probe allows for a many-to-many mapping between verbs and nouns, implemented by applying a Sigmoid function rather than a Softmax on the output of the probe.

The probe model is trained on Dutch sentences from the Lassy Small corpus, of which some contain ellipsis, and some do not. This is beneficial since training the model on only ellipsis sentences would create an unrealistic accuracy and overfitted model; the model could internalise the rule-based grammar from which the sentences originate.

The Dutch BERT model, BERTje, was used to convert sentences to their corresponding contextualised representations with embedding before feeding them to the probe model (Devlin et al. 2018). This is visualised in Figure 2.

As mentioned before, the probe model is trained to identify the subject-verb dependencies in verb phrase ellipsis and noun phrase ellipsis using training data from Lassy Small (van Noord et al. 2013). The training data consists of 6465043 sentences including 2957 elliptical sentences, where 2766 contain noun phrase ellipsis and 376 verb phrase ellipsis. After the training, the model is tested on sentences generated using the CFG (Chomsky 1959). Next, Kogkalidis and Wijnholds (2022) introduced a probe model trained to link verbs to their corresponding subject using a natural real-world data set of Dutch sentences. This probe model's performance is tested on sentences generated by the two different CFGs. BERTje is used to convert sentences to their corresponding contextualised representations before feeding them to the probe model (Devlin et al. 2018). The complete process is visible in Figure 3. By mapping BERTje-contextualized word tokens to scale values, the model links embeddings to every phrase in the sentences; this is included in the training of the probe model. The model gives a global attention to each sentence as if it were a probability or an estimate that specific nouns or verbs belong in a certain ellipsis. After this, a Sigmoid function is applied to give a 0 or 1 by classification, called Sparse Attention. This last step connects a subject or verb with the accompanying ellipsis.

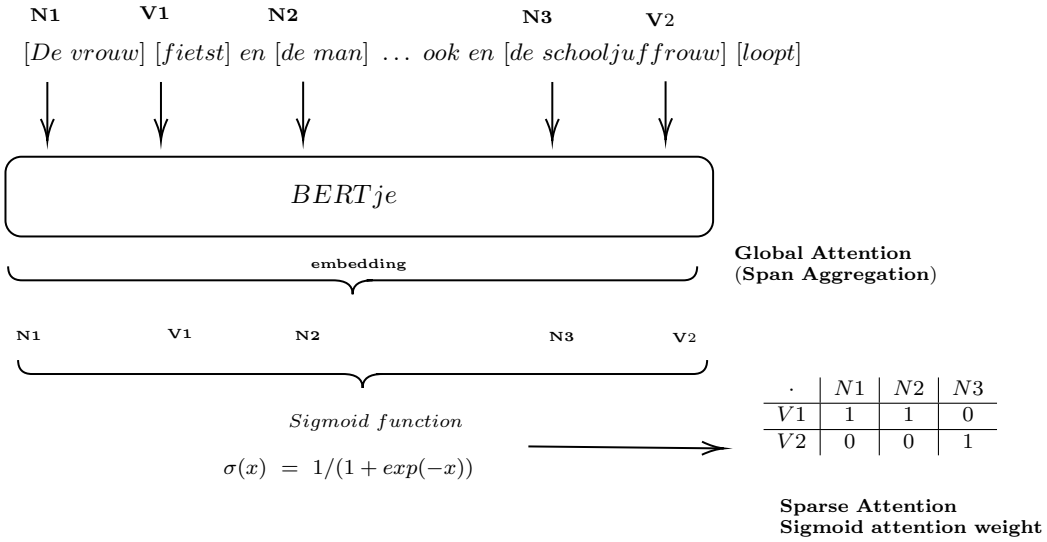

Figure 2: Encoding of phrases and the probe model

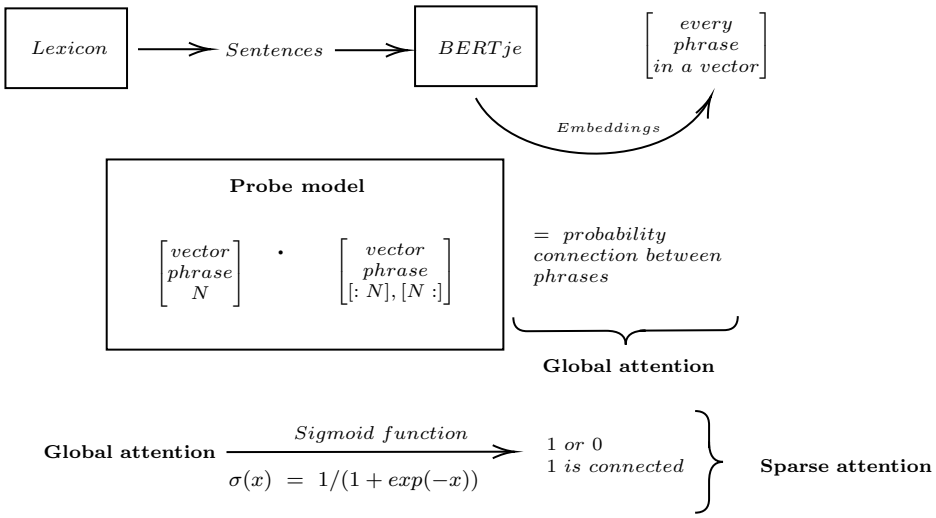

Figure 3: The complete model including the lexicon, BERTje and the probe model

4. Results and Discussion

The performance metrics (accuracy, precision, recall and F1) have been calculated across all types of sentences in the test data. Next to the test data, these performance metrics have also been calculated across the validation set of the Lassy Small data. All these results are displayed in table 3. The generated sentences have a tree depth of two or three. Although the grammar allows for infinite depth, the generation process is limited to a depth of three to generate only natural sentences. Results for these different tree depths separately are shown in Appendix A. Since we use randomly generated sentences, we performed three different tests on the base structure, which are shown in Appendix B. Since all metrics are similar, it is clear that the randomness of sentence generation is not affecting the performance. Considering that accuracy is a performance measure that attributes the same cost to false positives and false negatives and identifying subject-verb dependencies is an

Metric	Validation set	Base structure of CFG	NP ellipsis	VP ellipsis
Precision	0.872	0.579	0.781	0.839
Recall	0.991	0.669	0.663	0.394
F1	0.891	0.621	0.717	0.536
Accuracy	0.989	0.872	0.882	0.805
True baseline accuracy	0.048	0.197	0.191	0.287
False baseline accuracy	0.952	0.803	0.810	0.713

Table 3: Performance metrics of the probe model on the Lassy corpus, the baseline, verb, and noun elliptical sentences

imbalanced problem (as observed from the varying true and false baseline accuracy values in table 3), we have opted to ignore accuracy as it is a skewed measure in this scenario. Since the F1-score is indicated for uneven class distributions, we opted for the measurement for comparison between the different sets.
The results indicate that the probe performs well on the validation set from the Lassy data set with an F1 score of 89,1%. However, the sentences generated from our base structure CFG, where no ellipsis is present, have an F1 score of 62,1%. This is a significant difference, showing that the probe performs lower on our generated sentences, even when no ellipsis is present, despite being well-optimized on the Lassy data set. The expectation is that this happened because BERTje makes contextualised embeddings. Our generated sentences originate from a CFG in which phrases that substitute the terminal symbols are chosen randomly from the lexicon. Even though it has made sure that verbs are suitable for the subjects and objects that are always persons, it could still be that certain combinations of verbs and nouns are very unlikely. Therefore, these sentences can be unnatural, which could make their contextualised embeddings of lousy quality since the context in these sentences is not meaningful.

Furthermore, the noun phrase ellipsis shows a significant improvement from the base structure CFG, with a total F1 score of 71,7%. While we are not entirely sure why the base structure CFG has such a low performance when compared to the noun phrase ellipsis, we have a few hypotheses. The first likely reason for this to happen is that the noun phrase ellipsis occurs more in natural language and thus appears more in the training set (Goksun et al. 2011). Furthermore, the distance between the ellipsis and its corresponding antecedent is often relatively small, possibly making the task of identifying subject-verb dependencies less challenging for the probe model.

At last, the results allow us to conclude that the probe performs worst in the context of verb phrase ellipsis since it only has an F1 score of 53,6%.

We believe this decreased performance compared to the noun phrase ellipsis is due to the fact that the verb phrase ellipsis is less common in natural language and thus less common in the training data (Goksun et al. 2011). Indeed, this ellipsis is less present in the training data, causing the probe model to be less trained for spotting subject-verb dependencies in the context of verb phrase ellipsis. Furthermore, the distance between the ellipsis and its corresponding antecedent is often larger than for noun phrase ellipsis, which could make the task of identifying subject-verb dependencies more challenging.

On top of the overall results presented in table 3, the performance of the probe model on different tree depths has also been considered. Table 4 and 5 in Appendix A show the performance metrics for a tree depth of two and three separately. As mentioned before, a larger tree depth was hypothesised to make the task more challenging for the probe model. We can see a clear difference in the F1 score for the base structure CFG and the verb phrase ellipses performance between tree depths of two and three. Where the base structure has an F1-score of 66,7% at depth two, it only has an F1-score of 59,8% at depth three. For verb phrase ellipsis we have F1-scores of 59,2% and 38,6%

respectively. The noun phrase ellipsis has F1-score of 75,1% for depth 2 and 70,7% for depth 3. Tree depth seems to affect performance in the context of verb phrase ellipsis more than it does an effect in the context of noun phrase ellipsis. The aforementioned higher frequency of noun phrase ellipsis in the training data and the smaller distance between the ellipsis and its antecedent could account for this observation. The observed differences for verb phrase ellipsis confirm that the task gets more challenging when sentences are of higher syntactic complexity.

From these results can be concluded that the probe model used in this research is generally inadequate at spotting subject-verb dependencies in the context of noun and verb phrase ellipsis. This could have multiple explanations. First, the randomly chosen lexical items create meaningless contexts, which could harm the quality of BERTje's embeddings. Second, the differences between performance for noun and verb phrase ellipsis could be attributed to them being of a different difficulty level to resolve. Lastly, the analysis of different tree depths confirms that syntactic complexity is indeed an important factor in the model's performance.

5. Conclusion and Future Work

From the results, we can conclude that the probe trained by BERTje embeddings can not find subject-verb dependencies in elliptical sentences made by the current CFG. Furthermore, the model struggles more with verb phrase ellipsis than noun phrase ellipsis. Multiple factors contribute to these results, namely the random nature of lexical choice in the sentence generation, the frequency of both ellipsis types in natural language and the difficulty of the tasks at hand. Still, the overall lower performance of the model on ellipsis compared to the validation and base structure CFG performance indicates that BERTje's syntactic capability is limited and, therefore, should only be used for semantic tasks. This is supported by the dropping performance of the probe model as sentences get more syntactically complex. Thus, much work must be done before general language models can successfully capture ellipsis. Nevertheless, these results are a valuable input to future research on how to improve language models like BERTje.

5.1 Future Work

Several directions for future work could be fruitful. Firstly, there is significant room for improvement in the CFG. More complex data could be generated that contains even more distractors in addition to the DC component to achieve distribution in syntactic structures that might be closer to actual data. In the research, synthetic data was used, which makes it harder to parse since all semantic clues are missing. More complex data makes it more visual how the model would work with real data, containing many complex noun phrases. Furthermore, since BERTje is contextualised, a structure with natural context would likely result in better performance. Apart from controlling for semantic plausibility by choosing verbs that are suitable for personal subjects and objects, an essential direction would be to constrain the algorithm to choose lexical items that result in natural sentences. Another direction would be to evaluate in what sense full parsers can reconstruct ellipsis. This could be used for the model to see if it improves the results and to see a clear trade-off between probing and full parsing. One additional improvement would be to figure out why the base structure CFG has such a low performance, especially when compared to the noun phrase ellipsis CFG.

Limitations in computational resources limited the generation of sentences to a tree depth of three. Future research could investigate more significant tree depth to investigate its effect on the performance of the probe model. This reveals more about the relationship between syntactic complexity and the ability of BERTje to capture ellipsis.

The current work has researched two types of ellipsis: noun phrase and verb phrase ellipsis. However, many other types of ellipsis exist. Future work could focus on the capabilities of language models in capturing these linguistic structures.

Furthermore, the ellipsis can express itself differently in other languages. As mentioned, there is no verb phrase ellipsis in German, but there is in Dutch and English. A future work possibility would be to explore ellipsis in such languages.

This paper aimed to show the syntactic capabilities of BERTje. Furthermore, it would be interesting to see how our results compare to those of similar well-known language models, such as RoBERTa (Liu et al. 2019).

6. Acknowledgement

We are grateful to Dr. Konstantinos Kogkalidis (Utrecht University) for his assistance with the research.

References

Aralikatte, Rahul, Matthew Lamm, Daniel Hardt, and Anders Søgaard (2019), Ellipsis resolution as question answering: An evaluation. https://arxiv.org/abs/1908.11141.

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

Bresnan, Joan, Ronald M. Kaplan, Stanley Peters, and Annie Zaenen (1987), *Cross-Serial Dependencies in Dutch*, Springer Netherlands, Dordrecht, pp. 286–319. https://doi.org/10.1007/978-94-009-3401-6$_1$1.

Chomsky, Noam (1959), On certain formal properties of grammars, *Information and control* **2** (2), pp. 137–167, Elsevier.

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D Manning (2019), What does bert look at? an analysis of bert's attention, *arXiv preprint arXiv:1906.04341*.

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni (2018), What you can cram into a single vector: Probing sentence embeddings for linguistic properties, *arXiv preprint arXiv:1805.01070*.

Cyrino, Sonia and Gabriela Matos (2005), Local licensers and recovering in vp ellipsis, *Journal of Portuguese Linguistics*, Open Library of Humanities.

de Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019), BERTje: A Dutch BERT Model. http://arxiv.org/abs/1912.09582.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), RobBERT: a Dutch roBERTa-based language model, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 3255–3265. https://aclanthology.org/2020.findings-emnlp.292.pdf.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018), Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805.

Evers, Arnold et al. (1976), The transformational cycle in dutch and german, *Nieuwe (De) Taalgids* **69** (2), pp. 156–160.

Goksun, Tilbe, Tom Roeper, Kathy Hirsh-Pasek, and Roberta M Golinkoff (2011), From nounphrase ellipsis to verbphrase ellipsis: The acquisition path from context to abstract reconstruction, *Occasional Working Papers in Linguistics* **38**, pp. 53–74.

Hardt, Daniel (1997), An empirical approach to vp ellipsis, *Computational Linguistics* **23** (4), pp. 525–541.

Kogkalidis, Konstantinos and Gijs Wijnholds (2022), Discontinuous constituency and BERT: A case study of Dutch, *Findings of the Association for Computational Linguistics: ACL 2022*, Association for Computational Linguistics, Dublin, Ireland, pp. 3776–3785. https://aclanthology.org/2022.findings-acl.298.

Lappin, Shalom (2005), A sequenced model of anaphora and ellipsis resolution, *Anaphora Processing: Linguistic, Cognitive, and Computational Modelling. Amsterdam: John Benjamins* pp. 3–16.

Lin, Chuan-Jie, Chao-Hsiang Huang, and Chia-Hao Wu (2019), Using bert to process chinese ellipsis and coreference in clinic dialogues, *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)* pp. 414–418.

Liu, Yinhan, 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. https://arxiv.org/abs/1907.11692.

McShane, Marjorie and Petr Babkin (2016), Detection and resolution of verb phrase ellipsis, *Linguistic Issues in Language Technology, Volume 13, 2016*.

McShane, Marjorie J et al. (2005), *A theory of ellipsis*, Oxford University Press on Demand.

Merchant, Jason (2005), Fragments and ellipsis, *Linguistics and philosophy* **27** (6), pp. 661–738, Springer.

Nielsen, Leif Arda (2004), Verb phrase ellipsis detection using automatically parsed text, *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1093–1099.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami (1991), On multiple context-free grammars, *Theoretical Computer Science* **88** (2), pp. 191–229, Elsevier. https://www.sciencedirect.com/science/article/pii/030439759190374B/pdf?md5=a25be6b71683a69af87be6060f523d5d&pid=1-s2.0-030439759190374B-main.pdf.

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. (2018), What do you learn from context? probing for sentence structure in contextualized word representations, *International Conference on Learning Representations*.

Van Craenenbroeck, Jeroen (2017), Vp-ellipsis, *The Blackwell companion to syntax,* pp. 1–35.

van Noord, Gertjan, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste (2013), *Large Scale Syntactic Annotation of Written Dutch: Lassy*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 147–164. https://doi.org/10.1007/978-3-642-30910-6$_9$.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017), Attention is all you need, *Advances in neural information processing systems*.

Wijnholds, Gijs and Mehrnoosh Sadrzadeh (2019), Evaluating composition models for verb phrase elliptical sentence embeddings, *NAACL HLT 2019-2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies-Proceedings of the Conference*, Vol. 2019, Association for Computational Linguistics (ACL), pp. 261–271.

Appendix A. Model performance per tree depth

Metric	Base structure of CFG	VP ellipsis	NP ellipsis
Precision	0.675	0.818	0.833
Recall	0.659	0.464	0.684
F1	0.667	0.592	0.751
Accuracy	0.877	0.813	0.882
True baseline accuracy	0.187	0.292	0.242
False baseline accuracy	0.813	0.708	0.758

Table 4: Performance metrics of the probe model the baseline, verb, and noun elliptical sentences with depth two

Metric	Base structure of CFG	VP ellipsis	NP ellipsis
Precision	0.550	0.737	0.765
Recall	0.657	0.261	0.657
F1	0.598	0.386	0.707
Accuracy	0.874	0.777	0.881
True baseline accuracy	0.143	0.268	0.187
False baseline accuracy	0.857	0.732	0.813

Table 5: Performance metrics of the probe model the baseline, verb, and noun elliptical sentences with depth three

Appendix B. Three test model performance base structure

Metric	Base 1	Base 2	Base 3
Precision	0.574	0.5695	0.5691
Recall	0.741	0.6567	0.6571
F1	0.647	0.6100	0.6100
Accuracy	0.869	0.8748	0.8747
True baseline accuracy	0.161	0.1491	0.1490
False baseline accuracy	0.838	0.8508	0.8509

Table 6: Performance metrics of the probe model on three different random sentences with the base structure