Evidence of Generative Syntax in Large Language Models

Mary Katie Kennedy
Linguistics
University of Southern California
mkkenned@usc.edu

Abstract

The syntactic probing literature has been largely limited to shallow structures like dependency trees, which are unable to capture the subtle differences in sub-surface syntactic structures that yield semantic nuances. These structures are captured by theories of syntax like generative syntax, but have not been researched in the LLM literature due to the difficulties in probing these complex structures that have many silent, covert nodes. Our work presents a method for overcoming this limitation by deploying Hewitt and Manning (2019)'s dependency-trained probe on sentence constructions whose structural representation is identical in a dependency parse, but differs in theoretical syntax. If a pretrained language model has captured the theoretical syntax structure, then the probe's predicted distances should vary in syntactically-predicted ways. Using this methodology and a novel dataset, we find evidence that LLMs have captured syntactic structures far richer than previously realized, indicating LLMs are able to capture the nuanced meanings that result from sub-surface differences in structural form.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable language capabilities that have been steadily increasing ever since BERT (Devlin et al., 2019). This impressive performance has prompted a body of research interested in investigating *why* these models are so successful. From this came a subset of research seeking to understand what, if any, linguistic features or knowledge these models have acquired (Jawahar et al., 2019; Belinkov and Glass, 2019; He et al., 2024; Kallini et al., 2024) as means to better understand their language performance. The focus of linguistic inquiry can vary from the semantic (Nikolaev and Padó, 2023; Kamath et al., 2024) to the morphological (Coleman, 2020; Anh et al., 2024) to the syntactic (Clark et al., 2019; Chi et al., 2020; Kulmizev et al., 2020; Maudslay and Cotterell, 2021), the latter of which our current research seeks to extend.

Much of the research into syntactic representation in LLMs have utilized dependency parses to represent a sentence's syntactic structure (Hewitt and Manning, 2019; Maudslay and Cotterell, 2021; Tucker et al., 2022; Eisape et al., 2022; Buder-Gröndahl, 2024). However, these relatively shallow representations can fail to capture features of a sentence and the nuanced differences in meaning that result from different sub-surface syntactic structures. Adopting a dependency framework makes theoretical assumptions and imposes limitations on the richness of meaning that can be expressed, the consequences of which are often not addressed. These simple, compact representations stand in stark contrast to the deeper, hierarchically-complex structures that are posited in theoretical syntax, particularly the generative frameworks, which postulate these complexities in order to account for the difference in semantic meaning and syntactic patterning of certain syntactic constructions. Because the structures posed by theoretical syntax are far more complex with more tree nodes than words in the sentence, attempts to probe for generative syntactic structures have been stymied, and it remains unclear whether LLMs have captured these richer sub-surface structures.

Our work seeks to overcome this through our unprecedented application of Hewitt and Manning (2019)'s dependency-trained probe to test for theory-backed syntactic structures. To implement this, we identified two sentences structures—Subject Raising (SR) and Subject Control (SC)—whose surface and dependency representations are identical, but *whose generative structures* ***differ*** as their complement sizes differ (control predicates take larger complements than raising predicates). Using this method, we are able to circumvent the issue of handling empty nodes in the syntactic tree

as the probe only recovers dependency distances. Our hypothesis holds that if the probe predicts distances for the two sentence types that significantly differ in syntactically-predicted ways, then this is evidence that LLMs have captured the more complex structures of generative syntax, which can yield structures to explain the pattern variations and semantic nuances of certain constructions.

To test this, we developed a novel dataset of over 33,000 SR/SC sentences, which we fed into our dependency-trained probes using pretrained models of **BERT**, **RoBERTa**, **GPT2**, and **Qwen2.5**. When we probe the content words, our results find strong evidence of the SR/SC difference that aligns with SC taking a larger complement than SR as generative syntax proposes. However, probing the distances with infinitival "to" suggest that the functional word may encode syntactic structure in an aberrant manner. Together, our work suggests LLMs have encoded structure that is more complex than previously realized, and provides a novel method to probe for theoretical syntactic structure in LLMs.

2 Related Work

The impressive language abilities of recent LLMs have prompted researchers to ask whether this performance is due to some probabilistic modeling, or if these language models have managed to capture linguistic structures. To answer this question, a line of research known as **probing** was developed. This methodology feeds the model's contextualized vector representations into a neural network whose training objective is to predict a targeted linguistic structure from the representations alone (see Alain and Bengio, 2017 or Conneau et al., 2018 for example). The argument follows that if such a neural network probe is in fact able to predict the target pattern or structure, then it can be concluded that the language model has indeed implicitly learned that linguistic feature; otherwise, the probe task would have been doomed to failure.

This area of research has largely focused specifically on investigating whether models have learned to properly encode syntactic phenomenon (Mueller et al., 2020; Hu et al., 2020; Warstadt et al., 2020; Ravfogel et al., 2021; Davis et al., 2022). However, much of this structural syntactic research has relied on dependency parses as a means of representing syntactic structure (Hewitt and Manning, 2019; Chi et al., 2020; Maudslay and Cotterell, 2021; Tucker et al., 2022; Eisape et al., 2022), with one notable exception being Arps et al. (2022), which sought to (and largely succeeded) in training a probe to reconstruct a skeletal constituency tree. While this line of research is of value and great interest, there are theoretical assumptions made by using dependency parses, and there are limitations to using that particular syntactic framework.

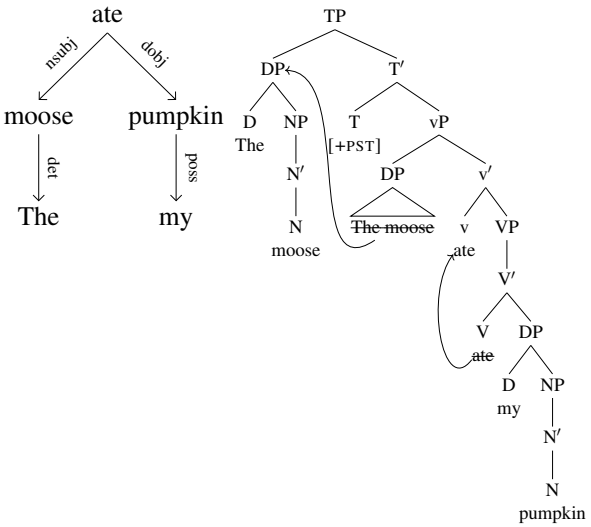

Figure 1: An example of the dependency tree (left) and generative syntax tree (right) for the sentence "The moose ate my pumpkin." Note how the dependency tree has a flatter structure with a one-to-one mapping of words to nodes in the tree. Compare this to the deeper generative tree where there are far more nodes in the tree than words in the sentence.

2.1 Syntactic Theories

Dependency parses derive from French linguist Lucien Tesnière's (1959) theory of syntax known as Dependency Grammar (DG), which focuses on the head-dependent relationship between words (see Figure 1). In these trees, each word can have one and only one incoming arc that indicates it is the dependent of its head, excepting the root of the sentence (often the matrix verb), which has no head.

DG trees are relatively flat structures with one-to-one mappings between words in the sentence and nodes in the tree. The appeal of such trees are largely three-fold: (1) the representations are compact and efficient due to the one-to-one mapping, (2) learning to parse a dependency tree is relatively easy once one understands the head-dependent relationships that exist, and (3) the dependency tree does not need to capture the sentence's linear order of words. The last factor makes DG an appealing theory for researchers working on languages with

freer word-order (Müller, 2019); however, this "feature" can become a bug when it loses nuance or creates ambiguous parses (see Figure 2).

An alternative, more structurally-rich approach to syntax has built off the theories of Chomsky (1957; 1981; 1986; 1995) and others who have refined this phrase-structure (also known as a constituency-based) framework to build up the syntactic framework known as generative syntax (GS). This family of syntactic theories are built on the X-bar theory, which proposes the operations *External Merge* and *Internal Merge* (formerly known as "Move"), and stipulates that nodes are binary-branching and that every phrase has a head (Chomsky, 1995). After all operations are applied in the course of derivation, the end result is the linearization of the sentence when read from left to right along the children nodes.[1] The generative framework is concerned with identifying the operations and rules that together generate licit sentences but do *not* generate illict constructions.

Unlike DG, generative syntax and other phrase-structure grammars thus yield deeper, more complex trees with hierarchical structures and phonologically null nodes whose presence must be deduced through testing. While this complexity is well-warranted (in that it can generate sentences that are grammatical and explain what causes ungrammaticality), the tree's size and complexity creates a complicated and unwieldy structure that is difficult for non-linguists to implement.

The result of this has been a limitation in the scope of feasible GS research in Natural Language Processing (NLP). Even work that has sought to test for the deeper, more complex phrase-structures in NLP has largely focused either on only seeking to recover a phrase's boundaries (Tenney et al., 2019; Kallini et al., 2024) or has otherwise trained their probe on the overly-simplified, *n*-branching constituency trees of the English Penn Treebank (PTB) (Marcus et al., 1993), which are only annotated with "skeletal" syntactic structure schema that is relatively atheoretical. The PTB's annotation is often used for automatic conversion into a dependency parse with little issue since the simplicity of PTB syntax is non-problematic for the simpler structures and principles of an NLP dependency parse. However, the result of this is a corpora that is not theoretically-sound for many of the deeper linguistic inquiries into phrase-structure grammars.

(1) Do you know who stole the Crown Jewels?

(2) Who do you know stole the Crown Jewels?

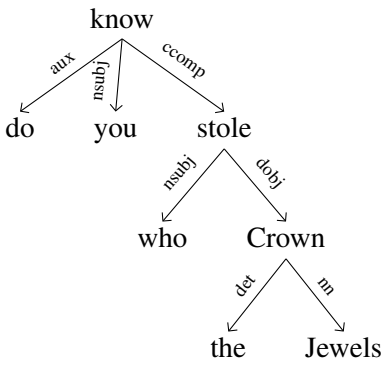

Figure 2: Above are two different sentences that yield *identical* dependency parses. While similar, the sentences have different meanings (imagine you are being questioned about the theft of the Crown Jewels: the first question merely seeks to inquire whether or not you know who the thief is, while the latter presumes you know who the thief is and seeks to learn the identity). This illustrates both the ways in which dependency trees do not capture linear order, and highlights some of the limitations of dependency trees.

Because of this, it has yet to be discovered whether LLMs have managed to capture the deeper, hierarchical structures of GS. However, there are three major barriers to testing whether models have captured the richer hierarchical structures as proposed by generative frameworks:

1. GS and similar frameworks often have "empty" nodes that are not overtly realized.[2] As such, they are not overtly present in the texts LLMs train on, and so probing at their presence is difficult because it raises the question: how can you probe at something that is not overtly represented?

2. Probes largely require a gold tree that indicates the correct structure or parse. Human annotation, while crucial when handling such

[1] In earlier versions of GS, it was argued that when an element moves, it leaves behind a coindexed trace element (Chomsky, 1973; Fiengo, 1977). Because such an derivation introduces a *new* element, Chomsky (1993) revised the approach to the Copy Theory of Movement, claiming that movement leaves behind a copied element that is not phonologically realized. Thus, the sentence used above would be "Do you [do] [you] know [know] who [who] [who] stole [stole] the Crown Jewels?" For a discussion on which elements are phonologically realized and why, see Corver (2007).

[2] This can be due to movement (see Footnote 1) or the feature not having an overt representation (e.g., there is no specific word or morpheme that indicates present tense for plural subjects as in "They $walk_{pres}$ to the store"). See Figure 1 for demonstration.

fine-grained analysis, is laborious and costly in resources and time.

3. Even if one is able to secure the resources necessary to create such a gold standard, there are competing theories even within the generative framework that would change a sentence's representation. As such, a gold parse would be subject to great theoretical scrutiny and likely face present or future dissenting opinions.

Our research develops a method that circumvents these obstacles while still addressing the fundamental research question of whether LLMs have captured something of the deeper, sub-surface syntactic representation theorized by many linguists.

3 Methodology

3.1 Probing Method

To combat the first issue of accounting for structures that are not phonologically realized, we have opted for the novel approach of re-purposing the original Hewitt and Manning (2019), which was trained to recover dependency trees, to investigate whether LLMs have encoded theoretically-motivated generative phrase-structure trees.

The structural probe developed by Hewitt and Manning (2019) proposes a model M that encodes a sequence of vector representations $\boldsymbol{h}^l_{1:n}$ from an input sequence of n words $\boldsymbol{w}^l_{1:n}$ where l identifies the sentence index. From there, they define a linear transformation matrix $\boldsymbol{B} \in \mathbb{R}^{k \times n}$ to parameterize 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 transformation matrix B's objective is to reproduce the gold parse distances between each pair of words (w^l_i, w^l_j) for all sentences l in the parsed training corpus T^l. The training uses a gradient descent objective:

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

In this equation, $\mid s^l \mid$ is the length of sentences, which the function normalizes using the square of the sentence's length as each sentence contains $\mid s^l \mid^2$ pairs of words. The probe's objective thus seeks to approximate a matrix of distances that most closely resembles the gold-standard distances. Because Hewitt and Manning (2019) use a dependency parse, gold-standard parses were converted into gold-standard distance matrices where distances are defined such that the distance between a parent node and its child nodes is 1, the distance between a child node and its grandparent node is 2, the distance between a child node and a so to speak "aunt" or "uncle" node is 3, and so on and so forth. Evaluation of the probe involved calculating the minimum spanning tree for each sentence's predicted distances to derive the sentence's predicted undirected, unlabeled attachment score (UUAS) compared to the gold tree, and the average Spearman correlation of the predicted matrix of distances compared to the gold-standard matrix.

We chose this method specifically *because* it is a probe trained *only* to capture dependency parses with their one-to-one mappings between a sentence's words and a tree's nodes. Though there are critical limitations to dependency parses as discussed in Section 2.1, we argue that its simplicity and overgeneralization can in fact be converted into a benefit. It is because the probe is superficially only supposed to capture shallow-level, generalized syntactic structures that we can use the method to tease apart syntactic structures whose representations are **identical** in a *dependency* parse but **vary** in a *generative* framework.

3.2 Syntactic Structures of Interest

Our method hinges on testing syntactic structures whose representations are crucially different in generative accounts, but are invariant in a dependency parse. In doing so, we propose turning the limitations of a dependency probe to an asset. If the probe's predicted dependency distances vary between the sentences in question in ways that align with generative theoretical predictions, then we have evidence that not only do LLMs' contextualized vector representations capture generative syntactic structures, but that a probe trained only to recover dependency parses is additionally sensitive to hierarchical phrase-structure distances.

To test this, we have selected the well-researched Subject Control (SC) and Subject Raising (SR) constructions as our experimental condition. Observed first by Rosenbaum (1967), SR constructions are those that consist of two clauses: a matrix clause and an infinitival Tense Phrase (TP) complement. Since its initial observation by Rosenbaum (1967), it's largely been accepted that the subject position

of the embedded clause is occupied by a trace element (later revised to a copy element, see footnote 1) due to the subject being raised into the matrix clause by the EPP features[3] in the matrix clause. SC constructions, meanwhile, are assumed to take a larger complement than a raising verb, with many typically assuming an SC complement to be a Complement Phrase (CP). Many theories follow Chomsky and Lasnik (1993) and posit a silent PRO element that is co-indexed with and controlled by the matrix's subject. This PRO receives its theta-role from the embedded verb while the matrix subject receives its theta-role from the matrix verb, thus satisfying the Theta Criterion (Chomsky, 1957).[4]

For the purposes of our experiment, the crucial things to know are: Subject Raising takes a Tense Phrase (TP) as its complement, while Subject Control takes the larger Complement Phrase (CP) as its complement, which inherently contains a TP itself. Thus, the result are two structures whose surface forms and dependency parses are *identical*, but whose hierarchical syntactic representations are **different**. Thus, we would expect that if the LLM has not acquired any knowledge of deeper syntactic representations or if the dependency-trained probe is insensitive to phrase-structure representations, then the probe's predicted distances between relevant word-pairs should *not differ* between the two structures. However, if such hierarchical representations are indeed captured and if the probe is sensitive to these structures, then we would anticipate that the distances between certain word-pairs in an SC construction are **longer** than the equivalent word-pairs in an SR construction due to SCs containing the larger CP complement as opposed to the smaller TP complement of SR predicates.

4 Experiments

4.1 Generating Data

For our experiment, we identified 6 SR verbs and 6 SC verbs, which we permutationally paired with a set of 8 subject words, 61 embedded verbs, and a set of possible direct objects (either a single pronominal direct object or a two-word definite

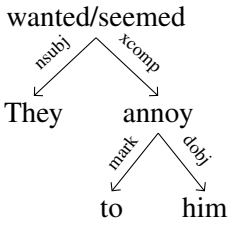

Figure 3: Dependency parse for the two sentences "They seemed/wanted to annoy him." The two trees are identical, and the distance between the subject and embedded verb is 2 while the distance between the subject and the infinitive or direct object is 3. This is true even if one were to use extended Universal Dependencies, which also conflates SC and SR verbs.

object that was matched to a specific embedded verb). Thus, we yielded 33,120 unique sentences, such as "They wanted/seemed to annoy him."

Metrics Should the LLMs *not* have any awareness of the deeper hierarchies or should the probe be insensitive to such differences, then should be *no difference* between SR's and SC's distances between words in the matrix clauses and words in the complement clauses. However, if such structures are captured and if the probe is sensitive to this, then we anticipate that the distance between a word in the matrix clause and a word in the complement clause will be **longer** in an SC construction compared to an SR construction since the CP complement is larger (see Figure 4).

For this reason, we opted to investigate the probe's predicted distances between the following word-pairs: subject and the infinitive (`subj-infin`, e.g., "they" and "to"), subject and the embedded verb (`subj-embed`, e.g., "they" and "annoy"), subject and the direct object (`subj-dobj`, e.g., "they" and "him"), and lastly, embedded verb and the direct object (`embed-dobj`, e.g., "annoy" and "him"), which serves as our baseline. We should acknowledge at this point that excepting our baseline comparison, none of our word-pairs have any direct dependent or syntactic relationship to each other. This is not a problem. Recall that the probe was trained on a the gold parses for dependency trees where the distance between two nodes can be counted as the number of edges between the two. Because of this design, we are able to probe the distances between the words in the matrix clause and the words in the complement clause despite there being no direct dependency or syntactic relationship.

As the dependency parses do not differ between

[3] Chomsky (1995) proposed the Extended Projection Principle, which stipulates that Tense bears a strong D-feature that requires a subject in its Specifier. This can be satisfied by either moving the subject to Spec,TP or by inserting an expletive like "it."

[4] For further discussion on Subject Control and Subject Raising and their structural and semantic differences, see Appendix A.

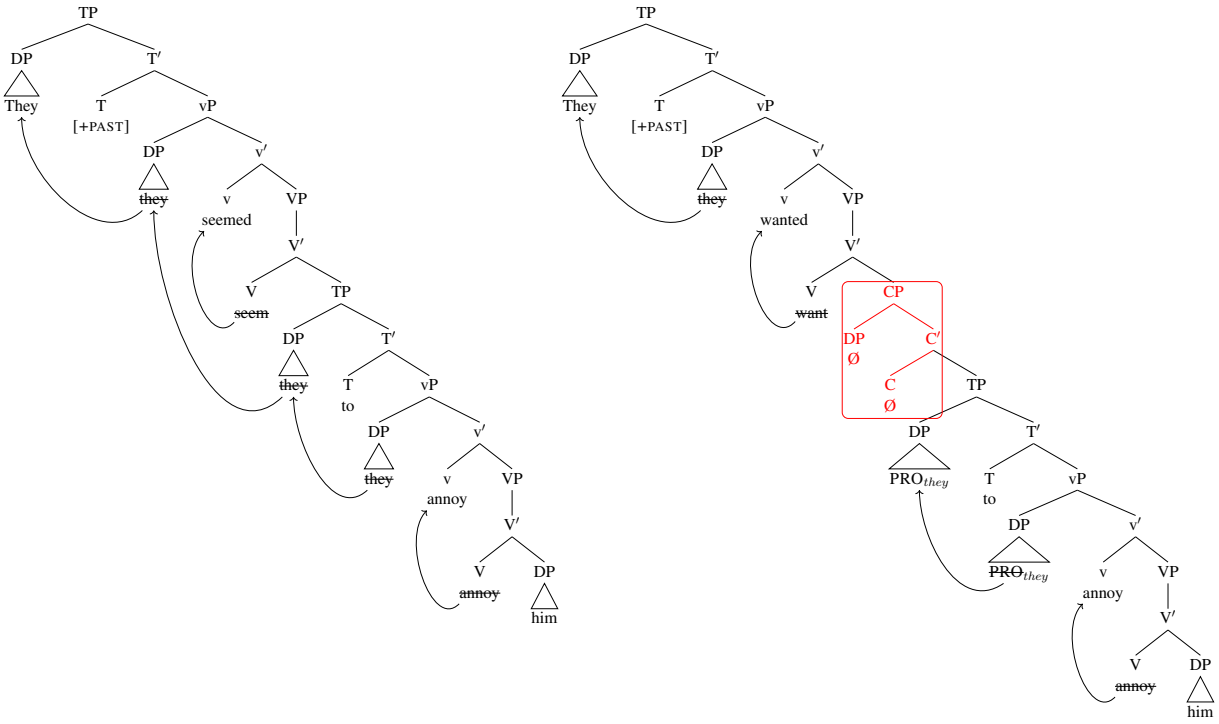

Figure 4: Syntactic trees of the SR sentence "They seemed to annoy him" (left) and the SC sentence "They wanted to annoy him" (right). The two structures are nearly identical, except SC contains a CP above the TP (red), which makes the hierarchical distances between the subject and the embedded clause's elements (i.e., the infinitive "to", the embedded verb "annoy", and the direct object "him") longer in the SC sentence.

the two structures, the gold-parse distances are also stable: `subj-embed` has a dependency distance of 2 while `subj-infin` and `subj-dobj` have dependency distances of 3 (see Figure 3). For this reason, if the LLMs do not capture generative syntactic hierarchies or if the probe is insensitive to such differences, then we should see *no difference in predicted distances between the two experimental conditions*. If, however, the models **do** capture this deep structural difference and if the probe is an adequate tool to measure this, then we should anticipate that the SC distances should be **longer** than their equivalent SR distances. To verify that our probe is working as anticipated, we included the baseline word-pair `embed-dobj`, which should *not* show any differences in distances as these words are *not* affected by the SC/SR distinction.

4.2 Experimental Setup

Models We probed three pre-trained Transformer (Vaswani et al., 2017) models: **BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), **GPT-2** (Radford et al., 2019), and **Qwen2.5** (Team, 2024). We constrained our probing to models with hidden dimensions of 768 and 1024, which corresponded to the `bert-base-cased`, `roberta-base`, and `gpt2-small` for the smaller models and `bert-large-cased`, `roberta-large`, and `gpt2-medium` for the larger models. Per the suggestions of our reviewers, we also included two newer models: `qwen2.5-0.5` and `qwen2.5-1.5` (referred to in this paper as "Qwen2.5Baby" and "Qwen2.5Small", respectively). All models were accessed using the Huggingface Transformers library (Wolf et al., 2020), and the probe was developed using the parsing train/dev/test splits of the Penn Treebank (Marcus et al., 1993).

Following Hewitt and Manning (2019),[5] a probe was trained to convergence (maximum of 40 epochs) on each layer with a batch size of 20. Analysis was conducted on the best-performing layer.

Once the best-performing layer[6] was selected, we fed our novel dataset to that probe and obtained the predicted distances for our word-pairs of interest. Analysis was conducted on the predicted

[5] Hewitt and Manning (2019)'s original code can be found at `https://github.com/john-hewitt/structural-probes`, which includes the **BERT** models. A helpful starting point to modify the code for **RoBERTa** and **GPT2** can be found at `https://github.com/leoier/structural-probes`.

[6] See Appendix C, Figures 5, 6, and 7 for model performances.

distances for our word-pairs *if and **only** if* the probe properly established the necessary dependency relationships. That is to say, if the probe *misparsed* the tree in a relevant manner, that word-pair's predicted distance was excluded from analysis. Using Figure 3 as a gold-parse, if the probe's minimum spanning tree situated "him" as a dependent of "wanted," then we excluded the `subj-dobj` word-pair as the tree was misparsed in a critical way for that word-pair. We currently do not have strong reason to suspect that such a misparse would affect non-impacted word-pairs; therefore, the `subj-embed` and `subj-infin`'s predicted distances would still be used for analysis since the probe would have correctly parsed the subject and embedded verb as being dependents of "wanted" and parsed the infinitive as the dependent of "annoy."[7]

5 Results

As mentioned, we generated 33,120 sentences for which we gathered a total of 935,419 distances across our four word-pairs and all eight language models. Overall, this represents an 88.05% accuracy score for correctly parsed word-pairs. The accuracy for our four word-pairs can be found in Table 1 where we may observe that while the accuracy for the SC condition is slightly higher, both showed high accuracy with the lowest being attributable to the `subj-dobj` word-pair, which was due to the direct object not being tied to the embedded verb, hence the equivalent scores with `embed-dobj`.

Due to the large size of the data, we split the data by word-pair for statistical analyses. Mixed effect models were developed with the lmer function from lme4 (v. 1.1-31) (Bates et al., 2015) and lmerTest (v. 3.1-3) (Kuznetsova et al., 2017). Fixed effects were identified as the condition (SC or SR) and the linear distance (the number of intervening words plus 1 to avoid issues of 0 multiplication) as well as their interaction. The latter two only applied to word-pairs with the direct object as the direct object could be a single pronominal like "it" (in which case the linear distance would be 1) or a full nominal phrase like "the car" (in which case the linear distance would be 2). For the other word-pairs, there was no variation in linear distance, hence its exclusion as a fixed effect. Condition was contrast-coded with SC being -0.5 and SR being 0.5.

Model comparison via anova was used to determine the random effects of by-MatrixVerb, by-SubjectWord, by-EmbedVerb, by-ObjectWord, and by-LanguageModel as well as random slopes for Condition (and LinDist for `dobj`). For the full linear models, see Appendix B.

To recap, our hypothesis is that the probe's predicted distances between the matrix subject and elements in the embedded clause (i.e., the infinitive, the embedded verb, and the direct object) should be **longer** in the SC condition compared to the SR condition. Should this be the case, this effect should appear in all of our word-pairs (excepting our baseline of `embed-dobj`). In this regard, our study uses conjunction testing in that we require all tests be significant in order to reject the null hypothesis (Weber, 2007). We thus follow Rubin (2021) and do not adjust our alpha level.

Table 1 reports our results where we find a main effect for Condition in our `subj-embed`, `subj-infin`, and `subj-dobj` data ($p = 2.77e^{-5}$, $p = 0.035$, and $p = 1.90e^{-12}$). Thus we find evidence that the predicted distances from an SC construction are significantly longer than an SR construction when considering the distance between the subject and a word within the embedded clause.

Crucially, we do **not** find Condition to be a significant predictor for our baseline, suggesting the probe is not spuriously attributing higher distances to SCs than SRs in ways that are not predicted by the syntax. However, interaction between Linear Distance and Condition *is* found to be a main effect for `embed-dobj`. To conduct follow-up models to investigate this result, we split the data by linear distance, meaning sentences were grouped into those that took a pronominal direct object such as "it" (linear distance of 1) and those that took a nominal phrase object such as "the car" (linear distance of 2). In doing so, we do **not** find Condition to be a main effect in *either* group.

Analysis of the data further reveals that linear distance decreases the predicted distance for subject raising verbs only. When we control for linear distance by splitting up the data by direct object type, though, our follow-up analyses find that the predicted distance between the embedded verb and the direct object do not significantly vary between

[7] For our current study, the best-performing layer was selected as the probe with the highest UUAS. As such, we opted only to use word-pairs in which the minimum spanning tree established the correct necessary dependencies for the word-pair in question. However, future work may investigate selecting the probe based on the Spearman correlation, in which case, the motivation to reject data based on improper parses disappears as the Spearman metric does not utilize the minimum spanning tree and instead seeks to globally reduce the differences between the gold distances and the predicted distances.

				Fixed Effects					
WordPair	**Condition**	**Acc**	**PredDist** *(avg)*	Coefficient	$\hat{\beta}$	SE($\hat{\beta}$)	df	t	p
Subj-Embed	Cont	96.01%	2.04	(Intercept)	1.93655	0.10610	13.24239	18.253	8.99e-11
	Raise	94.35%	1.82	Condition	-0.22879	0.04147	18.53383	-5.517	**2.77e-05**
Subj-Infin	Cont	89.50%	2.86	(Intercept)	2.81437	0.07561	13.77291	37.221	3.21e-15
	Raise	84.75%	2.74	Condition	-0.13397	0.05517	10.25726	-2.428	**0.035**
Subj-Dobj	Cont	86.29%	2.98	(Intercept)	2.823	0.3260	24.68	8.659	6.01e-09
	Raise	84.45%	2.76	Condition	-4.805e-01	4.309e-02	31.38	-11.152	**1.90e-12**
				LinDist	2.572e-03	6.602e-02	22.62	0.039	0.969
				Interaction	6.023e-02	5.436e-03	2.260e+05	11.081	**<2e-16**
Embed-Dobj	Cont	86.29%	1.50	(Intercept)	1.54498	0.08508	75.04190	18.160	<2e-16
(baseline)	Raise	84.45%	1.50	Condition	-0.04512	0.02990	41.13811	-1.509	0.138948
				LinDist	-0.04085	0.05312	44.15045	-0.769	0.445915
				Interaction	0.03901	0.01095	48.82859	3.561	**0.000835**

Table 1: Results of the probes' predicted squared Euclidean distances between the word-pairs of interest. Accuracy records what percentage of the sentences properly established the necessary dependency relationships for that particular word-pair. The right side of the table reports the fixed effects findings for the linear mixed-effect models that were built for each word-pair. See Appendix C, Figures 8 and 9 for visuals.

the two conditions ($p = 0.816$ for pronominal direct objects and $p = 0.238$ for nominal phrase objects).

The significantly longer predicted distances of the SC condition in `subj-embed`, `subj-infin`, and `subj-dobj`, paired with Condition *not* being a significant predictor for our baseline comparison of `embed-dobj` (even when accounting for interaction effects), together show strong evidence to reject our null hypothesis.

5.1 Results by LLM

While our results indicate that the probe is sensitive on some level to syntactic hierarchies, this is not equally true for all word-pairs across all models. As can be seen in Table 2 in Appendix C, a significant main effect for the SC/SR Condition for word-pairs `subj-embed` and `subj-dobj` was found for the probes of all models *except* `gpt2-medium`, which revealed Condition to be marginal ($p = 0.0818$), though the reason for this is unclear. As for `subj-infin` however, Condition was significant *only* for the `Qwen2.5-1.5` model ($p = 0.0419$), and `roberta-large` ($p = 0.00345$).[8] For all other models, Condition was *not* a significant predictor.

If the structures proposed by generative syntax to account for Subject Raising and Subject Control are indeed captured by LLMs and subsequently by the probe, then we would anticipate that *all* three word-pairs of interest across *all* LLMs should show significantly longer predicted distances in the SCs compared to SRs while the baseline comparison of `embed-dobj` (which is *not* affected by an SC or SR construction) should not. While these predictions are largely borne out by `subj-embed` and `subj-dobj` along with our baseline of `embed-dobj`, it does not hold true for `subj-infin` for many models.

This finding is particularly puzzling. Should it be that the LLMs do not capture the SC/SR distinction, then *none* of the word-pairs should have significant differences in distances rather than just one (namely, `subj-infin`). Additionally, there is no theory in any school of syntax (generative or otherwise) we are aware of that suggests SC verbs take larger complements *below* the TP head of "to." We might then posit that the infinitive's seemingly imperviousness to the SC/SR distinction may arise from these LLMs somehow building a novel and alien structure in which the infinitival "to" sits in the matrix clause while the complement size distinctions are displayed beneath it. Again, however, we resist this notion as we know of no theory postulating such an arbitrary and alien structure.

It is evident this matter requires further investigation, but it is possible the aberrant behavior of the infinitive is due to the nature of infinitives themselves. Infinitival "to" is semantically vacuous: there is little to any rich semantic meaning to the word, which is entirely functional in nature—denoting either non-finiteness as an infinitive[9] or directionality or telicity as a preposition. For this reason, we suspect the lack of semantic-richness of purely functional words may impact how structure is captured by embedding vectors.[10]

[8] For `gpt2-small`, Condition was marginal ($p = 0.0688$).

[9] See (Satik, 2022) for discussion on the subtle semantic differences between different types of infinitives.

[10] We exempt pronouns from this hypothesis. Our dataset subjects were pronominal and our single-word direct objects were also pronouns. Unlike infinitival "to," pronouns pick out referents in the real world, and can furthermore carry

We also cannot attribute the lack of significant findings amid subj-infin to model complexity or novelty. While the model with the highest number of parameters (qwen2.5-1.5 at 1.54B parameters) did find Condition to be a significant predictor for subj-infin, so did the 340m parameter roberta-large. Despite this, models with similar parameter sizes as roberta-large did *not* find the SC/SR distinction to significantly predict the distance for subj-infin. Nor can we suggest that it is the newer models whose embedding representations capture linguistic aspects that correlate to syntactic hierarchy; the 2024 Qwen2.5-0.5 failed to find Condition to be significant for subj-infin. Further research is needed done to understand why only a select few models' (Qwen2.5-1.5 and roberta-large) embedding representations for infinitives appear to capture linguistic aspects corresponding to syntactic hierarchies.

6 Discussion

While the matter of infinitives remains murky, our findings suggest that models are capable of encoding some linguistic aspects corresponding to the syntactic hierarchies as proposed in generative syntax. That SC verbs yield longer predicted Euclidean distances than SR (as opposed to the reverse) already aligns with generative theories that control verbs take larger, more complex complements (Chomsky and Lasnik, 1993; Landau, 2007, 2013, 2024) than raising verbs.

It may be asked whether these results are merely a product of semantics rather than syntax. However, we maintain that seeking to entirely divorce syntax from semantics should not be the main goal. As Leivada and Murphy (2021) comments, "syntax, semantics, and the other levels of linguistic analysis are not undecomposable modules that work autonomously," which makes it difficult to separate the two when researching the neural processing of the human mind, and, we argue, when researching the artificial neural processing of an LLM. Our findings may be due to hierarchical distance being larger in SC, or it may be due to the generative syntactic theory that SCs assign an extra theta role(Chomsky and Lasnik, 1993; Landau, 2024; Beockx and Hornstein, 2010). Both explanations speak to LLMs being able to encode deeper linguistic aspects that interface with syntactic structure.

information such as Case, Gender, and Number as opposed to infinitival "to," which indicates non-finiteness only.

In order to determine if these findings are in fact indicative of syntactic hierarchical distance or merely a quirk of the SC/SR constructions, future work should aim to test other syntactic structures. Preliminary work by Kennedy (2025) tests *wh*-extraction from different sized complements (e.g., "What did she see him eat" vs "What did she expect him to eat" vs "What did she think he ate") and finds that Hewitt and Manning (2019)'s probe's predicted distance between the extracted *wh*-word and its embedded verb (e.g., "What" and "eat") increases as the size of the complement increases. With continued research like this, should multiple different sentence structures all converge on larger syntactic hierarchical distances yielding longer predicted probe distances, then we can say with even greater confidence that LLMs are capable of encoding linguistic attributes that correspond to the structures propose by generative syntax.

7 Conclusion

The implications of this work have impact on both the field of NLP and the field of linguistics. Our work suggests that LLMs have learned to capture elements of deeper and more complex syntactic structures within their embeddings than previously realized and thus have the ability to capture the semantic nuances that result from sub-surface structural differences. Our findings therefore further the interpretability research of LLMs to discover what these models have actually learned regarding the features and structures of language. We also find evidence that neural networks trained using the dependency framework can still capture deeper syntactic features, suggesting these simpler representations may be adequate for downstream tasks as they appear to be capable of reaping the benefits of deep structure without needing to explicitly train on deep structure. As for linguists, the findings of our work warrant further investigation into the viability of using language models as a means to test syntactic structures. Our work begins to open up the possibility of utilizing LLMs as another source of data to help augment, build, and perhaps even test syntactic theories.

Taken together, we situate our work as a realization of Linzen (2019) and Futrell and Mahowald (2025)'s claim that the skillsets and knowledge of the fields of NLP and linguistics complement each other, and that the two stand primed to advance each other's respective fields through collaboration.

Limitations

Our work still faces limitations in that it does not enable a full reconstruction of the hierarchical syntactic tree. This is a limitation currently inherent to the data and format of LLMs. As can be seen in Figure 4, the generative syntax trees consist of branches and nodes that do not overtly appear in the final derivation. That is to say, trace nodes/moved elements are not surfaced, nor are all syntactic elements (such as tense) realized by a separate word. Because of this, an LLM's contextualized word embeddings cannot currently be used to directly derive the sub-surface syntactic trees. The methodology that we've deployed allows us to probe for behaviors that would indicate that LLMs have captured more complex, hierarchically-rich structural information within their embeddings, but this cannot be directly shown the way Hewitt and Manning (2019) did with the one-to-one mappings of dependency parses. Thus, our work is still largely in the tradition of much of linguistics. We cannot directly observe people's mental grammars, but we probe for their knowledge and structures using measurements that indicate how people process and produce language. Similarly, our use of Hewitt and Manning (2019)'s probe also provides an apparatus to measure behaviors that we can use to reverse-engineer the possible behaviors and mechanisms that would derive such results. The interpretability question of LLMs is not far at all from the research questions of linguistics.

Acknowledgments

I would like to thank Khalil Iskarous, Jon May, Jesse Thomason, Andrew Simpson, and Travis Major 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 experimentation with newer models and a deeper engagement with the results.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *The 5th International Conference on Learning Representations*.

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.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

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

Cedric Beockx and Norbert Hornstein. 2010. *Control as Movement*. Cambridge University Press.

Tommi Buder-Gröndahl. 2024. What does parameter-free probing really uncover? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 327–336, Bangkok, Thailand. Association for Computational Linguistics.

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. 1993. A minimalist program for linguistic theory. In Kenneth Locke Hale and Samuel Jay Keyser, editors, *The View From Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*. MIT Press.

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

Noam Chomsky and Howard Lasnik. 1993. Syntax: An international handbook of contemporary research. In Joachim Jacobs, Arnim von Stechow, Wolfgang Sternefeld, and Theo Vennemann, editors, *The theory of principles and parameters*. De Gruyter.

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.

N.F.M. Corver. 2007. From trace theory to copy theory. In *The Copy theory of movement*, pages 1–10, Netherlands. John Benjamins.

Mark Davies. 2008–. The corpus of contemporary american english (coca).

Christopher Davis, Christopher Bryant, Andrew Caines, Marek Rei, and Paula Buttery. 2022. Probing for targeted syntactic knowledge through grammatical error detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 360–373, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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.

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

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *Preprint*, arXiv:2501.17047.

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 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.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

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 Kennedy. 2025. Evidence of hierarchically-complex syntactic structure within BERT's word representations. In *2025 Meeting of the Society for Computation in Linguistics*.

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.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26.

Idan Landau. 2007. *New Horizons in the Analysis of Control and Raising*, chapter Movement-Resistant Aspects of Control. Springer.

Idan Landau. 2013. *Control in Generative Grammar: A Research Companion*. Cambridge University.

Idan Landau. 2024. *Elements in Generative Syntax*, chapter Control. Cambridge University.

Evelina Leivada and Elliot Murphy. 2021. Mind the (terminological) gap: 10 misused, ambiguous, or polysemous terms in linguistics. *Ampersand*, 8:100073.

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.

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

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.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

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

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.

Maria Polinsky. 2013. *The Cambridge Handbook of Generative SYntax: Grammar and Syntax*, chapter Raising and Control. Cambridge University Press.

Paul M. Postal. 1974. *On raising: One rule of English and its theoretical implications*. MIT Press.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Peter Rosenbaum. 1967. *The grammar of English predicate complement constructions*. MIT Press.

Mark Rubin. 2021. When to adjust alpha during multiple testing: A consideration of disjunction, conjunction, and individual testing. *Synthese*, 199:10969–11000.

Deniz Satik. 2022. The semantics of infinitival tense. Under review.

M.E. Sánchez, Y. Sevilla, and A. Bachrach. 2016. Agreement processing in control and raising structures. evidence from sentence production in spanish. *Lingua*, 177:60–77.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *Preprint*, arXiv:1905.06316.

Lucien Tesnière. 1959. *Eléments de Syntaxe Structurale*. Klincksieck, Paris.

Mycal Tucker, Tiwalayo Eisape, Peng Qian, Roger Levy, and Julie Shah. 2022. When does syntax mediate neural language model performance? evidence from dropout probes. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5393–5408, Seattle, United States. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Rene Weber. 2007. Responses to matsunaga: To adjust or not to adjust alpha in multiple testing: That is the question. guidelines for alpha adjustment as response to o'keefe's and matsunaga's critiques. *Communication Methods and Measures*, 1:281–289.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Data Generation

Our data was generated through combinatorics of sets of words for each grammatical role. In short, our sentences followed the base structure of:

(3) [Subject] [past-tense matrix verb] [to] [embedded verb] [direct object].

In order to easily control linear distance, subject verbs were limited to pronominal subjects. Because control verbs are typically volitional, all subjects were prototypically [+HUMAN], but varied in Case, Gender, and Number (see List (4)).

We additionally selected 61 transitive verbs for our embedded verb (see List (5)). Of these verbs, 30 verbs implied human direct objects while 31 implied non-human direct objects. That is to say, a person can *flatter* the king, but it's nonsensical for them to *drink* the king. Conversely, they can drink sodas, but it would be hard to flatter an inanimate soda. This dichotomy was taken into account when selecting direct objects. Thus, when the direct object was a single-word pronominal, inanimate-coded verbs permutated through *it, that, this, stuff,* and *things* while animate-coded verbs permutated through *me, you, him, her, us, them, everyone,* and *someone*. The animate list is longer; however, the animates were truncated as we omitted direct objects that were the correspondent of the subject. That is to say, if the subject was "she", the direct object would *not* be *her*. Additionally, to avoid scope ambiguities, we excluded instances where the subject was "someone" and the direct object was "everyone".[11] Nominal direct objects ("the" + the noun) were more limited as we selected only one plausible noun to pair with the embedding verb.

(4) **Subjects**: You, He, She, We, They, Everyone, Someone

(5) **Embedded Verbs**

a. *Inanimate-coded Verbs*: say, yell, whisper, shout, think, write, read, cook, eat, drink, buy, sell, rent, provide, offer, collect, grab, steal, bump, move, kick, break, destroy, build, wash, wear, sew, mend, fix, enjoy

b. *Animate-coded Verbs*: kiss, hug, slap, wrestle, fight, bully, harass, intimidate, insult, slander, annoy, tease, seduce, flatter, comfort, compliment, question, interrogate, interview, meet, fire, hire, pay, reward, punish, scold, teach, train, serve, admire

(6) **Pronominal Direct Objects**

a. *Inanimates*: it, that, this, stuff, things

b. *Animates*: me, you, him, her, us, them, everyone, someone

(7) **Nominal Direct Objects and Their Corresponding Embedded Verb**: say the words, yell the answer, whisper the clues, shout the lyrics, think the worst, write the essay, read the book, cook the meal, eat the food, drink the sodas, buy the clothes, sell the toy, rent the apartment, provide the supplies, offer the bribes, collect the rocks, grab the keys, steal the gold, bump the table, move the chairs, kick the ball, break the glass, destroy the house, build the tower, wash the socks, wear the uniform, sew a shirt, mend the tears, fix the issue, enjoy the dessert, kiss the puppy, hug the baby, slap the clown, wrestle the children, fight the administration, bully the student, harass the reporter, intimidate the intern, insult the actress, slander the politician, annoy the teenagers, tease the toddlers, seduce the actor, flatter the king, comfort the victims, compliment the model, question the judge, interrogate the witness, interview the suspect, meet the manager, fire the employee, hire the applicant, pay the consultant, reward the winner, punish the cheaters, scold

[11] We did, however, include the distributive scopal alternative in which "someone" is the subject" of an "everyone" object. The two readings of this can either be there is some person *X* who [verbs] everyone, or it can be the distributive reading where for every person *X*, they are [verbed] by someone (not necessarily the same someone). The inclusion of a scopal ambiguity was due to an oversight on our part; however, because there were proportionally fewer of these pairings and because these pairings occurred in both conditions, the possible scopal ambiguity should not have an impact on our results.

the liars, teach the trainees, train the recruits, serve the queen, admire the hero

We utilized the following suite of diagnostics to select our condition matrix verbs:

1. SR predicates can be replaced by an expletive *it*; SCs cannot. (Polinsky, 2013; Landau, 2024)
   - Base: John seems/wants to annoy his brother.
   - SR: It seems John annoys his brother.
   - SC: *It wants John annoys his brother.
2. SR predicates can be replaced by an expletive *there*; SCs cannot. (Polinsky, 2013; Landau, 2024)
   - Base: A mouse seemed/wanted to be stuck in the house.
   - SR: There seemed to be a mouse stuck in the house.
   - SC: *There wanted to be a mouse stuck in the house.
3. SR predicates allow for idioms to retain their idiomatic meanings; SCs can only retrieve the literal meaning. (Polinsky, 2013; Landau, 2024)
   - Idiom: Every time my friend pet-sits, my fish *go belly up*. *(meaning: my fish die)*
   - SR: My fish seem to go belly up every time my friend pet-sits. *(Die meaning: still easily accessible)*
   - SC: My fish want to go belly up every time my friend pet-sits. *(Die meaning: less accessible if at all)*
4. When SR sentences are passivized, the meaning is equivalent. Passivization of the SC yields asymmetric meanings. (Sánchez et al., 2016)
   - SR: The teachers seemed to select the volunteers. = The volunteers seemed to be selected by the teachers.
   - SC: The teachers wanted to select the volunteers. $\neq$ The volunteers wanted to be selected by the teachers.
5. SRs allow for scope ambiguity, but SCs do not. (Polinsky, 2013; Landau, 2024)
   - SC: Someone from HR seems to win the office raffle every year.
     - *De re* reading: There is someone specific in HR who seems to win the raffle each year.
     - *De dicto* reading: It seems that the winner of the office raffle each year is someone from HR.
   - SR: Someone from HR wants to win the office raffle every year.
     - *De re* reading: There is someone specific in HR who wants to win the raffle each year.
     - *De dicto* reading: inaccessible.
6. Singular subjects of SC predicates can participate in plural-coded verbs,[12] but SRs cannot. (Landau, 2024). By plural-coded verbs, we mean those that necessitate multiple participants (e.g., it's ungrammatical to say "I met at midnight" as "meeting" requires two or more participants).
   - SR: *The student seemed to meet in the library.
   - SC: The student wanted to meet in the library.

From this, we selected 6 SC verbs—all of which met Landau (2024)'s criteria for logophoric control predicates—and 6 SR verbs, listed in List (8).[13]

(8) **Matrix Verbs**

a. *Subject Control Verbs*: wanted, expected, wished, liked, hated, promised

b. *Subject Raising Verbs*: appeared, seemed, happened, began, continued, tended

B Linear Mixed Effect Models

Below are the linear mixed effect models fit for results reported in Table 1. Random effects were identified via model comparison and included by-MatrixVerb, by-SubjectWord, by-EmbedVerb, by-ObjectWord, and by-LanguageModel random

[12] This is known as "partial control," and is a diagnostic for (Landau, 2024)'s logophoric control predicates.

[13] We acknowledge that three of our raising verbs are contentious: *begin* and *continue*, though they do appear as raising verbs in Postal (1974). There are instances of both appearing in the expletive construction (e.g., "It **continued** that the reserve would be 'a back-up solution only"' and "There **began** to be fewer men who paid taxes," both taken from Davies (2008–)).

slopes for our factor(s) of interest (Condition and LinDist). Word-pairs with direct objects made for more complicated linear models due to the addition of a by-ObjectWord grouping factor for random effects. Because of this, the linear model for `subj-dobj` included random intercepts for all grouping factors mentioned, but only warranted random slopes for the grouping factor of language model and linear distance. The linear model for `subj-dobj` included the same as well as a random slope for the group factor of the direct object noun/pronoun.

"Cond" refers to the Condition (SC vs SR), "CondVerb" refers to the matrix verb (6 in each condition); "Subjword" refers to the word used as the subject; "Objword" refers to the word used as the object; "Embed" refers to the embedded verb; and "Model" refers to the LLM.

- `subj-embed`: PredDist Cond + (1 | CondVerb) + (1 + Cond | SubjWord) + (1 + Cond | Embed) + (1 + Cond | Model)
- `subj-infin`: PredDist Cond + (1 | CondVerb) + (1 + Cond | SubjWord) + (1 + Cond | Embed) + (1 | Model)
- `subj-dobj`: PredDist Cond * LinDist + (1 | CondVerb) + (1 | ObjWord) + (1 | SubjWord) + (1 | Embed) + (1 + Cond + LinDist| Model)
- `embed-dobj`: PredDist Cond * LinDist + (1 | CondVerb) + (1 + Cond | ObjWord) + (1 | SubjWord) + (1 | Embed) + (1 + Cond + LinDist | Model)

C Figures

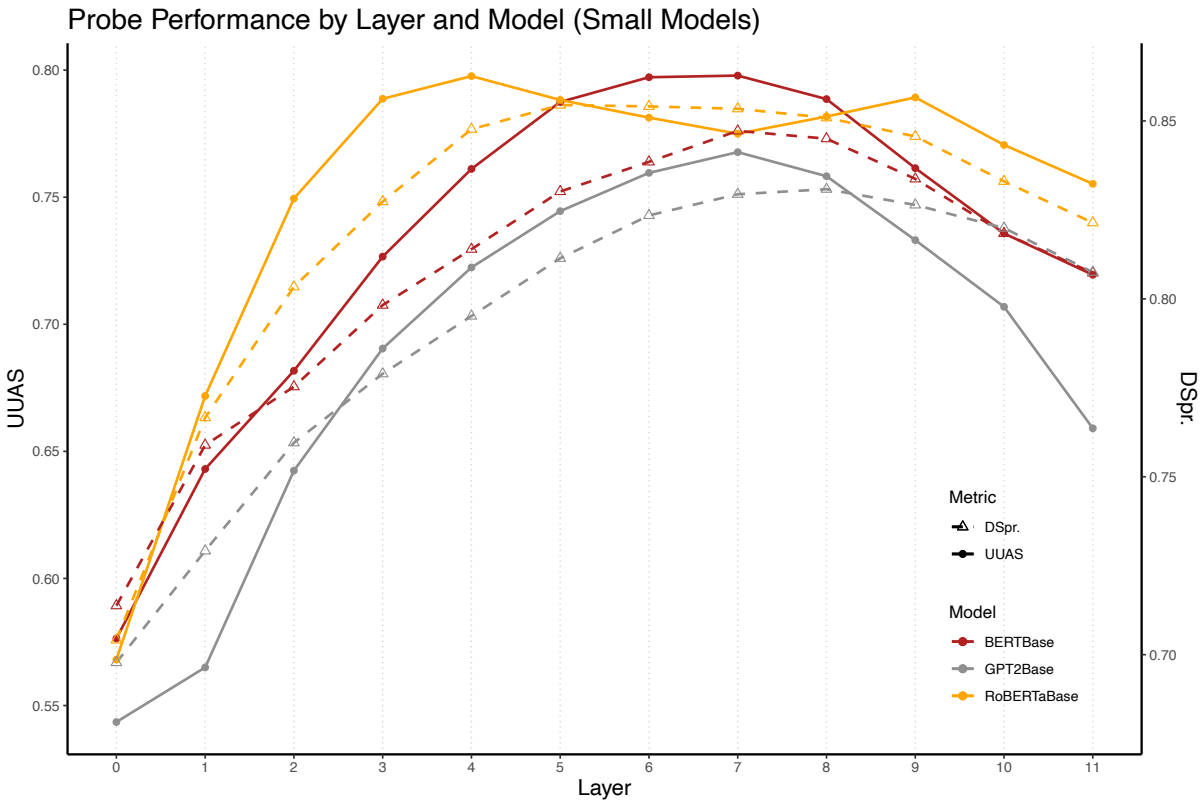

Figure 5: Probe performance for all small models. The solid lines are plotted against the left-hand y-axis and display the performance by Unlabeled Unattached Accuracy Score (UUAS) while the dotted lines plot the average Spearman correlation between the predicted and gold distances (DSpr.) along the right-hand y-axis. Highest-performing probes were `BERT-base-layer7`, `RoBERTa-base-layer4`, and `GPT2-layer7`.

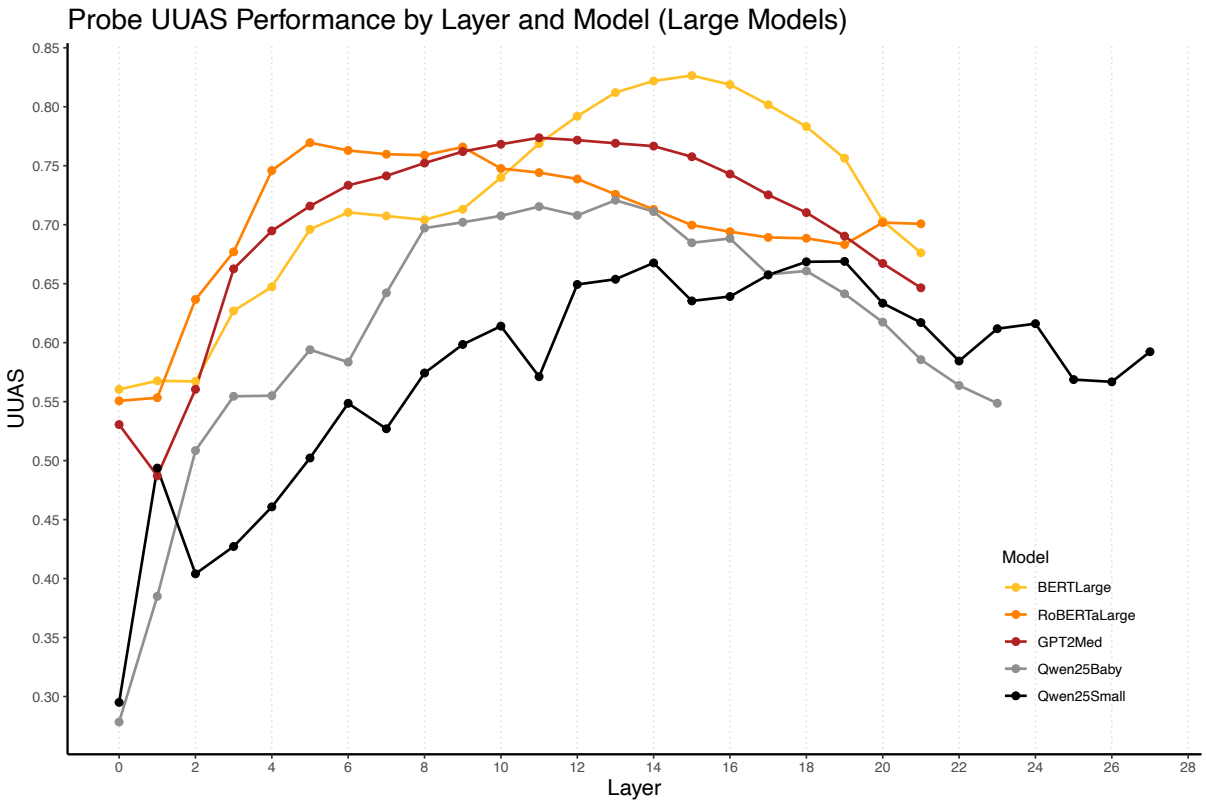

Figure 6: Probe Unlabeled Unattached Accuracy Score (UUAS) performance for all of the larger models. Highest-performing probes were `BERT-large-layer15`, `RoBERTa-large-layer5`, `GPT2-med-layer11`, `Qwen2.5-0.5-layer13` "Qwen25Baby", and `Qwen2.5-1.5-layer19` "Qwen25Small".

Figure 7: Probe average Spearman correlation (DSpr) performance for all of the larger models. Highest-performing probes were `BERT-large-layer15`, `RoBERTa-large-layer5`, `GPT2-med-layer11`, `Qwen2.5-0.5-layer13` "Qwen25Baby", and `Qwen2.5-1.5-layer19` "Qwen25Small".

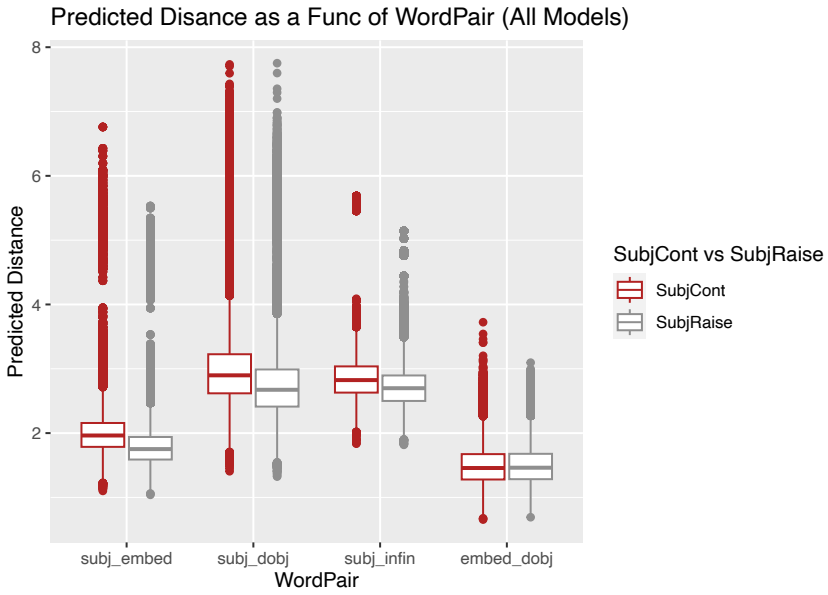

Figure 8: Predicted distances by WordPair for all LLMs. While the SC condition yields longer predicted distances than the SR condition, the baseline of `embed-dobj` shows no difference in the probes' predicted distance for the two conditions.

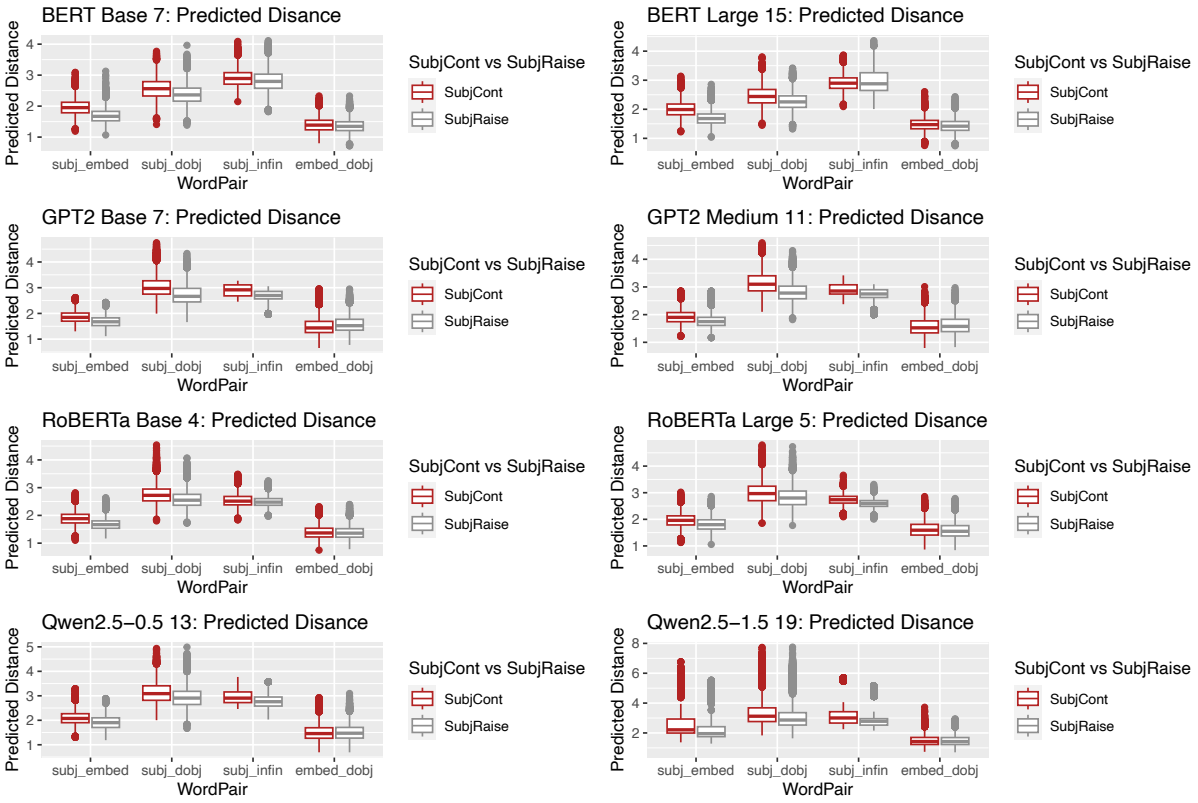

Figure 9: Predicted distances by WordPair and by LLM.

					Fixed Effects					
Model	**WordPair**	**Condition**	**Acc**	**PredDist** *(avg)*	Coefficient	$\hat{\beta}$	SE($\hat{\beta}$)	df	t	p
BB7	**Subj-Embed**	Cont	94.95%	1.96	(Intercept)	1.82245	0.04081	19.77873	44.655	<2e-16
		Raise	85.34%	1.69	Condition	-0.28385	0.04832	12.41598	-5.874	**6.58e-05**
	Subj-Infin	Cont	89.53%	2.56	(Intercept)	2.83493	0.08243	13.12998	34.39	2.97e-14
		Raise	81.48%	2.38	Condition	-0.11851	0.10040	12.28409	-1.18	0.26
	Subj-Dobj	Cont	68.54%	2.90	(Intercept)	2.77837	0.33444	69.00598	8.307	5.47e-12
		Raise	58.51%	2.81	Condition	-0.58313	0.13924	53.28816	-4.188	**0.000106**
					LinDist	-0.08603	0.06823	66.18457	-1.261	0.211758
					Interaction	0.09568	0.02659	43.99406	3.598	**0.000808**
BL15	**Subj-Embed**	Cont	96.51%	2.00	(Intercept)	1.84524	0.04304	24.16131	42.87	<2e-16
		Raise	90.34%	1.69	Condition	-0.31813	0.06931	13.65580	-4.59	**0.000448**
	Subj-Infin	Cont	95.22%	2.46	(Intercept)	2.92094	0.08590	12.97574	34.002	4.54e-14
		Raise	90.16%	2.28	Condition	0.02512	0.16398	11.02102	0.153	0.881
	Subj-Dobj	Cont	55.15%	2.91	(Intercept)	2.48195	0.26788	69.34548	9.265	9.42e-14
		Raise	58.58%	2.96	Condition	-0.38234	0.12774	39.42419	-2.993	**0.00475**
					LinDist	-0.03076	0.05465	66.48586	-0.563	0.57547
					Interaction	0.05318	0.02370	29.59535	2.243	**0.03251**
RB4	**Subj-Embed**	Cont	96.42%	1.89	(Intercept)	0.03251	0.02186	45.07316	81.24	<2e-16
		Raise	91.27%	1.68	Condition	-0.20958	0.01251	13.41216	-16.75	**2.23e-10**
	Subj-Infin	Cont	96.31%	2.54	(Intercept)	2.51492	0.04079	19.76975	61.652	<2e-16
		Raise	91.18%	2.49	Condition	-0.04289	0.06567	10.36167	-0.653	0.528
	Subj-Dobj	Cont	96.21%	2.75	(Intercept)	2.44086	0.46053	67.63892	5.300	1.37e-06
		Raise	90.68%	2.58	Condition	-0.55603	0.10135	53.91938	-5.486	**1.12e-06**
					LinDist	0.05054	0.09478	67.00031	0.533	0.596
					Interaction	0.09185	0.01952	45.36245	4.705	**2.41e-05**
RL5	**Subj-Embed**	Cont	98.61%	1.97	(Intercept)	1.88802	0.03756	71.54606	50.261	<2e-16
		Raise	96.61%	1.81	Condition	-0.15973	0.03398	11.28757	-4.701	**0.000605**
	Subj-Infin	Cont	96.91%	2.74	(Intercept)	0.000605	0.02767	24.94310	96.339	<2e-16
		Raise	93.80%	2.60	Condition	-0.14062	0.03783	10.90377	-3.717	**0.00345**
	Subj-Dobj	Cont	95.31%	2.99	(Intercept)	1.84384	0.66272	69.63534	2.782	0.00694
		Raise	91.96%	2.83	Condition	-0.62076	0.10333	52.19872	-6.008	**1.85e-07**
					LinDist	0.25798	0.13658	69.36638	1.889	*0.06308*
					Interaction	0.10951	0.02027	45.48663	5.402	**2.32e-06**
GS7	**Subj-Embed**	Cont	99.52%	1.86	(Intercept)	1.77275	0.03519	32.62122	50.379	<2e-16
		Raise	98.88%	1.69	Condition	-0.17882	0.05412	13.69496	-3.304	**0.00536**
	Subj-Infin	Cont	99.22%	2.90	(Intercept)	2.76690	0.06601	11.33192	41.91	8.7e-14
		Raise	87.23%	2.67	Condition	-0.26820	0.13411	11.95000	-2.00	*0.0688*
	Subj-Dobj	Cont	98.85%	3.04	(Intercept)	3.02957	0.44599	69.57371	6.793	3.05e-09
		Raise	96.32%	2.73	Condition	-0.66348	0.14587	54.34750	-4.548	**3.07e-05**
					LinDist	-0.03567	0.09176	68.84865	-0.389	0.69867
					Interaction	0.08601	0.02911	48.70634	2.954	**0.00481**
GM11	**Subj-Embed**	Cont	99.67%	1.92	(Intercept)	1.84576	0.05312	19.30353	34.745	<2e-16
		Raise	98.18%	1.77	Condition	-0.16218	0.08630	13.61600	-1.879	*0.0818*
	Subj-Infin	Cont	98.74%	2.89	(Intercept)	2.77503	0.07324	14.28224	37.891	9.76e-16
		Raise	93.16%	2.69	Condition	-0.18272	0.13783	12.19611	-1.326	0.209
	Subj-Dobj	Cont	97.49%	3.15	(Intercept)	2.93307	0.05039	54.90142	58.210	<2e-16
		Raise	90.68%	2.83	Condition	-0.33108	0.07295	17.11412	4.538	**0.000286**
QB13	**Subj-Embed**	Cont	95.14%	2.10	(Intercept)	2.01250	0.06700	11.27262	30.039	4.08e-12
		Raise	99.32%	1.91	Condition	-0.18818	0.06758	12.38379	-2.785	**0.0161**
	Subj-Infin	Cont	72.19%	2.95	(Intercept)	2.8575	0.1087	12.3465	26.276	3.27e-12
		Raise	72.32%	2.78	Condition	-0.2119	0.1363	11.6518	-1.555	0.147
	Subj-Dobj	Cont	92.73%	3.12	(Intercept)	2.39605	0.40124	68.76982	5.972	9.24e-08
		Raise	95.83%	2.93	Condition	-0.19786	0.06810	12.92618	-2.905	**0.0123**
					LinDist	0.15452	0.08148	65.01205	1.896	*0.0624*
QS19	**Subj-Embed**	Cont	87.29%	2.66	(Intercept)	2.55208	0.43185	6.06759	5.910	0.00100
		Raise	94.86%	2.30	Condition	-0.31453	0.08549	14.93464	-3.679	**0.00225**
	Subj-Infin	Cont	67.86%	3.23	(Intercept)	3.2211	0.3615	6.1781	8.911	9.44e-05
		Raise	68.65%	2.95	Condition	-0.2839	0.1270	14.2346	-2.236	**0.0419**
	Subj-Dobj	Cont	86.06%	3.50	(Intercept)	3.98534	0.53651	13.82113	7.428	3.47e-06
		Raise	93.00%	3.18	Condition	-0.27527	0.05825	11.01648	-4.726	**0.000621**
					LinDist	-0.14038	0.06518	66.57211	-2.154	**0.034893**

Table 2: From top to bottom, models are: `BERT-base-layer7`, `BERT-large-layer15`, `RoBERT-base-layer4`, `RoBERT-large-layer5`, `GPT2-small-layer7`, `GPT2-medium-layer11`, `Qwen2.5-0.5-layer13`, and `Qwen2.5-1.5-layer19`. Model comparison using anova revealed LinDist did not significantly improve the linear mixed-effects model for GPT2-medium; this method also showed the interaction between the Condition (SC vs SR) and LinDist was not a main effect for the **Qwen2.5** models.