Evidence of Hierarchically-Complex Syntactic Structure Within BERT's Word Representations

Mary Katie Kennedy
University of Southern California, Los Angeles, CA, USA
mkkenned@usc.edu

Abstract

Our research provides empirical support that LLM's contextualized word embeddings have captured deep and hierarchical syntactic structure. In 2019, Hewitt and Manning found evidence that LLMs have captured features of structural dependency parses within their word representations; we extend this work by deploying their methodology on sentence structures that are differentiated only in a constituency-based account like Minimalism rather than a dependency-based account. Our novel work creates a dataset containing several carefully selected sentence structures whose dependency parses are identical, but whose constituency trees differ due to to the size of the complement (vP versus TP versus CP). We find differences in the probe's predicted distances that can only be explained if the embeddings have indeed captured some Minimalist structural difference between these sentence types. The impact of our work helps to realize Linzen (2019)'s argument that linguists can further the study and understanding of LLMs and that the field of NLP provides novel tools for further linguistic research.

1 Introduction

Since the release of BERT (Devlin et al., 2019), much research has been done to test and expand the impressive performance of large language models. A subset of research interest lays in understanding what linguistic structures and knowledge these models have acquired (Jawahar et al., 2019; Belinkov and Glass, 2019; He et al., 2024; Waldis et al., 2024; Kallini et al., 2024), including syntactic (Clark et al., 2019; Chi et al., 2020; Kulmizev et al., 2020; Maudslay and Cotterell, 2021; Arps et al., 2022), morphological (Coleman, 2020; Anh et al., 2024), and semantic knowledge (Nikolaev and Padó, 2023; Kamath et al., 2024).

Our work extends this body of research by utilizing a probe method developed by (Hewitt and Manning, 2019), which finds that a dependency parse can be recovered solely from the contextualized vector embeddings of a pretrained language model like BERT (Devlin et al., 2019). We further these findings by deploying the probe on sentence structures whose dependency parse is invariant (i.e., the distance between a head and its dependent is always 1, see Section 2.1 for explanation), but whose hierarchical distances vary depending upon the size of a phrasal complement in a Minimalist constituency framework (see Section 3.2 for details). In doing so, we seek to discover whether large language models like BERT have captured the complex hierarchies and subsurface structures postulated by syntacticians in the Minimalist Program. This work thus follows in the research vein of Linzen (2019), who argues that linguists and NLP researchers stand in a unique position for collaboration to leverage the skills and tools of their respective fields to better understand, test, and develop the two bodies of research.

2 Background

2.1 Syntactic Theories

In the field of NLP, there are two main approaches to syntax that a researcher can utilize: a Dependency Grammar (DG) approach or a constituency grammar (CG), also known as a phrase-structure grammar. In brief, Dependency Grammar focuses more on the relationship between constituents without needing to represent a sentence's linearized word-order, making it popular for work on languages with freer word order (Müller, 2019).[1]

The core of the theory centers around the concept of *valence*, which indicates which words govern

[1]Various schools of thought in the theory have proposed different mechanisms to derive linear order from a dependency structure, including the idea that linear order is dictated by surface syntactic rules (Müller, 2019). The author of this approach, Ulrich Engel, published in 2014 in (Öhl, 2015), though the original source is in German.

Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 271-286.
Eugene, Oregon, July 18-20, 2025

(1) Dependency Tree

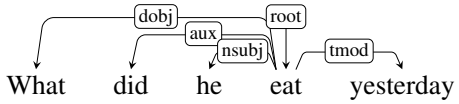


Figure 1: An example of the dependency tree for the sentence "What did he eat?" Note the flatter structure, the one-to-one mapping of words to nodes in the tree, and how each word has one and only incoming arc, excepting the root.

which words in a sentence. The governing word in a phrasal pair is considered the "head" and the governed word is its "dependent," sometimes called its "valence" (Müller, 2019). Each sentence will have one and only one "root," which is typically the matrix verb of the sentence, that will have no head itself. Thus, in a dependency tree, all words—except the root—will have one and only one incoming arc from its head. Though a word itself can head several other words, it itself can only be headed by one other word (see Example (1)).

On the other hand, constituency grammars are popular amongst many syntacticians and linguists who have built theories off of the work of Chomsky and others who have refined various aspects of phrase-structure/constituency-based grammars (Chomsky, 1957, 1981, 1986, 1995). Phrase-structure grammars are based around X-bar theory and operations of *Merge* and *Move* (Chomsky, 1995) and their consequent traces (Chomsky, 1973; Fiengo, 1977) (such as question formations where "He ate chicken" transforms into "What did he eat?"). After all syntactic operations are applied and all relevant nodes have been moved and/or merged, the end result is the sentence's linearization, meaning the final locations of the words in the hierarchy should match what is actually uttered if the tree is read from left to right (see Example (2)). Constituency-based grammars (CGs) thus result in trees with deep and complex hierarchies wherein empty nodes must be inferred as the traces and remnants of previous operations.

Like DG, many constituency theories incorporate the concept of valence, albeit with some modifications. Some of Chomsky's earlier work in the theory of Government-Binding (Chomsky, 1981) stipulates that certain categories (particularly the lexical categories of Verb, Noun, Adjective/Adverb, and Preposition in addition to the functional category of Tense) head/govern/dominate other con-

(2) Constituency Tree

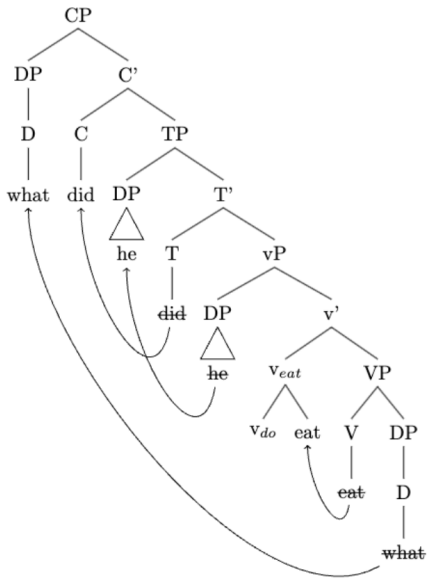


Figure 2: An example of a constituency tree for the sentence "What did he eat?" Note the depth of the tree and the movement of elements.

stituents.[2] Later theories (Chomsky, 1995) refined this by defining specific operations, such as Merge, where the head element provides the properties of the combined result (e.g., Verb *eat* + Noun *chicken* = VerbPhrase *eat chicken*, not NounPhrase *eat chicken*), and which enables the recursive feature of language (e.g., "The old lady swallowed a fly that was then caught by a spider she later swallowed that was...."), thus allowing for infinite embeddings.

In short, both theories postulate a primitive building operation that allows for the combination of two elements into a single, new element whose features are determined by the head word, enabling the recursive nature of language to appear. For DG, this is through the dependency relationship, which establishes the head; for CG, this is through the Merge operation, which assigns the features of the phrase by referring to the phrase's head. The core differences, meanwhile, can be summed up as:

1. Dependency Grammars use a one-to-one mapping between words and nodes in the tree. Constituency Grammars more often use a one-to-many mapping between nodes in the tree,

[2] A constituent *A* can govern another constituent *C* iff *C* does not govern *A*, and there is no intervening element *B* that governs *A* but not *C*.

postulating branches and nodes that are not overtly present in the spell-out.

2. Dependency Grammars root at the verb. In Constituency Grammars, generally the Complementizer Phrase (CP) or Tense Phrase (TP)[3] exists as the highest level, though it is true that all sentences must have a verb in order to valid.

3. Structurally, Dependency Grammars do not distinguish between a head's arguments (e.g., the subject or object of a verb) and its adjuncts (e.g., modifiers, such as an adverb or prepositional phrase modifying the verb). The difference is left to the dependency label, but the structure remains changed. In contrast, Constituency Grammars, particularly Minimalism, structurally distinguish between the two, and even between argument types.

4. Dependency Grammars opt for reduced, flatter, more horizontal representation of word-to-word relationships. Constituency Grammars opt for a more hierarchically complex, vertically-organized representation.

When syntax is leveraged in NLP, the framework adopted tends to be DG rather than CG (compare 14,900 ACL papers on Dependency Grammar as opposed to only 3,630 on Constituency Grammar). There are several reasons for this: DG's trees are simpler (nodes are in a one-to-one relationship with words), DG is more static (dependencies are assessed in-situ, meaning one needs not be concerned whether or not an element moved to its location or base-generated there), DG utilizes flatter representations (because elements are assessed in-situ, there is no need to postulate more complex and empty hierarchies that might explain how or why the word is currently where it is), and its simplicity and avoidance of contentious theoretical debates—such as those in Minimalism—allow for faster and more consistent inter-annotator agreement.

The DG framework is appealing to many in NLP as it is relatively easy to learn and its compact and efficient representation has proven to be salutary to downstream tasks, such as question-answering, relation extraction, summary (de Marneffe et al., 2006), spam detection (Milner, 2024), sentiment analysis (Liang et al., 2021), sentence classification and matching as well as sequence labeling and machine translation (Zhang et al., 2021), and more. However, the theory fails to capture linear order, nor does it explain the patterns and restrictions that form licit sentences and their interpretations, and it furthermore entirely skirts the issues of the deep and complex hierarchies that have been argued for in Minimalism. In this vein, we seek to investigate to what extent LLMs have captured the deeper and more complex syntactic structures proposed by constituency grammar frameworks, such as Minimalism.

[3]Some languages do not include tense, like Chinese, and so the top level is often represented as IP for Inflectional Phrase.

2.2 Probes

Since LLMs took the world by storm with their impressive performance in multiple language tasks, researchers have sought to understand what linguistic properties LLMs have actually acquired. A popular method is the probe method, first proposed by Shi et al. (2016), which used the embeddings from neural machine translation encoders to train a logistic regression classifier in order to identify what syntactic features were acquired by the models. This field of research and these probe models are not concerned with improving state-of-the-art performance; rather, they seek to investigate, or "probe", what latent linguistic features a language model has acquired.

The tasks specified by probes depend on the linguistic feature under investigation (e.g., semantics, syntax, etc.), but often utilize a pretrained language model's latent features, such as their vector representations (Conneau et al., 2018; Jawahar et al., 2019; Tenney et al., 2019b,a; Starace et al., 2023) or attention mechanisms (Clark et al., 2019; Manning et al., 2020).

One form of structural probe, developed by Hewitt and Manning (2019), found that the pretrained contextualized embeddings of BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) could be used to recover dependency trees from those vector representation of words. To find this, Hewitt and Manning trained a linear transformation matrix to take the contextualized word embeddings and project them into a subspace where the squared Euclidean distance between word nodes ultimately recovers a dependency parse. That is to say, their probe's training objective was to learn to map words' contextualized embeddings to new positions within a subspace where the probe's predicted squared Euclidean distance between each head and its de-

pendent is approximately 1.[4]

While Hewitt and Manning (2019) and others (Chi et al., 2020; Kulmizev et al., 2020; Müller-Eberstein et al., 2022; Eisape et al., 2022) have found evidence that dependency structures are encoded within the contextualized vector representations, it remains unclear whether LLMs have acquired the deep, hierarchically-complex structures of constituency grammars such as those proposed in the Minimalist framework. To this end, we utilize the structural probe of Hewitt and Manning and test sentence types whose hierarchical distance varies in a constituency/Minimalist account, but whose head-dependency distance does not vary in a Dependency Grammar account. If the probe is sensitive to the nuances of a constituency account, this indicates that not only have the language models captured something of the hierarchically complex and subsurface structures of Minimalism, but that a probe trained only to recover a dependency parses is capturing constituency syntax for free.

3 Methods

Our work is not the first research to probe at constituencies (Tenney et al., 2019b; Arps et al., 2022; Kallini et al., 2024). However, these previous methods either focus solely at the phrase-level by seeking to train a probe to recover a phrase's boundaries (Tenney et al., 2019b; Kallini et al., 2024) or by training on the English Penn Treebank for their probe (Arps et al., 2022). While constituency trees represented in the English Penn Treebank (Marcus et al., 1993) are deeper than their equivalent dependency trees, they do not adhere to the binary branching requirement postulated in Minimalism and do not capture Merge and Move operations. As such, the representations are not as rich nor as complex as those which have been posited in the Minimalist constituency framework.

For this reason, we opt for the novel approach of utilizing the original Hewitt and Manning (2019) structural probe that was trained to recover dependency trees to probe for variations in constituency hierarchies. To that end, our stimuli involve sentences wherein the distance between a head and its dependent is invariant in a DG account, but whose hierarchical distance depends upon the sentence structure as captured in the Minimalist framework. The choice to probe for a dependency parse as opposed to a constituency in fact allows us to avoid several potential pitfalls of constituency trees: namely that constituency trees make assumptions about the underlying structure and may predispose the probe to recover the constituency parses utilized in the training data rather than probing for a latent representation of constituency hierarchies as captured by the model.

3.1 Computational Model

The structural probe by (Hewitt and Manning, 2019) stipulates a model *M* that produces a sequence of vector representations $h^l_{1:n}$ from an input sequence of *n* words $w^l_{1:n}$ where *l* identifies the sentence. A linear transformation $B \in \mathbb{R}^{k \times n}$ parameterizes the parse tree-encoding distances:

$$d_B(\mathbf{h}^l_i, \mathbf{h}^l_j)^2 \quad = \quad (B(\mathbf{h}^l_i - \mathbf{h}^l_j))^T (B(\mathbf{h}^l_i - \mathbf{h}^l_j))$$

where *i* and *j* are the words in the sentence and where the matrix *B* is trained to reproduce the gold parse distances between each pair of words (w^l_i, w^l_j) in each sentence for all the sentences within the parsed training corpus T^l.[5] This training is accomplished through the gradient descent objective:

$$\min_B \sum_l \frac{1}{\mid s^l \mid^2} \sum_{i,j} \mid d_{T^l}(w^l_i, w^l_j) - d_B(h^l_i, h^l_j)^2 \mid$$

In doing so, the objective seeks to approximate the matrix that most closely reproduces distances that align with the gold-standard distances. $\mid s^l \mid$ is the length of the sentences, and the function normalizes using the square of the sentence's length since each sentence contains $\mid s^l \mid^2$ pairs of words.

Hewitt and Manning (2019) trained their structural probe using BERT-large (cased) with 1024 dimensionality for all 24 layers. The probe was trained with the objective of minimizing the L1 loss of the predicted squared distance with respect to the true distance (i.e., the distance between a head and its dependent should be 1; the distance between the dependent of a dependent of a head should be 2; and so on). They used Adam optimizer (Kingma and Ba, 2014) with an initialized learning rate of 0.001 with $\beta = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-2}$ and an epoch maximum of 40 or to convergence with a batch size of 20. Dev loss

[4]The specific mathematics and model information can be found in Section 3.1.

[5]The authors found that training on squared distances and using the square root to retrieve the final distance performed better than using the direct distance. Hewitt and Manning (2019) left the possible reasoning for this for future work.

was calculated at each epoch; if the dev loss was not a new low for the model, the optimizer was reset with an initial learning rate multiplied by 0.1. The probe was implemented using DyNet (Neubig et al., 2017) and PyTorch (Paszke et al., 2019).

Probe evaluation was based on how closely the predicted distances between word pairs align with the gold parse structures, which were created by converting the constituency trees from the English Penn Treebank (Marcus et al., 1993) into dependency parses.[6] To measure this, the authors calculated the minimum spanning tree for each sentence's predicted distances and scored the undirected, unlabeled attachment score (UUAS), which merely measures whether or not the proper word-pairs are in a dependency relationship, ignoring the matter of directionality (which indicates which word is the head and which is the dependent in a head-dependent pair) and labels.

3.2 Linguistic Data

To probe whether vector embeddings encode the hierarchical distances captured by Minimalist constituency trees, we utilize the filler-gap dependencies that result from *wh*-question formation of sentences with embedded sentential complements (e.g., "What did she see [him eat __]"). By varying the size of the complement taken by the matrix verb and extracting out of that embedded complement, we can vary the constituency tree's hierarchical distance while keeping dependency distances constant.[7]

In traditional Minimalism, there is an accepted order to the hierarchy of phrases. At the highest level is the **complementizer phrase**, which introduces whether the clause is interrogative or declarative; under the CP is the **tense phrase**, which hosts tense information; the TP nests a **verb phrase**, which can further be subdivided into a small verb phrase (vP) also known as a **voice phrase** that takes a VP complement itself (Adger, 2003).

Different verbs can vary in the type and size of the complement they can take. At the largest level, a verb can take an entire finite clause as its complement (see Example (3)). Examples of such verbs include *think, believe, suspect, claim,* etc., which can all optionally include an overt complementizer like *that* or *who.*

(3) Full CP Complement
  a. I think [$_{\mathrm{CP}}$ (that) he ate the chicken]

The next smallest complement size is a non-finite complement. The easiest one to discuss is the infinitive complement in sentences known as *exceptionally case marked* (ECM) (see Example (4)), which include matrix verbs that take TP complements (Adger, 2003). ECMs are called exceptionally marked because the subject of the embedded clause receives its accusative case (rather than the typical nominative case) from the matrix verb.

(4) ECM TP Complement
  a. I expect [$_{\mathrm{TP}}$ him to eat the chicken]

Another small subset of verbs in English allow for phrasal complementation. This subset of verbs include causatives (e.g., *make, let*) and perception verbs (e.g., *see, hear, watch, feel*) that take bare infinitives (see Example (5)). We follow in the steps of (Sheehan and Cyrino, 2023) in analyzing these as vPs, which we dub "bare vPs" to emphasize that the nonfiniteness is not overtly realized with an infinitival *to* as it is in ECMs.

(5) Bare Infinitive Complement
  a. I saw [$_{\mathrm{TP/vP}}$ him eat the chicken]

For our experimental design, we specifically needed sentence structures in which the dependency parse remained consistent, but the constituency parse yielded differing distances between two elements. For this reason, we leveraged the ability for verbs to take complements of differing sizes (vP, TP, and CP) and created *wh*-questions (e.g., *what did you see him eat/what did you expect him to eat/what did you think he ate*). *Wh*-question structure was specifically selected as the distance between the embedded verbal head (e.g., *eat*) and

[6] It is important to note here that the constituency trees of the Penn Treebank are not the binary branch trees with Merge and Move operations as postulated in Minimalism.

[7] While our experiment utilizes filler-gap dependencies, our probe method can be applied to any sentence structure types whose constituency tree varies but whose relevant dependency parse does not. Hewitt and Manning (2019) probe's training objective allows for flexibility in possible Minimalist structures. Its training objective is such that a parent-child relationship between a head and its dependent should return a distance of approximately 1, while a "grandparent"-child relationship (the dependent of a dependent of a head) should return a distance of approximately 2, and so on. Using this feature, Kennedy (2025) deploys our probing method on declarative Subject-Raising and Subject-Control constructions—the former of which is argued to take a smaller TP complement compared the latter's larger CP complement—and finds that the predicted Euclidean distance between matrix elements and embedded elements are larger in the Subject-Control condition despite the two structures having identical dependency parses.

its dependent (e.g., *what*) is consistently 1 in all conditions; however, in a Minimalist account, the hierarchical distance between embedded verb and its moved object depends on the size of the complement taken (vP, TP, or CP). For visualization, see the trees in Appendix B, Examples (6)-(9).

To add further complexity, two more sets of sentences were constructed that took advantage of the recursive property by creating sets for double-nested ECMs (e.g., *What did you expect her to want him to eat*) and double-nested full-CP complements (e.g., *What did you believe she suspected he ate*).

Using only pronouns for the subjects, the minimum linear distance (meaning the number of intervening words) between the extracted *wh*-constituent and the embedded verb ranged from 5 (bare vP and single CP) to 6 (single TP) to 7 (double CP) to 9 (double TP). Because the sentences could not be started at identical linear distances due to the presence of necessary words (such as *to* in ECMs), the linear distance was increased incrementally through the change of a pronoun (e.g., *you*) to a nominal phrase (e.g., *the professor*) to a modified nominal phrase (e.g., *the brilliant professor*) to a possessive nominal (e.g., *the brilliant professor's friend*) to the inclusion of an adverb.[8]

Using the above schema, we created a total of 18,252 carefully constructed sentences that strictly conformed to one of the five specific syntactic constructions that are well-accepted in traditional syntax as demonstrating different syntactic hierarchies.

4 Experiment

For our experiment, we used the best-performing pretrained probe from Hewitt and Manning (2019), which they found to be the probe for Layer 16 and which they released and made publicly available on their Github.[9] Our methodology sought to discover whether the probe's predicted squared Euclidean distances between head-dependent words were sensitive to hierarchical depth as postulated in a Minimalist framework. In a DG framework, the distance between a head and its dependent **should always be 1** across our five conditions. However, in a Minimalist account, the size of the complements (vP, TP, CP, TP-TP, and CP-CP) yields **longer and longer hierarchical distances** between the moved *wh*-object and the embedded verbs.

The contextualized embedding representations of our 18,252 sentences were fed into the pretrained probe, and we extracted the squared Euclidean distances between the new projections of the *wh*-word and the embedded verb *if and only if* the minimum spanning tree correctly established a head-dependent relationship between moved *wh*-word (the first word) and the in-situ embedded verb (the last word). As our experimental design rests upon comparing the predicted squared Euclidean distance of a *dependency* probe when given sentences whose structures vary only in a *constituency* Minimalist account, we were only interested in sentences in which the probe correctly identified the head-dependent relationship because there is little point in comparing the predicted dependency distances of an incorrect dependency parse.[10]

4.1 Predictions

The structural probe was trained only to recover latent dependency representations captured by the pretrained BERT model. Thus, the probe has no specific or overt reason to show sensitivity to constituency-based distances. If the probe is sensitive only to dependency representations, then the five conditions should show no difference in distances predicted by the model. Alternatively, it is possible that the contextualized vector representations have captured Minimalist-like syntax, but that the dependency-trained probe is insensitive to such features.

The more interesting outcome, however, would be if the model's predicted distances *are* affected by the constituency distances. If predicted distances are reflective of an influence of constituency distances, this would suggest 1. that the model itself captures some representation of Minimalist-like constituency in addition to dependency, and 2. that the dependency representations themselves are sensitive to constituency differences. Such findings would have implications for modeling this distinction in the theory of Dependency Grammar.

If it is found that the probe is able to pick up on constituency hierarchies, then we would anticipate that embedded verbs with CP complements should have the highest predicted distance as it has the highest number of hierarchical nodes between the

[8] For more detail on our dataset creation, see Appendix A.

[9] https://github.com/john-hewitt/structural-probes

[10] While the fail cases are of interest for further research and investigation, for our current purposes, robust analysis could only be conducted when the probe achieved its trained gold parse.

extracted *wh*-object and the embedded verb within the constituency tree. ECM verbs that take TP complements and perception verbs that take either bare vP complements should trail behind this.

4.1.1 Dependency vs Constituency for Probes

As mentioned, our probe is intentionally trained to recover dependency parses as opposed to constituency trees. While it may seem intuitive to utilize a probe trained to recover constituency trees like Arps et al. (2022), we argue that using a dependency probe for Minimalist constituency structures actually has several advantages.

The logic behind linguistic probes is that in order for them to be successful, the embedding representation (or attention scores for some probes) must encode some feature(s) of that linguistic phenomenon in order for the probe to be able to solve the task. However, one critique of probing methods is the concern that the probe may simply be learning the linguistic task rather than revealing latent features encoded within the representation (Hewitt and Liang, 2019). Our stance is that using a dependency probe to test for constituency-based hierarchical distances avoids this possible liability.

The Hewitt and Manning (2019) probe is trained to recover *only* head-dependency relationships such that the distance between a head and its dependent is approximately 1. While the constituency trees for our stimuli will vary in the number of intervening nodes between the extracted *wh-* word and its verb (with the hierarchical distance being largest with a CP complement followed by a TP complement followed by a vP complement), the dependency parses have an invariant distance of 1 (see examples (6)–(9) in Appendix B for visualization). Because the probe *isn't* trained to predict a syntactic size difference between the complement types, the predicted squared Euclidean distances shouldn't vary *unless the probe is picking up on some additional linguistic feature within the vector representation*. The training objective is naive to a difference in the complement sizesm, and because of this, the training objective cannot bias the probe to output a desired structural difference. Therefore, if the probe's distances *do* vary in theoretically-predicted ways, we can have a greater confidence in significant results that constituency hierarchical distances are captured within vector representations and that such representations are utilized to some extent to recover dependency parses. In this regard, our methodology helps to address issues raised by Maudslay et al. (2020) that an overly powerful probe blurs the line between probe and parser.

The second benefit of using a dependency-trained probe as opposed to a constituency-trained probe is that we can avoid biasing certain debated syntactic analyses. Kuznetsov and Gurevych (2020) finds that the linguistic formalism utilized can impact how a probe performs, both in its accuracy scores and in the means through which it makes predictions (e.g., which attention layers are utilized). A probe that seeks to recover constituency parses will inevitably need to pick a "gold" standard tree that includes structure whose syntactic analysis varies even within the Minimalist framework.

For example, we mentioned how perception verbs are debated to take either a vP (Sheehan and Cyrino, 2023) or bare TP (Felser, 1998) complement. Were we to train a constituency probe, we would need to overtly pick one side of the argument and would include training data that reflects one analyses, thus risking biasing the probe towards that particular analysis. Dependency parses, meanwhile, are minimalistic (but not Minimalist) in that they make few theoretical assumptions with the most important being that there exists a dominance relationship between a head and its dependent. Using a probe trained for minimalistic dependency parses lets us to remain as theoretically-agnostic as possible within the general Minimalist framework and allows us to probe for models' representational differences as opposed to imposing debated syntactic structures upon the probe.

5 Results

Of the 18,252 sentences fed to the probe, 4,034 properly established a dependency relationship between the *wh*-word and the embedded verb.[11] A linear mixed effect model was then fit using the constituency hierarchical representation (EmbedType), the linear distance between the target words (LinDist), and the interaction of the two as predictors. EmbedType was a categorical predictor that included perception verbs (BareVP), singular ECMs (SingTP), singular CP complements (SingCP), double ECMs (DoubTP), and double CP complements (DoubCP), which were all simple coded with BareVP as the reference level. Linear distance was a discrete variable. A by-Verb (the

[11] As mentioned, overall probe performance on these edge-case sentences is not the focus of this research, but discussion can be found in Appendix D.

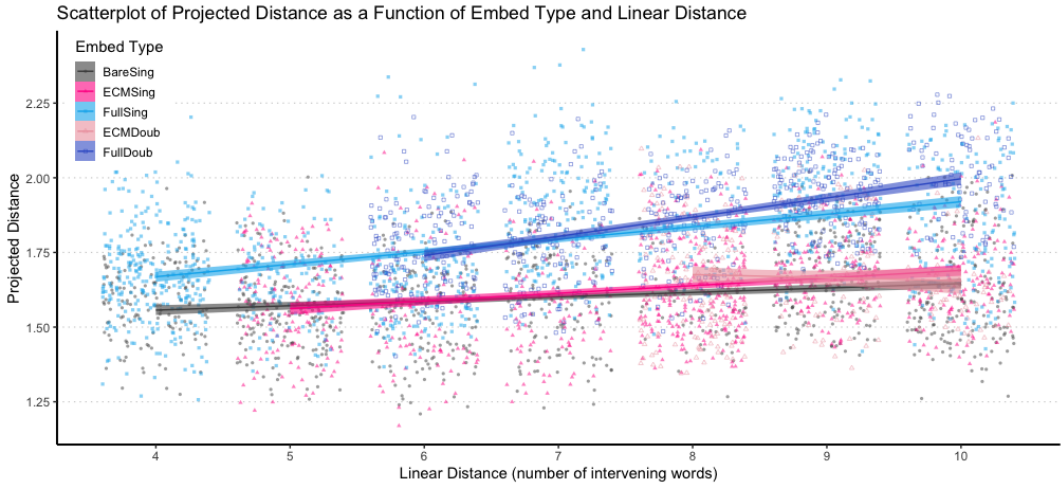


Figure 3: Scatterplot of projected distances as a function of linear distance (LinDist) and size of the verbal complement (EmbedType). There exists a stark difference between the larger CP complements and VP/TP complements. Statistical analysis reveals a significant difference between all conditions when considering their interactive effective with linear distance.

most deeply embedded verb; "eat" in our previous examples) uncorrelated random slope was added to the model.

In general, we can observe that as linear distance increases, so does the projected distance (see Figure 3). This is not surprising as it is well known that longer linear spans between dependencies tends to worsen performance as the number of intervening tokens are more likely to exceed that which is observed in training (Tenney et al., 2019b). More interesting is the clear divide in projected distances for the CP-levels versus the TP and vP levels.

The linear mixed effect model revealed significant main effects for singular TP and double CP embeddings (SingTP and DoubCP) compared to perception verb embeddings (BareVP) (see Table 1). That both SingTP and DoubCP reported projected distances that were significantly longer than the perception verb condition suggests that the probe is sensitive to constituency size.[12]

Additionally, increases in linear distance significantly corresponded to larger projected distances, though this was anticipated. Furthermore, significant interactions were found between linear distance and SingTP, linear distance and SingCP, and linear distance and DoubCP. The interaction between linear distance and DoubTP did not achieve significance, but that may be due to the notably fewer examples due to low UUAS performance.

Follow-up models were run on all categorical predictors (BareVP, SingTP, SingCP, DoubTP, DoubCP) to investigate interactions with linear distance. For all constructions, linear distance was a significant factor and the projected distances of all constructions, except DoubTP, increased with linear distance. This is expected as the greater linear distances between the two target words yielded poorer parse accuracy by the probe. That DoubTP does not conform to this behavior is likely due to it being a rare construction with few samples in our statistical analysis as the probe struggled to correctly establish the proper dependency relationship for this sentence structure.

6 Discussion & Conclusion

When linear distance is taken into account, a picture emerges in which the size of the complement (vP vs TP vs CP) is distinctly captured by the probe's correlatively larger projected distances (for further discussion, see Appendix C). These findings reveal to us several important conclusions:

1. The significant and correlative differences in projected distances between the different complement types suggest that pretrained models like BERT have learned representations that approximate in some capacity this hierarchical distinction between different complement sizes. Or, at the very least, it has picked up on

[12]That SingTP is significantly longer than BareVP but not DoubTP likely comes down to DoubTP having a much smaller sample size as this particular construction is more rare in natural data and yielded some of the lowest performance results by the probe.

Fixed Effects

Coefficient	$\hat{\beta}$	SE($\hat{\beta}$)	t	df	p
Intercept	1.462e+00	2.915e-02	8.458e+01	50.144	**2e-16**
SingTP	-1.692e-01	3.583e-02	9.217e+01	-4.723	**8.30e-06**
SingCP	-2.437e-02	2.521e-02	9.242e+01	-0.967	0.336258
DoubTP	-1.208e-01	9.775e-02	1.166e+03	-1.236	0.216882
DoubCP	-1.474e-01	4.091e-02	1.752e+02	-3.602	**0.000411**
LinDist	3.918e-02	2.399e-03	3.214e+03	16.332	**2e-16**
SingTP:LinDist	2.387e-02	3.383e-03	3.855e+03	7.055	**2.04e-12**
SingCP:LinDist	2.804e-02	2.595e-03	3.942e+03	10.805	**2e-16**
DoubTP:LinDist	1.893e-02	1.083e-02	2.703e+03	1.747	*0.080677*
DoubCP:LinDist	4.480e-02	4.361e-03	3.657e+03	10.275	**2e-16**

Random Effects

Group	Term	Variance	Std.Dev	Corr.			
Verb	Intercept	0.009815	0.09907				
	SingTP	0.030851	0.17564	-0.03			
	SingCP	0.039091	0.19772	0.10	0.82		
	DoubTP	0.008259	0.09088	-0.34	-0.30	-0.35	
	DoubCP	0.036986	0.19232	-0.24	0.72	0.82	-0.12
Residual		0.010048	0.10024				

Table 1: Number of observations: 4034. Groups: Verb (26). *P*-values/df calculated using the Satterthwaite approximation. Model formula: ProjDist EmbedType*LinDist + (1 + EmbedType | Verb). Marginal $R2 = 0.2735$, Conditional $R2 = 0.6487$.

some quality of these constructions (e.g., finite vs non-finite) that corresponds to a greater or lesser extent with a distance in which finite constructions establish further distances from their moved object and their embedded verb when compared to non-finite counterparts.[13] This benefits the field of NLP by helping to better understand what qualities and features of languages these models have implicitly learned.

2. That a probe, specifically one trained only to recover dependencies, shows a sensitivity corresponding to a constituency-based analysis indicates to us that the theory of Dependency Grammar may have reason to specifically account for these relative distances. At the very least, we must postulate that this dependency probe is sensitive to finite constructions in that they show longer dependencies compared to non-finite constructions. The possibility of needing to account for some nested hierarchy in Dependency Grammar has already been proposed in order to explain certain syntactic patterns (Müller, 2019).

3. If pretrained models have indeed implicitly learned constituency representations in some capacity (or some parallel measure), then it may be that for the purpose of further NLP work, we do not need to incorporate the far denser and more complex constituency-based grammatical representations. While such theory has advantages and we find support for its analysis as a means to explain our data, the fact remains that the representations are extensive, requiring many branches, movement, empty nodes, and redundancies. The structures, though detailed, are too cumbersome to be easily implemented in NLP architectures, nor is it as accessible of a theory to utilize; scientists from other disciplines will have an easier time quickly learning and easily representing a dependency structure rather than a phrase structure. And if the dependency representations themselves are already affected by some constituency elements, then there may be less of an impetus to require computer scientists to learn an interesting and detailed but laborious representation when the nuances of the structures are already gotten for free in the models' geometries of their dependency representations.

[13] Such coincidences already would be suspicious enough, and warrant further investigation to draw more conclusive interpretations.

The findings of this work have implications for the NLP field and the field of theoretical syntax. Not only does this work find evidence for the rich, subsurface syntax postulated by constituency theories such as Minimalism, but it furthermore finds evidence that LLMs are not only capable capturing generative Minimalist syntactic structures, but that they already do so to some extent. Our results also show support for the continuation of work like Müller (2019), who proposes utilizing nested hierarchies in Dependency Grammar to account for the structures captured by Minimalism and now by LLMs, too. Furthermore, the work teases as the possibility of utilizing LLMs for linguistic research. If these models are capturing theories postulated in syntax, might they not also be suitable as a means of testing theories when paired with human-based judgments? Already, our results suggest that BERT may favor (Sheehan and Cyrino, 2023)'s vP analysis over bare TP accounts as the probe's distances are significantly shorter than ECM's TP distances.

For the field of NLP, this provides evidence that the linguistic properties captured by LLMs are richer and more complex than previously realized, and that utilizing a dependency framework is still adequate as it appears that methods using dependencies are likely capturing constituency hierarchies for free. Overall, this work helps to realize Linzen (2019)'s claim that the skillsets and knowledge of

the fields of NLP and Linguistics complement each other, and that the collaboration of two can help to further the respective fields.

Acknowledgements

I would like to thank Khalil Iskarous, Jon May, and Andrew Simpson for their informative discussions that have helped to enrich this research. I would also like to acknowledge and thank the reviewers whose feedback has lead to greater clarity within this project.

References

David Adger. 2003. *Core Minimalism*. Oxford University Press.

Dang Anh, Limor Raviv, and Lukas Galke. 2024. Morphology matters: Probing the cross-linguistic morphological generalization abilities of large language models through a wug test. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. 2022. Probing for constituency structure in neural language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6738–6757, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton.

Noam Chomsky. 1973. Conditions on transformations. In *A Festschrift for Morris Halle*. Hole, Rinehard Winston.

Noam Chomsky. 1981. *Lectures on government and binding*. Foris Publications.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Haley Coleman. 2020. This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

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.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Claudia Felser. 1998. Perception and control: a minimalist analysis of english direct perception complements. *Journal of Linguistics*, 34(2):351–385.

Robert Fiengo. 1977. On trace theory. *Linguistic Inquiry*, 8:35–61.

Linyang He, Peili Chen, Ercong Nie, Yuanning Li, and Jonathan R. Brennan. 2024. Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4488–4497, Torino, Italia. ELRA and ICCL.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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.

Richard Hudson. 1984. *Word Grammar*. Blackwell Publishers.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. 2024. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754.

Mary Katie Kennedy. 2025. Evidence of generative syntax in LLMs. In *The SIGNLL Conference on Computational Natural Language Learning*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Ilia Kuznetsov and Iryna Gurevych. 2020. A matter of framing: The impact of linguistic formalism on probing results. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. A dependency syntactic knowledge augmented interactive architecture for end-to-end aspect-based sentiment analysis. *Neurocomputing*, 454:291–302.

Tal Linzen. 2019. What can linguistics and deep learning contribute to each other? response to pater. *Language*, 95(1):e99–e108. Publisher Copyright: © 2019, Linguistic Society of America. All rights reserved.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Michael Milner, Helen Baron. 2024. Establishing an optimal online phishing detection method: Evaluating topological nlp transformers on text message data. *Journal of Data Science and Intelligent Systems*, 2:37–45.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2022. Probing for labeled dependency trees. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7711–7726, Dublin, Ireland. Association for Computational Linguistics.

Stefan Müller. 2019. *Superseded: Grammatical theory*. Number 1 in Textbooks in Language Sciences. Language Science Press, Berlin.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit.

Dmitry Nikolaev and Sebastian Padó. 2023. Investigating semantic subspaces of transformer sentence embeddings through linear structural probing. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 142–154, Singapore. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *ArXiv*, abs/1802.05365.

Michelle Sheehan and Sonia Cyrino. 2023. Restrictions on Long Passives in English and Brazilian Portuguese: A Phase-Based Account. *Linguistic Inquiry*, pages 1–35.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Giulio Starace, Konstantinos Papakostas, Rochelle Choenni, Apostolos Panagiotopoulos, Matteo Rosati, Alina Leidinger, and Ekaterina Shutova. 2023. Probing LLMs for joint encoding of linguistic categories. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7158–7179, Singapore. Association for Computational Linguistics.

Stanley Starosta. 1988. *The Case for Lexicase: An Outline of Lexicase Grammatical Theory*. Pinter.

Stanley Starosta. 1997. *Reconnecting Language : Morphology and Syntax in Functional Perspectives*, chapter Control in constrained dependency grammar. John Benjamins Publishing Company.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *Preprint*, arXiv:1905.06316.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2021. Dependency-based syntax-aware word representations. *Artificial Intelligence*, 292:103427.

Peter Öhl. 2015. Jörg hagemann sven staffeldt (hg.). 2014. syntaxtheorien. analysen im vergleich. *Zeitschrift für Rezensionen zur germanistischen Sprachwissenschaft*, 7(1-2):19–26.

A Stimuli

Our dataset utilized five structure conditions (Bare vP, Singular TP, Singular CP, Double TP, and Double CP). Our key verbs for the five conditions included:

1. Bare vP: see, hear, watch
2. Singular TP: require, allow, want
3. Singular CP: think, suspect, claim
4. Double TP: expect + {require, allow, want}
5. Double CP: believe + {think, suspect, claim}

Additionally, we varied the subjects for our sentences in order to vary the linear distance between the *wh*-word and the embedded verb. These subjects included:

1. Pronouns: you/I/she/he/they
2. Nouns: the {teacher/student/woman/man/people}
3. Modified Nouns: the {brilliant teacher/new student/clever woman/smart man/rowdy people}
4. Possessive Noun: {the modified noun}'s friend[14]

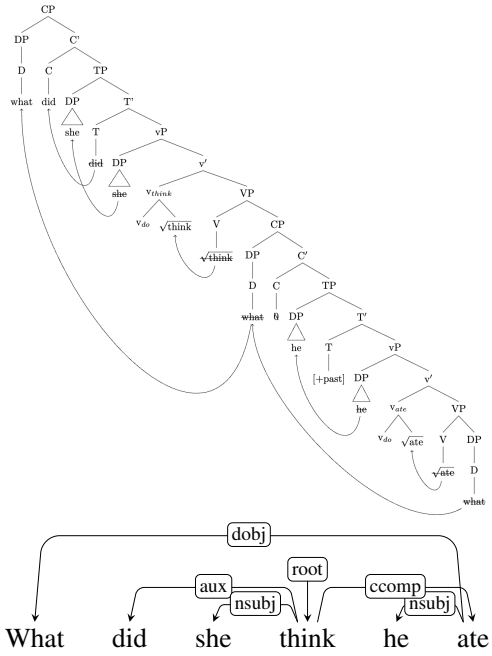
B Minimalist Trees

For illustrative purposes, we have utilized verb-flavors and roots from the school of Distributed Morphology. However, this is not of importance to the hierarchical distance as it is calculated from the merged result of the root and verb flavor. Other theoretical representation choices are a consequence of personal ideology, but does not impact the critical distinction that CP > TP > vP/VP. Two analyses for perception verbs are provided: one which utilizes a bare vP à la Sheehan and Cyrino (2023) (Example (9)) and one which utilizes a bare TP like Felser (1998) proposes (Example (8)). Our work favored the bare vP analysis—and furthermore found support for such an analysis—but a discussion on the two approaches can be found in Appendix C.

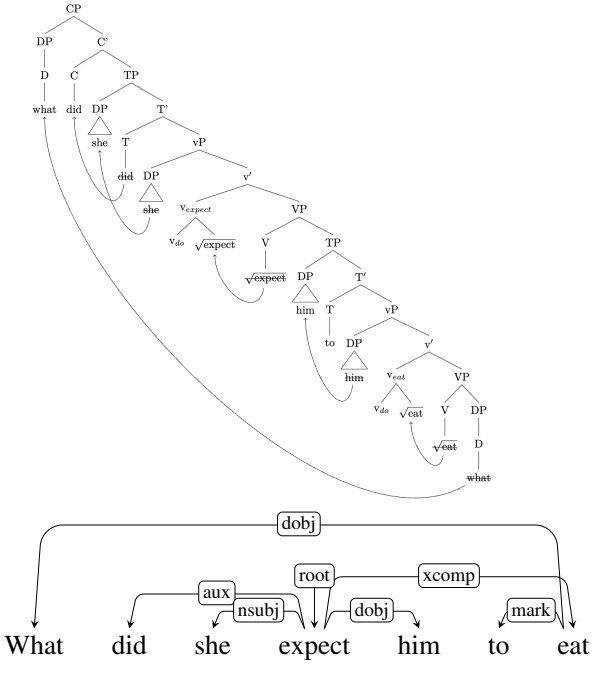
While not included, DoubTP and DoubCP trees contained hierarchical distance of approximately 18 and 22 and follow the same tree diagramming as illustrated in Examples (6)-(9).

[14]When necessary for BareVP and SingCP, an adverb was inserted before the mostly deeply embedded verb.

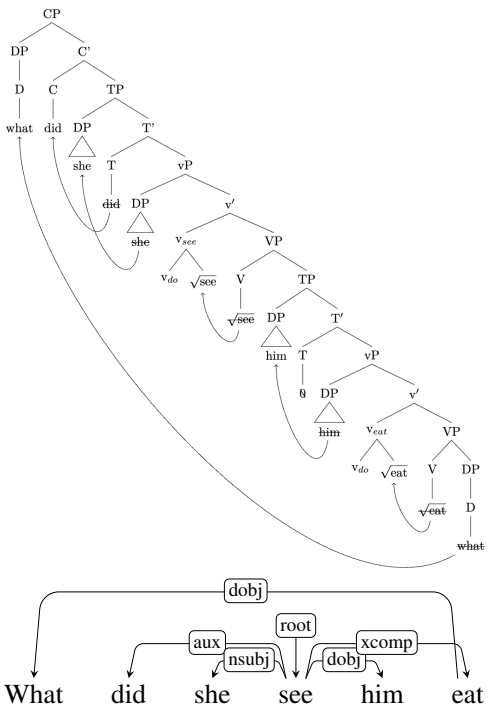
(6) CP Complement *("what" and "eat" constit dist ≈ 15; dep dist = 1)*



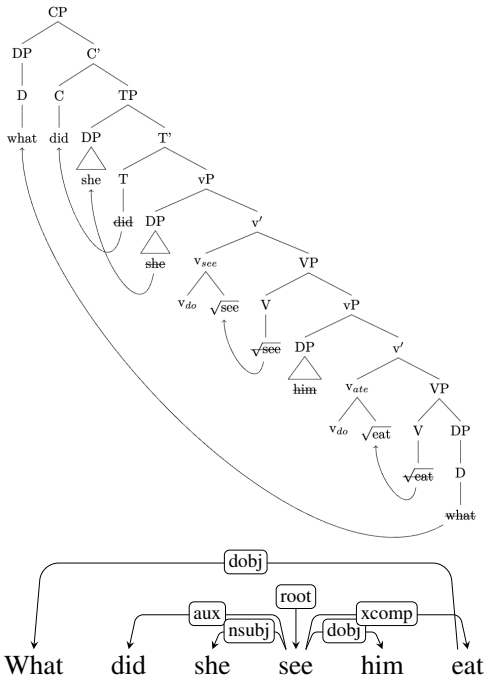
(7) ECM TP Complement *("what" and "eat" constit dist ≈ 13; dep dist = 1)*



(8) Bare TP Complement *("what" and "eat" constit dist* ≈ 13*; dep dist = 1)*



(9) Bare vP Complement *("what" and "eat" constit dist* ≈ 11*; dep dist = 1)*



C Further Analyses

Examining only TP and BareVP's difference from CP complements may not fully suggest that constituency structures are captured by pretrained language models. If we look only at vP/TP versus CP, it is possible that it is simply that BERT and the dependency probe are sensitive to finiteness, with CP being a finite phrase and vP/TP being non-finite.

Even under this possible interpretation, the implications for Dependency Grammar would be significant. Various theories of Dependency Grammar have postulated different treatments of the matter of finiteness; Lexicase (Starosta, 1988) and Word Grammar (Hudson, 1984) incorporate case relations in order constrain case assignment, which helps to assist in determining finiteness in English since finite verbs are generally conceived of as assigning nominative case in addition to incorporating features that help to distinguish the two structures (Starosta, 1997). However, the distinction between the two is not well discussed, and there exists no discussion that would explain why a verb embedded under a finite CP complement would be represented as being further away from a moved *wh*-constituent compared to a nonfinite TP or vP complement in the Chomskyan syntax. That CP complements show a further distance from their non-finite counterparts is already well captured and explained in constituency-based theories; that the dependency probe is sensitive to such distinctions in their representation is worth pursuing in the Dependency Grammar framework in order to explain this new data.

Additionally, the complements of perception verbs have been debated amongst constituent linguists (see Felser (1998) for bare infinitival TP argument and see Sheehan and Cyrino (2023) for bare vP argument analysis). Looking only at Figure 3, the distances for perception verb condition and singular ECM appear similar. However, analyses reveal statistically significant behavior in which ECMs showed significantly longer distances. Given that neither are finite, it becomes difficult to posit that the difference is due to some non-finite quality. This leads us to suspect that such differences are perhaps linked to a constituency-based analysis in which perception verbs take a complement whose size is smaller than that of the well-established TP phrase in ECM constructions, which lends support for the analysis in Sheehan and Cyrino (2023).

D Extra Figures and Results

The probe model frequently did not establish a dependency relationship between the direct object (the *wh*-question word) and the most deeply embedded verb, achieving undirected unlabeled accuracy scores far lower than those reported in Hewitt and Manning (2019), which ranged from 79.8%-82.5%, depending on the model probed. This low accuracy is likely due to various elements, such as the linear distance being a negative factor (accuracy worsens with increased linear distance, which is a well-known feature, or bug rather, of LLMs and their bottle-neck struggle to handle long-range dependencies) as well as questions being poorly represented in probe's training data and therefore more prone to inaccurate parsing. The probe's performance on the various conditions can be seen in Table 2.

In general, DoubTP achieves consistently low performance, even at the first initial and simplest iteration (0.218 for a sentence such as "What did you expect her to require him to eat?"), which is perhaps unsurprising as this construction is rather rare in natural data and is unattested in the probe's training data from the Penn Treebank (Marcus et al., 1993), which utilizes newspaper articles, which is inherently less likely to include questions, particularly those that are extracted out of doubly-embedded clauses. Similar performance appears—likely for similar reasons—with the doubly-embedded CP (DoubCP) which likewise performs poorly even at the simplest form (0.167 for a sentence like "What do you believe she thought he ate?"). Improving performance on these structures is worth further research.

LinDist		BarevP	SingTP	SingCP	DoubTP	DoubCP
4	Total	416		520		
	Corr.	144		285		
	Acc.	0.3462		0.5481		
5	Total	832	312	520		
	Corr.	185	150	137		
	Acc.	0.2224	0.4808	0.2635		
6	Total	728	312	624		840
	Corr.	148	104	199		140
	Acc.	0.2033	0.3333	0.3189		0.1667
7	Total	728	312	624		858
	Corr.	128	95	195		114
	Acc.	0.1758	0.3045	0.3125		0.1329
8	Total	728	624	624	702	858
	Corr.	58	261	88	153	114
	Acc.	0.0797	0.4183	0.1410	0.2179	0.1329
9	Total	728	624	624	702	858
	Corr.	178	160	180	68	141
	Acc.	0.2445	0.2564	0.2885	0.0969	0.1643
10	Total	728	624	624	702	858
	Corr.	174	134	172	47	82
	Acc.	0.2390	0.2147	0.2756	0.0670	0.0956

Table 2: The total number of sentences generated (Total) per condition per linear distance for the structural probe experiment. The number of sentences that correctly established a dependency between the *wh*-question word and the deepest embedded verb is also listed (Corr). Additional sentences were added as needed in order to achieve at least approximately 50 sentences. The percentage of sentences that correctly established the proper dependency relationship is also recorded (Acc.).