Crossed with Entailments:
What do Language Models Infer from Crossing Dependencies?

Cove Geary
For Ling. 380 / Spring 2020 / with Bob Frank
Yale University / New Haven, CT
`cove.geary@yale.edu`

Abstract

What, if any, syntactic structure can a neural network language model "induce" by being trained on language? This study seeks to offer another perspective on this question by studying *crossing dependencies* via language modeling and entailment. By framing the entailment as a simple language modeling task, the model's representation of crossing dependencies may be tested alongside its ability to leverage that understanding. Although the language model tested (GPT-2) seems primed to crossing dependencies in certain cases, its ability to make inferences about crossing dependencies seems heuristic. Tentatively, it seems that GPT-2 may rely on crossing dependencies only when coerced by phrase structures which are particularly antagonistic to non-hierarchical interpretations.

All code and data for this experiment may be found in the following GitHub repository: https://github.com/covertg/crossing-nn.

1 Introduction

In order to process complex phrases of natural language, language models must learn to navigate language's hierarchical structure. Despite vast breakthroughs in various statistical language models, the degree to which a language model truly induces the hierarchical structure of language—which we will refer to as *syntax*—remains the subject of much study. A particular difficulty lies in the interpretation of so-called "black box" neural network models.

As Wilcox et al. (2018) describe, one fruitful line of study for this question is to present neural network language models with "specially controlled sentences designed to draw out behavior that indicates representation of a syntactic dependency" and measure the model's performance. Many recent studies have followed in this line to characterize recurrent neural networks' understanding of syntax, including but not limited to Linzen et al. (2016), Linzen and Leonard (2018), Wilcox et al. (2018), Gulordava et al. (2018), with generally mixed but positive results.

But the field of computational linguistics is far from consensus. Kodner and Gupta (2020) critique a number of findings based in "behavioral probes" with template-based data for jumping to conclusions about high-level syntax too quickly. Comparing the state-of-the-art LSTMs to syntax-naive baselines, they find that the baselines adequately solve a task intended to mete out gradient recognition of different phrasal structures (as in Prasad et al. (2019)). They argue that, given current evidence, LSTMs seem unlikely to truly induce syntactic representations, instead hypothesising word-order and lexical similarity effects. Alongside advocating for nonsyntactic baselines, the authors suggest "consequence-based" behavioral analyses of language models—testing with downstream tasks which should require syntactic understanding to complete, and using naturalistic rather than template-based language data. Relatedly, by probing LSTMs' ability to capture the morphosyntactic properties of filler-gap constraints, Chaves (2020) suggest that extant work on LSTMs' understanding of filler-gap dependencies (Wilcox et al., 2018) may rely on surface-level similarities in the experimental data rather than induction of higher-level structure.

Beyond recurrent models, a new and exceptional challenger has emerged with the development of *Transformer* architectures, which can capture long-distance dependencies purely via attention mechanisms (Vaswani et al., 2017). A new subfield dubbed "BERTology," named after the popular BERT model, seeks to describe how these models work, but many questions remain unanswered (Rogers et al., 2020). Initial evidence suggests that

BERT understands subject-verb agreement in many cases (Goldberg, 2019), while performance with structurally unbounded filler-gap dependencies is decidedly mixed: Transformer-XL and XLNet both fail, while BERT and GPT-2 each seem to represent only certain types of filler-gap relations (Da Costa and Chaves, 2020).

In this paper I develop a method for querying a language model's mastery of syntax by probing its ability to make **entailment**-like inferences. By presenting a model with a premise sentence, we can query its inferences about what that sentence may entail via a simple downstream language modeling task. Then by varying the phrasal structures embedded in the premise and assessing the correctness of its inferences, we may query the extent to which the model's inferences leverage syntactic understanding.

For the variation of of phrase structures, I look to subject-verb-object **crossing dependencies**. These dependencies are a core part of language understanding, tantamount to understanding the subjects and objects of any premise sentence, but the English language allows for a variety of structures where the argument of a verb may ”cross over” other arguments. As a result, a certain amount of syntactic parsing is required to understand sentences with crossing dependencies. Furthermore, it is known that young children struggle to understand phrases with crossing dependencies, and the capability is not acquired until later in development (Friedmann and Costa, 2010), further motivating the case to study crossing dependencies in language models. In sum, this task is no walk in the park for language-learners; to succeed, understanding of syntax is required.

Finally, due to Transformer models' noted contributions to the state-of-the-art on semantic understanding tasks, including the GLUE and SuperGLUE benchmarks (Wang et al., 2019), they make for an interesting test case with this method. In the experiments to follow, I take GPT-2 (Radford et al., 2019) as a test case on seven different forms of phrases. While this paper focuses only on GPT-2 due to time constraints, this method may apply to any language model that assigns conditional probability to a word.

1.1 Research Questions

1. Can we probe a language model's induction of syntax by characterizing its ability to make inferences?
2. Specifically: can GPT-2 correctly infer subject-verb relations given various forms of crossing dependencies? Do its successes or failures arise in syntactically meaningful ways?

2 Method

2.1 Experiment Design

Following the cue of (Friedmann and Costa, 2010), I studied the five phrase types that they did alongside two more. Table 1 lists all phrase forms and an example for each; I added forms (6) and (7), the relative clauses in NP, as further candidates for particularly challenging phrases. Each sentence given by each phrase type, then, represents a *premise* involving two nouns and one or two verbs. Simple entailments are then created from those premises, with a ”correct” entailment being a hypothesis that is in fact entailed by the premise, and an ”incorrect” entailment being any hypothesis which contradicts or is not implied by the premise. Then to join premise-entailment pairs into ”inferences,” I apply **connectives**, either indicating summary or repetition, to form a single sentence/phrase.

Taking (4) *Here is the lion that the tiger chased*, as a running example, then the correct entailment is *The tiger chased the lion*, while the incorrect entailment is *The lion chased the tiger*. Thus taking *again* as a connective, we have:

1. Here is the lion that the tiger chased; again, the tiger chased the lion.
2. *Here is the lion that the tiger chased; again, the lion chased the tiger.

So for these forms (1-7), varying the possible subject/verb combinations of each premise yields two or four pairs of entailments; appending these to the premises via punctuation and a connective then yields correct or incorrect inferences; and a language model is then assessed on its judgment of each inference.

2.2 Assessment

To assess a general-purpose language model's judgment of each inference, I measure **total surprisal**, summed over the entailment region. If the model truly understands the premise and the connective, then it should be significantly more surprised at the incorrect inferences than the correct inferences; if so, we can say that the model makes correct inferences.

#	*Desc.*	*Ex.*
1	Coordination (nX)	The lion growled and the tiger tripped.
2	Coordination (X)	The lion chased the tiger and growled.
3	Coordination (nX)	Near the lion, the tiger growled and tripped.
4	Object Relative in VP (X)	Here is the lion that the tiger chased.
5	Subject Relative in VP (nX)	Here is the lion that chased the tiger.
6	Subject Relative in NP (X)	The lion that chased the tiger growled.
7	Object Relative in NP (X)	The lion that the tiger chased growled.

Table 1: All phrase forms tested. ”X” indicates subject-verb crossing dependency, ”n” signals negation.

Thus, as this task involves contextual inference, we may consider it to be more ”behavioral” and downstream than simpler probes—beyond requiring just an understanding of how a sentence (the premise) should look, it requires the model to understand what that sentence actually means. Yet it still leverages the model's unsupervised knowledge by shoehorning the matter into a language modeling task, rather than a specialized task that the network is fine-tuned to do. In this sense, the task may be difficult, but it is particularly compelling as a high bar to demonstrate syntactic understanding. In related work, it has been shown that surprisal itself is an effective measure to solve Winograd schemas, a reasoning task (Trinh and Le, 2018).

To properly compare total surprisal between cases, it is important to note that the length of the entailments should be the same, as total surprisal is not normalized for sequence length.[1]

2.3 Data

As described, each premise contains two subjects and one or two verbs, and each premise admits either two or four pairs of possible entailments, with items of each pair sharing the same verb but different choices of subjects and objects.

Noting this, I applied a templating process to generate data: given a 5-tuple *(A, B, V_{tr}, V_{in1}, V_{in2})* where A and B are subjects and V's are verbs (transitive or intransitive), each template maps these items to a premise and its possible entailments—with possible entailments either employing a verb which *did* require crossing dependencies in the premise, or a verb which *did not* require crossing dependencies. Then given these premise-entailment pairs, connectives and other phrase-specific lexical items (prepositions, expletives/demonstratives) were filled-in from a small set of possibilities for each, finally yielding full sentences which we may consider to be ”inferences.” Each inference is labeled as *correct* or *incorrect* depending on its entailment, and as *crossing* or *not-crossing* depending on the verb that its entailment uses. Appendix A Table 2 contains examples of the template for each phrase form.

Crucially, the templating process creates **all combinations** of subject/verb pairings, with respective objects. In other words, continuing with the running example with (4), the same 5-tuple generated the following sentences as well.(Forms which involve two verbs, then, yielded even more ”inference” sentences, as the number of combinations is greater.)

3. Here is the tiger that the lion chased; again, the tiger chased the lion.
4. *Here is the tiger that the lion chased; again, the lion chased the tiger.

This controls for the possible effect whereby some subject-verb-object pairings may be inherently more surprising than others. For example, perhaps the model believes it less surprising that lions chase tigers than the other way around. This would be a confound if we evaluated only sentences 1-2 or only sentences 3-4, but by evaluating them all, we neutralize the effect.[2]

Additionally, to deter possible number cues in subject-verb agreement, the simple past tense is used for all verbs. To maintain symmetrical phrase lengths, all subjects were a simple noun phrase consisting of the determiner *the* and a noun; and incidentally, most subjects were singular,

[1] This is the primary reason I ended up not studying entailments which involved negation. But so long as the language model understands negation (a tricky matter itself), it would be useful to use negation to create definitely-contradictory entailments; so future work could look into normalized or relative surprisal measures.

[2] Another benefit of the templating process is that it will facilitate the generation of arbitrarily long sentences, in the case of forms with long-distance dependencies. Again due to the time, this investigation is left for another paper.

Lastly, to avoid related surprisal effects, I constrained the 5-tuples to only those where every possibility that could be generated from them were semantically plausible; that is, the simple sentences A V_{tr} B, B V_{tr} A, $\{A, B\}$ $\{V_{in1}, V_{in2}\}$, are all plausible English sentences. This made for a word puzzle fun for the whole family.

So: with $n = 25$ handcrafted 5-tuples, expanding on all possible combinations for all 7 forms yielded a total of 86,400 sentences.

3 Results

Figure 1 displays the primary findings on the model's *inference* capabilities. If GPT-2's inference capabilities depend on syntactic knowledge, then we should see a syntactically-meaningful trend across phrase types. On first glance, there are some significant findings: (2), coordination with crossing definitely yields high performance; while (3-5), long-distance coordination without crossing and the VP relatives, definitely yield low performance. These do *not*, however, pattern well with phrasal crossing dependencies. Indeed, (1) can be seen as a control case—simple coordination, no crossing—and the model seems to perform at chance.

Appendix A Figure 5 offers a closer look into summarizing the model's total surprisal for each phrase form with respect to inference. However, a deeper dive is necessary to explore the interaction with crossing dependencies: while some phrase types (1-3, 4, 7) yield entailments which are either only-crossing or only-not-crossing,[3] other phrase types (2-4) afford both crossing and not-crossing entailments. Thus before returning to the inference results, I explore the interaction with crossing dependencies, as well as possible interaction with other factors.

3.1 Crossing Dependencies

For the two phrase types that afford both a crossing and a non-crossing entailment, a surprising pattern emerges: inferences which involve crossing yielded *lower* total surprisal than inferences which involve non-crossing verbs. This is evidenced by the vivid cluster of not-crossing datapoints sitting below the crossing ones, in both experiments (Figure 2(a)).

Is it possible that, in these cases, the network becomes *more confident* upon seeing a crossing dependency than it does otherwise? One alternative is that (2) and (6) both share the same entailment structure: all crossing entailments involve the intransitive verb, while all non-crossing entailments involve the transitive verb. Then, could this trend be explained by transitivity?

Perhaps, and it is a limitation of this two-noun dataset that it does not include all the possible crossing phrase structures that could arise with transitive verbs. (The object relatives, (4), and (7), are our holdouts.) But this cannot be the only explanation: looking to (4), the natural comparison structure is (5), as these are the two structures with relative clauses in the VP. Their only difference is in the placement of the *that* complementizer, but as a result, while the object relative (4) yields crossing transitive entailments, the subject relative (5) yields non-crossing transitive entailments.

A between-structure comparison reveals the same trend as the within-structure comparison: average surprisal is lower for crossing dependencies (Figure 2(b)). Therefore, the network's higher confidence upon seeing a predicate which has required a crossing dependency cannot be fully explained by verb transitivity.

It should be noted with a grain of salt that this is not an overwhelming trend across all phrase structures studied. Averaging across all experiments and both types inferences, the model actually produces higher average surprisal for not-crossing than for crossing (Appendix A Figure 4). However, given the asymmetrical differences between these 7 phrase structures, it is not clear whether this finding should be meaningful at all. (For example, a simple ablation study showed that (3) is the main culprit in decreasing the average surprisal for not-crossing. Appendix A Figure 6 gives a full summary of the crossing interaction for each phrase type.)

3.2 Connectives, Other Factors

As mentioned in Section 2.3, the experiment involved an arbitrary choice of connectives, as well as introductory prepositions for (3) an expletive or demonstrative to begin (4, 5). If the language model correctly understands the role that these lexical items play in their respective sentences, then there should be no systematic effect differences in inference evaluation nor overall surprisal for each of these choices. If it fails to understand certain choices, or understands different choices differently, then we would see systematic differences. Boxplots that compare overall model inference,

[3] That is, relying on a verb which required crosssing dependencies in the premise.

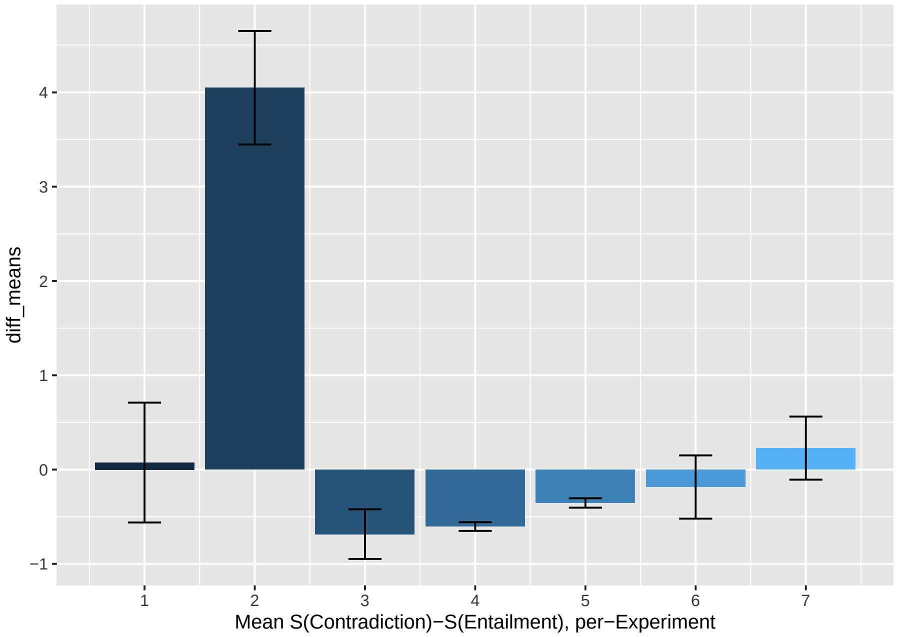


Figure 1: Difference in means of total surprisal for incorrect minus correct inference. Error bars are 95% confidence intervals based on paired t-tests.

with unique datapoints overlaid for each choice of a factor, can reveal both whether that factor has an overall effect on surprisal and whether that factor may have a confounding effect on inference decisions.

Looking to connectives (Fig. 3(a)), there is no noticeable interaction in either case: the datapoints are generally homogeneously mixed in about equal manner across both inference decisions; and while it seems possible that there might be some weak bands forming (e.g. pink in the higher outlier zone), again these appear symmetrical across Entailment and Not-Entailment. A more granular view confirms that each connective seems to have no significant effect on inferences, and each connective also yields a very similar surprisal distribution (Appendix A Figure 7).

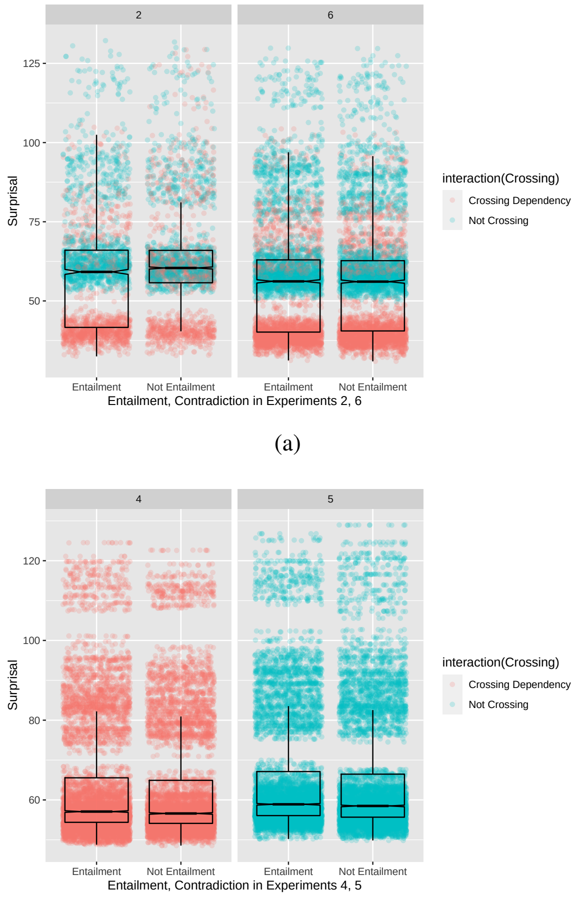
Looking now to prepositions (Fig. 3(b)) and connective/expletive (Fig. 3(c)), the same pattern emerges: there seems to be no significant effect, given the choices made for data generation. This suggests that these choices, despite their literal differences, achieve the same desired effects as sentence constituents in the data; the network has learned their general function as connectives, prepositions, expletives, and demonstratives. Therefore, any effects regarding inference and entailment cannot be explained by these factors.

This is relevant to one noteworthy trend made visible in all the boxplot charts: the jittered datapoints all reveal the presence of systematic gaps in total surprisal, within all phrase types and inferences (see A5). Each phrase type seems to have two or three clusters in the surprisal axis. But by the analysis above, arbitrary word choices made in all sentences play no role in the phenomenon. Further, given that these are shared across phrase types and correct/incorrect entailments, these clusters are likely related to the 5-tuples themselves and whatever *semantic* information the model encodes about them.

3.3 Entailment Inferences

Returning to the model's assessment of correct and incorrect inferences: could crossing dependencies still explain the model's performance? The model seems agnostic to the premises' correct entailments in (1), the easy control, which suggests that the network could instead be relying on contextual cues or heuristics other than true subject-verb dependencies—yet it also seems to exhibit lower surprisal at crossing dependency cues for at least some phrase types, including (2), which it performs significantly well on ($p <$ 2.2e-16, one-sided). The model also seems to have some relative success at performing the inference task on (7), though only approaching significance ($p = 0.09132$).

Further, in (3-5) the model seems to perform below chance—something is cuing it incorrectly. In these cases, whether or not the network correctly



(a)

(b)

Figure 2: Crossing dependency comparisons within- and between- experiments. Blue is crossing, coral is cot-crossing.

represents the crossing dependencies, it does not leverage them to make inferences. This behavior also diverges from the language acquisition literature, wherein subject relatives (5) are acquired earlier.

Searching for structural similarities in the models' successes, we have: (2) yields crossing and not-crossing entailments, all transitive, while (7) yields only crossing entailments, both transitive and intransitive. Thus they do not seem to share many unique structural similarities, and transitivity does not seem to be at play.

However, these phrase structures *are* the ones where shorter n-grams over the dependency-crossed region would be definitively ungrammatical outside of the crossing dependency context. For example, considering the 4-grams at the end of (2) *The lion chased the tiger and growled* and (7) *The lion that the tiger chased growled*, we get ”the tiger and growled” and ”the tiger chased growled,” which make no sense, and are in fact unsalvageable without the broader context of a crossing dependency. Meanwhile, the other phrase structures are immune to this nonsensical windowing *except* for the case of (6). Considering (6) *The lion that chased the tiger growled*, although the 4-gram is ungrammatical, the 3-gram is perfectly fine.

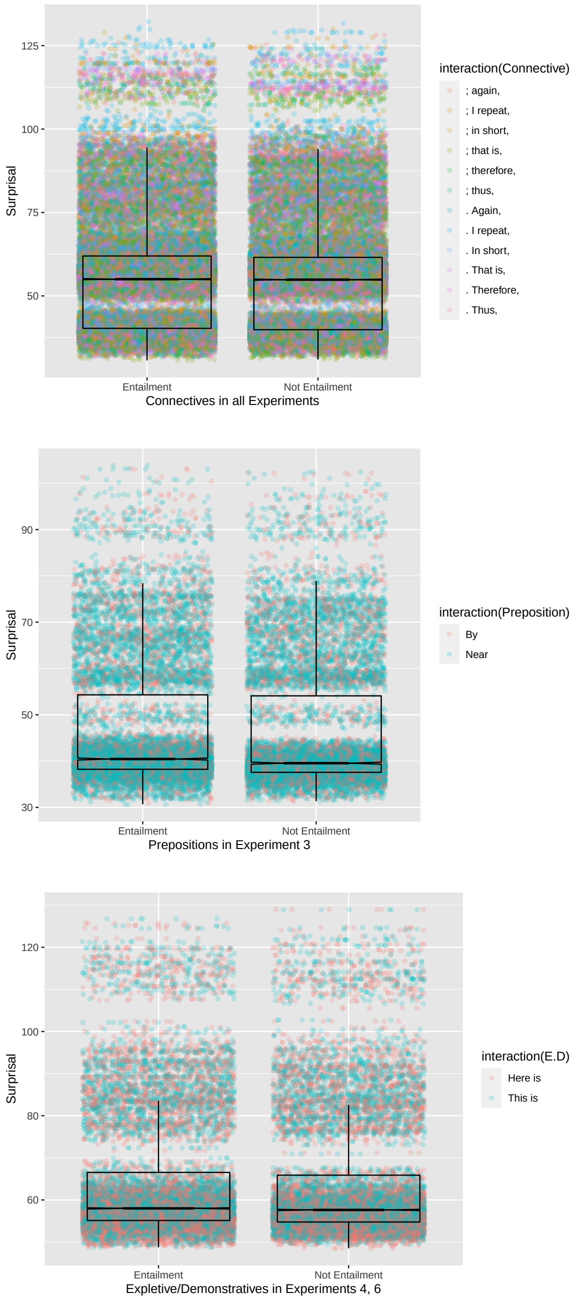


Figure 3: Connective (1-7)), Preposition (3), and Expletive/Demonstrative (4,5) effect on inferences and surprisal. Different color dots represent different choices.

Perhaps, then, these smaller-window contexts such as “the tiger and growled” and “the tiger chased growled” serve to disallow more linear interpretations, forcing the network to consider other alternatives, and thus better-perform on the entailment task in the face of crossing dependencies. In this analysis, the model may still leverage knowledge about crossing dependencies, but only when coerced in certain ways.

4 Discussion

Overall, these findings point to a limited understanding of syntax, insofar as GPT-2 can leverage syntactical understanding to complete the inference task. A breakdown of the model's response to crossing dependencies reveals that it is sensitive to the syntactical dance of crossing dependencies in at least some cases, but it may not necessarily act upon them correctly. This supports the claim that this new crossing dependency inference task is a difficult one, but it also suggests that the task could be a fruitful one for future work to leverage.

These findings may also corroborate extant knowledge on BERT, whereby although BERT has been shown to encode aspects of syntactic structure, it may not actually rely on that knowledge (Rogers et al., 2020).

Future work could easily compare this GPT-2 baseline to other models, including cutting-edge ones and older (e.g. LSTM, n-gram) ones, to better-understand the capabilities and limits of unsupervised language learning. One particularly interesting model to investigate might be Google's T5, as it is trained for a variety of high-level, downstream tasks via a unified language modeling objective (Raffel et al., 2019). It should also be noted that although surprisal has been the assumed metric throughout this paper, because the crux of each inference is understanding the correct subject for a verb, any model which assigns probability to a single word given some surrounding context could easily be applied to this method—thus BERT's masked language modeling could apply here, as well.

In addition, this study is not without its limitations, and much could be done to refine and expand on the work done here. Given that the model did not understand the simple coordination baseline, it could be fruitful to explore which types of entailments the model does reliably ”infer” with this language modeling task. Although the phrase structures studied here were based in past work, they do not cover the range of even all simple phrase structures which have crossing dependencies; thus broader coverage of phrases could reveal more apparent patterns. Furthermore, this study did not address the unbounded nature of many of the dependencies discussed, as sentence length was kept relatively short and fixed. Expanding the distance between predicates and arguments would be another fruitful way to explore the limits of language models' dependency inferences.

Acknowledgments

Thank you, Professor Bob Frank for your kind assistance and advising, and especially for your patience throughout this process. Thank you, Sophie Hao, for your advising throughout this class. Thank you to my mom, for being my mom, and also for the assistance in generating 5-tuple data :).

I also wish to apologize for including charts which rely on color vision to fully interpret.

References

Rui P Chaves. 2020. What don't rnn language models learn about filler-gap dependencies? *Proceedings of the Society for Computation in Linguistics*, 3(1):20–30.

Jillian K Da Costa and Rui P Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. *Proceedings of the Society for Computation in Linguistics*, 3(1):189–198.

Naama Friedmann and João Costa. 2010. The child heard a coordinated sentence and wondered: On children's difficulty in understanding coordination and relative clauses with crossing dependencies. *Lingua*, 120(6):1502–1515.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *CoRR*, abs/1803.11138.

Jordan Kodner and Nitish Gupta. 2020. Overestimation of syntactic representationin neural language models. *arXiv preprint arXiv:2004.05067*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Tal Linzen and Brian Leonard. 2018. Distinct patterns of syntactic agreement errors in recurrent networks and humans. *CoRR*, abs/1807.06882.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. *arXiv preprint arXiv:1909.10579*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

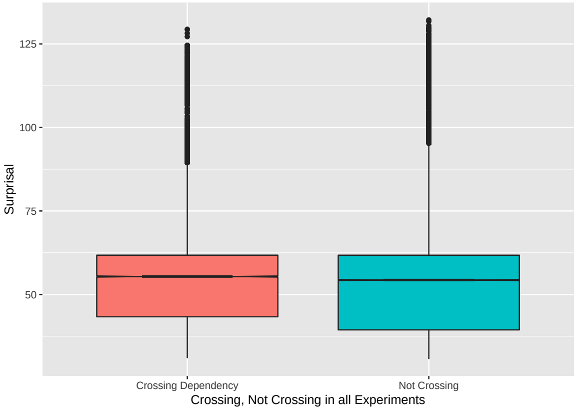


Figure 4: Crossing dependency interaction, averaged across all phrase types and inferences.

#	*Form*	*Premise*	*Correct, X*	*Correct, nX*	*Incorrect, X*	*Incorrect, nX*
1	Coord.	A V_{in1} and B V_{in2}		A V_{in1}; B V_{in2}		A V_{in1}; B V_{in2}
2	Coord.	A V_{tr} B and V_{in1}	A V_{in1}	A V_{tr} B	B V_{in1}	B V_{tr}
3	Coord.	{Prep} A, B V_{in1} and V_{in2}		B V_{in1}; B V_{in2}		A V_{in1}; A V_{in2}
4	O.Rel VP	{E/D} A {Comp} B V_{tr}	B V_{tr} A		A V_{tr} B	
5	S.Rel VP	{E/D} A {Comp} V_{tr} B		A V_{tr} B		B V_{tr} A
6	S.Rel NP	A {Comp} V_{tr} B Vin	A V_{in1}	A V_{tr} B	B V_{tr} A	B V_{in1}
7	O.Rel NP	A {Comp} B V_{tr} Vin	A V_{in1}; B V_{tr} A		B V_{in1}; A V_{tr} B	

Table 2: Sampling of templates made for each form. A and B represent subjects; V's represent transitive and intransitive verbs; X/nX signifies that the verb does/does not employ a crossing dependency in the premise; items in {braces} expanded multiple sentences from the possible options, with the prepositions being *near*, *by*; the expletive/demonstrives beings *here is*, *it is*; the complementizer being only *that*; and the connectives being *I repeat*, *again*, *in short*, *therefore*, *that is*, *thus*, with preceding punctuation being both semicolon and period; and lastly, if a form yields multiple possible entailments of the same type, they are separated by a semicolon above. Note that this is just an exemplary sample of one template line from each form—all possible combinations/orderings of subjects and verbs were generated, both for premises and entailments, for each form.

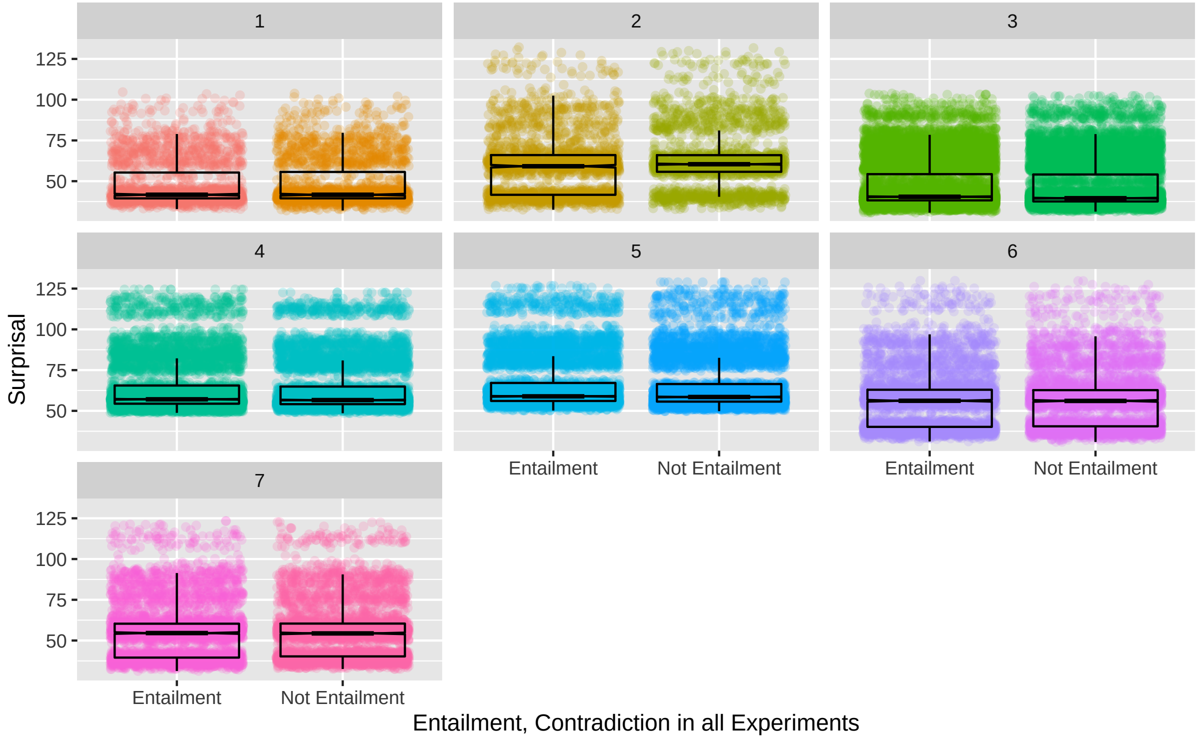


Figure 5: Notched boxplots for correct, incorrect inferences in all phrase types, overlayed with jittered datapoints. There is a high range in total surprisal across all sentences; however, the tiny notches of the boxplots indicate small interquartile range.

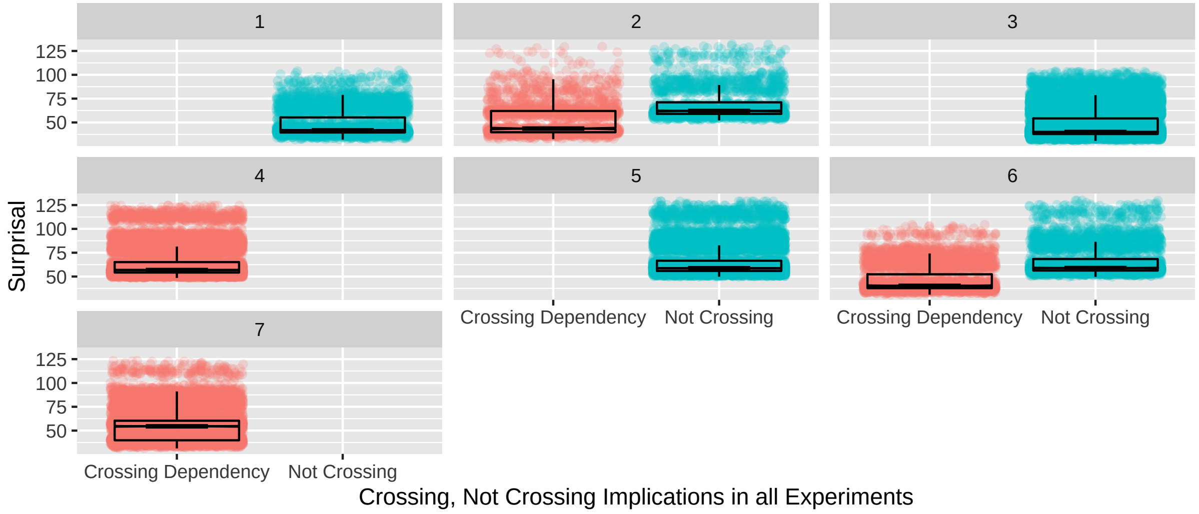


Figure 6: Crossing dependency interaction, marginalized across all phrase types.

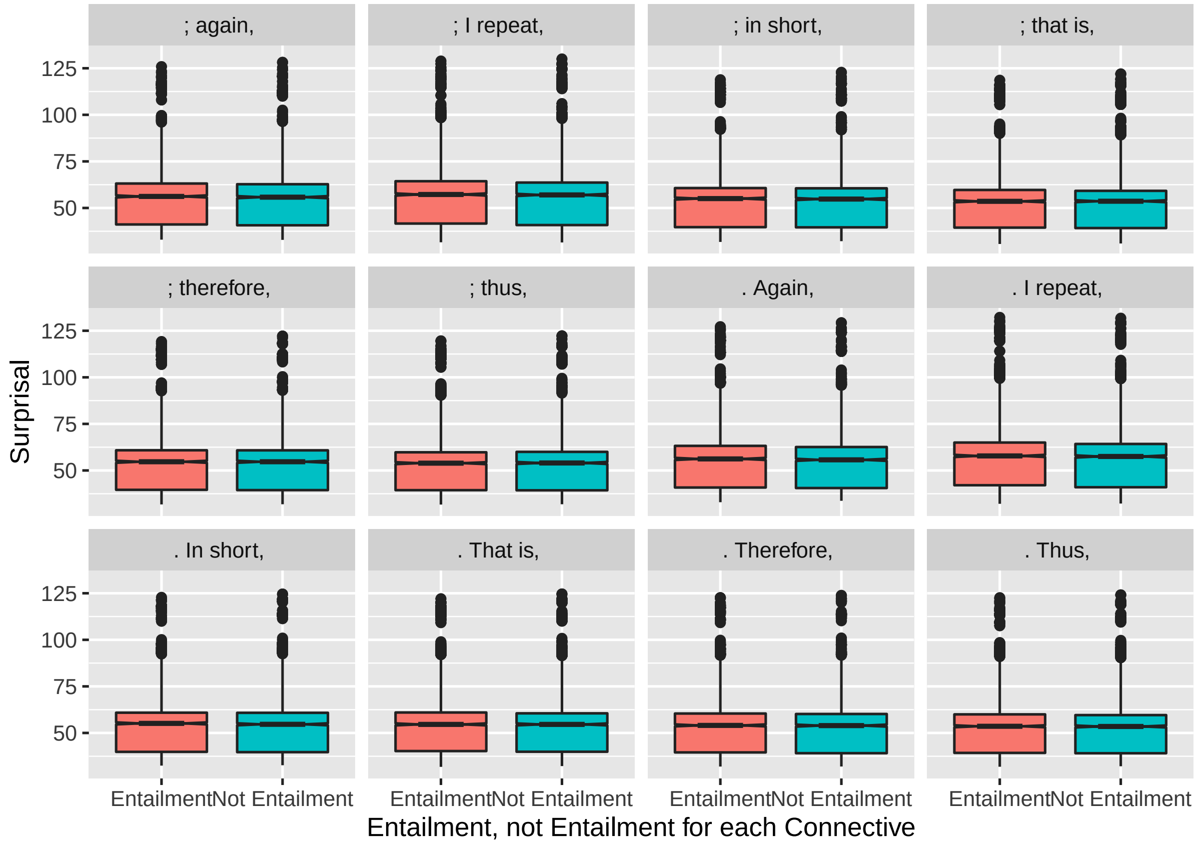


Figure 7: Connective effect on inference across all connectives. There seems to be no effect; and further, each connective seems to yield a very similar surprisal distributions.