How Much Syntactic Supervision is "Good Enough"?

Hiroshi Noji[†*]
Artificial Intelligence Research Center
AIST
hiroshi.noji@aist.go.jp

Yohei Oseki[*]
Graduate School of Arts and Sciences
University of Tokyo
oseki@g.ecc.u-tokyo.ac.jp

Abstract

In this paper, we explore how much syntactic supervision is "good enough" to make language models (LMs) more human-like. Specifically, we propose the new method called *syntactic ablation*, where syntactic LMs, namely Recurrent Neural Network Grammars (RNNGs), are gradually ablated from full syntactic supervision to zero syntactic supervision (≈ unidirectional LSTM) by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof. The 17 ablated grammars are then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. The results of our syntactic ablation demonstrated that (i) the RNNG with zero syntactic supervision underperformed the RNNGs with some syntactic supervision, (ii) the RNNG with full syntactic supervision underperformed the RNNGs with less syntactic supervision, and (iii) the RNNG with mild syntactic supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL. Those results may suggest that the "good enough" approach to language processing seems to make LMs more human-like.

1 Introduction

In the literature on targeted syntactic evaluation (Linzen et al., 2016; Marvin and Linzen, 2018), recurrent neural networks (RNNs) such as LSTMs have been demonstrated to implicitly learn syntactic structures of natural language (e.g., subject-verb agreement), despite the lack of explicit syntactic supervision (cf. Hewitt and Manning, 2019). Moreover, those RNNs also turned out to benefit from explicit syntactic supervision. RNNs integrated with explicit syntactic supervision, namely Recurrent Neural Network Grammars (RNNGs; Dyer et al. 2016), have received considerable attention for their cognitive plausibility and outperformed RNNs in not only targeted syntactic evaluation (Kuncoro et al., 2018; Wilcox et al., 2019) but also psychometric predictive power (Hale et al., 2018; Wilcox et al., 2020; Yoshida et al., 2021).

However, despite the previous debate over the dichotomy between the presence and absence of syntactic supervision, how much syntactic supervision is necessary and sufficient remains to be investigated. Especially, there are two potential reasons to believe that full syntactic supervision is suboptimal. Theoretically, full syntactic supervision may override lexical heuristics implicitly learned with RNNs, where information on terminal symbols vanishes via recursive composition operations (cf. Kuncoro et al., 2017). Empirically, full syntactic supervision seems to destroy the performance of long-distance dependencies, especially (pseudo-)cleft constructions, where both acceptable (e.g., *What he **did** was prepare the meal.*) and unacceptable (e.g., **What he **ate** was prepare the meal.*) sentences share the exactly same syntactic structure (Figure 1) and should be distinguished via lexical heuristics alone (cf. Noji and Oseki, 2021). Therefore, it is reasonable to hypothesize that optimal syntactic supervision lies somewhere between full and zero syntactic supervision in order to balance syntactic structures and lexical heuristics. Intuitively speaking, if we teach too much syntax to language models, those models will forget lexicon.

In this paper, we explore how much syntactic supervision is "good enough" to make language models more human-like. Specifically, we propose the new method called *syntactic ablation*, where RNNGs are gradually ablated from full syntactic supervision to zero syntactic supervision (≈ unidirectional LSTM) by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof. The 17 ablated grammars are then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark (Gauthier et al., 2020). The results demonstrate that the RNNG with mild syntactic

[†]Currently affiliated with LeapMind Inc.: noji@leapmind.io.
[*]Denotes equal contribution.

Findings of the Association for Computational Linguistics: EACL 2023, pages 2300–2305
May 2-6, 2023

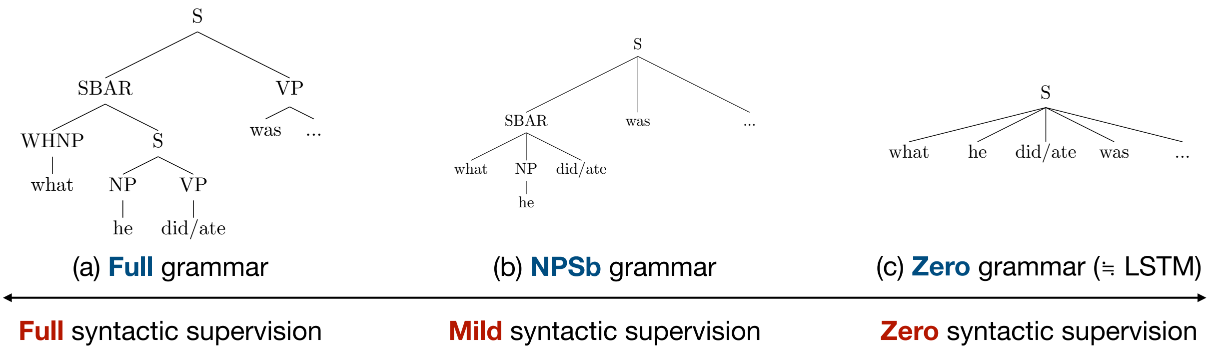

Figure 1: Our proposed method of syntactic ablation. RNNGs are gradually ablated from (a) full syntactic supervision, through (b) mild syntactic supervision, to (c) zero syntactic supervision (≈ unidirectional LSTM) by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof, hence the 17 ablated grammars.

supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL, which are then discussed in the broader context of the computational psycholinguistic literature (Ferreira et al., 2002; Ferreira and Patson, 2007).

2 Methods

2.1 Recurrent Neural Network Grammars

Recurrent Neural Network Grammars (RNNGs; Dyer et al. 2016) are deep generative models of sentences and structures. RNNGs employ the stack LSTM (Dyer et al., 2015) to compute probability distributions over 3 parsing actions below:

- `NT`: Open nonterminal symbols.
- `GEN`: Generate terminal symbols.
- `REDUCE`: Close nonterminal symbols.

For the `REDUCE` action, RNNGs adopt the bidirectional LSTM to encode terminal and nonterminal symbols both left-to-right and right-to-left into phrasal representations. For inference, RNNGs utilize word-synchronous beam search (Stern et al., 2017) implemented in Noji and Oseki (2021).[1]

2.2 Syntactic ablation

Our proposed method of syntactic ablation is summarized in Figure 1. RNNGs are gradually ablated from full syntactic supervision to zero syntactic supervision by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof, hence 17 ablated grammars below:

- `Zero`: Zero grammar.
- `N`: NP nonterminal symbol only.
- `V`: VP nonterminal symbol only.
- `P`: PP nonterminal symbol only.
- `Sb`: SBAR nonterminal symbol only.
- `NV`: NP and VP nonterminal symbols.
- `NP`: NP and PP nonterminal symbols.
- `NSb`: NP and SBAR nonterminal symbols.
- `VP`: VP and PP nonterminal symbols.
- `VSb`: VP and SBAR nonterminal symbols.
- `PSb`: PP and SBAR nonterminal symbols.
- `NVP`: NP, VP, and PP nonterminals.
- `NVSb`: NP, VP, and SBAR nonterminals.
- `NPSb`: NP, PP, and SBAR nonterminals.
- `VPSb`: VP, PP, and SBAR nonterminals.
- `NVPSb`: NP, VP, PP, and SBAR nonterminals.
- `Full`: Full grammar.

RNNGs are trained on the parsed sentences. We created the training data for each grammar, which only provides designated nonterminal symbols. Our original dataset is the same as the XL dataset of Hu et al. (2020), which is about 42M tokens from BLLIP corpus (Charniak et al., 2000) and re-parsed by Berkeley neural parser (Kitaev et al., 2019), from which we only kept the ablated nonterminals to create the dataset. For each grammar, we trained an RNNG with three different random seeds. For the other training settings, we follow Noji and Oseki (2021)'s 100M token experiment.

2.3 Targeted syntactic evaluation

Those ablated grammars were then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark (Gauthier et al., 2020) which includes 6

[1] https://github.com/aistairc/rnng-pytorch

syntactic *circuits*: Agreement, Garden-Path Effects, Licensing, Center Embedding, Gross-Syntactic State, and Long-Distance Dependencies.

We adopted the "perfect match" evaluation metric proposed in Hu et al. (2020), not the "partial match" evaluation metric utilized in the SyntaxGym leaderboard, which seems to overestimate the accuracies of syntactic generalization.

3 Results

3.1 Overall accuracies

Overall accuracies of our syntactic ablation experiments are summarized in Figure 2. Accuracies of SyntaxGym (the vertical axis) are plotted against grammars with different amounts of syntactic supervision (the horizontal axis), together with the accuracies of RNNG and GPT-2-XL reported in Hu et al. (2020). `Zero` (leftmost) and `Full` (rightmost, except RNNG and GPT-2-XL) represent zero and full grammars, respectively, the former of which is equivalent to the unidirectional LSTM.[2] `N`, `V`, `P`, and `Sb` indicate grammars with NP, VP, PP, and SBAR nonterminal symbols preserved, respectively. Therefore, `NP` represents the grammar with NP and PP nonterminal symbols preserved, not to be confused with the grammar with the NP nonterminal symbol preserved.

[2]They are practically equivalent because the `REDUCE` action does not occur except the end of the sentence, where the only difference affecting each word probability is the existence of "(ROOT" symbol at the beginning of the sentence.

There are three key observations here. First, the `Zero` grammar, which is equivalent to the unidirectional LSTM, underperformed the grammars with some syntactic supervision, suggesting that syntactic supervision plays an important role for human-like syntactic generalization. Second, the `Full` grammar also underperformed the grammars with less syntactic supervision and GPT-2-XL in Hu et al. (2020), meaning that full syntactic supervision does not always make LMs human-like. Finally, and most importantly, the `NPSb` grammar achieved the best performance (84.585417) comparable to (or even numerically larger than) the state-of-the-art GPT-2-XL (84.241459).

3.2 Circuit accuracies

Circuit accuracies of our syntactic ablation experiments are summarized in Figure 3. Accuracies of 6 circuits on SyntaxGym (the vertical axis) are plotted against 4 grammars with different amounts of syntactic supervision (the horizontal axis).

Interestingly, the `NPSb` grammar outperformed the `Full` grammar for 5 among 6 syntactic circuits (Agreement, Center Embedding, Garden-Path Effects, Licensing, Long-Distance Dependencies). Notice that the performance advantage of the `NPSb` grammar is significantly larger in Long-Distance Dependencies, especially (pseudo-)cleft constructions, corroborating the hypothesis that optimal syntactic supervision lies somewhere between full and zero syntactic supervision in order to balance syntactic structures and lexical heuristics.

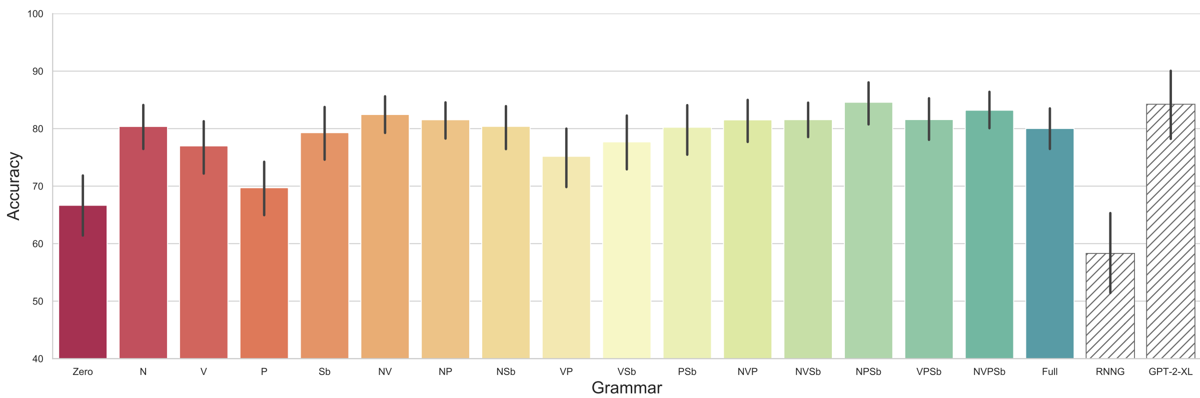

Figure 2: Overall accuracies of our syntactic ablation experiments. Accuracies averaged over 6 circuits on SyntaxGym and random seeds (the vertical axis) are plotted against grammars with different amounts of syntactic supervision (the horizontal axis), together with the accuracies of RNNG and GPT-2-XL reported in Hu et al. (2020). Error bars denote bootstrapped 95% confidence intervals. `Zero` (leftmost) and `Full` (rightmost, besides RNNG and GPT-2-XL) represent zero and full grammars, respectively. `N`, `V`, `P`, and `Sb` indicate the grammars with NP, VP, PP, and SBAR nonterminal symbols preserved, respectively. Therefore, `NP` represents the grammar with NP and PP nonterminal symbols preserved, not to be confused with the grammar with the NP nonterminal symbol preserved.

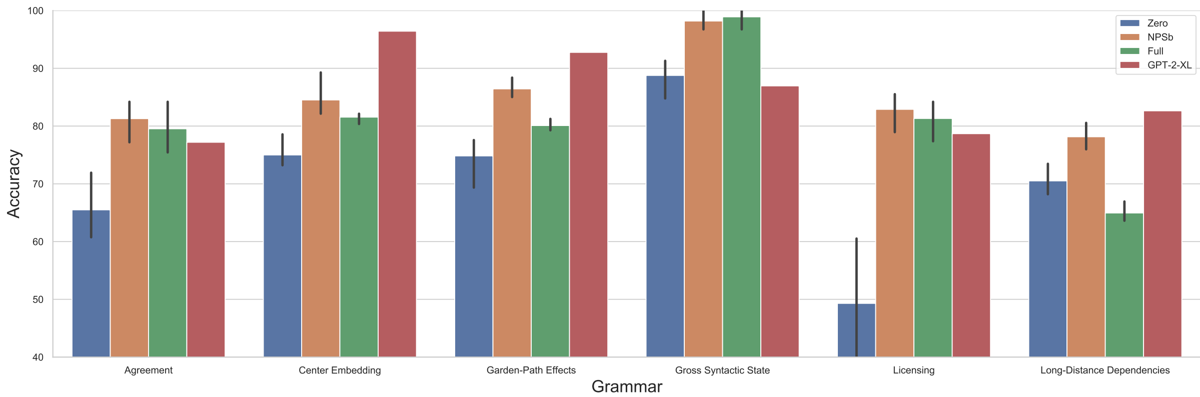

Figure 3: Circuit accuracies of our syntactic ablation experiments. Accuracies of 6 circuits on SyntaxGym (the vertical axis) are plotted against 4 grammars with different amounts of syntactic supervision (the horizontal axis).

4 Discussion

In summary, we performed the syntactic ablation experiments where RNNGs were gradually ablated from full syntactic supervision to zero syntactic supervision (≈ unidirectional LSTM), and then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. In this section, the results of our syntactic ablation experiments will be discussed in the broader context of the computational psycholinguistic literature.

4.1 The "good enough" language processing

The overall accuracies reported in Section 3.1 demonstrated that the RNNG with mild syntactic supervision, especially the `NPSb` grammar, outperformed the RNNGs with zero and full syntactic supervision, as well as GPT-2 XL in Hu et al. (2020). Those results are consistent with the "good enough" approach to language processing (Ferreira et al., 2002; Ferreira and Patson, 2007), where human language processing does not always generate deep syntactic structures, but rather employs shallow syntactic structures and frugal lexical heuristics. Here, we suggest that the RNNG with mild syntactic supervision serves as the mechanistic model of the "good enough" approach to language processing, in that neither deep/hierarchical syntax is necessary nor shallow/flat syntax is sufficient; rather, some syntax in between is "good enough".

4.2 Long-Distance Dependencies

The circuit accuracies reported in Section 3.2 revealed that the `NPSb` grammar outperformed the `Full` grammar for 5 syntactic circuits such as Agreement, Center Embedding, Garden-Path Effects, Licensing, Long-Distance Dependencies. Upon closer inspection (cf. Hu et al., 2020), those 5 syntactic circuits share the isomorphic syntactic structure with long-distance dependencies between dependents inside and outside "heavy" subjects (where the *dependents* are italicized):[3]

- **Agreement**: [$_{\text{NP}}$ The *farmer* [$_{\text{PP}}$ near the clerks]] *knows* many people.
- **Center Embedding**: [$_{\text{NP}}$ The *painting* [$_{\text{SBAR}}$ that the artist painted]] *deteriorated*.
- **Garden-Path Effects**: [$_{\text{NP}}$ The *child* [$_{\text{SBAR}}$ kicked in the chaos]] *found* her way back home.
- **Licensing**: [$_{\text{NP}}$ *No* managers [$_{\text{SBAR}}$ that respected the guard]] have had *any* luck.
- **Long-Distance Dependencies**: [$_{\text{SBAR}}$ What he *did*] was *prepare* the meal.

Importantly, NP, PP, and SBAR representations effectively make linearly distant dependents hierarchically close, while VP representations have no designated raison d'être and, moreover, may override lexical heuristics of verbs (e.g., *knows*, *deteriorated*) via recursive composition operations (cf. Kuncoro et al., 2017; Noji and Oseki, 2021). Thus, at least for those 5 syntactic circuits, the `NPSb` grammar is the optimal syntactic supervision that balances syntactic structures and lexical heuristics.

[3]While those 5 syntactic circuits are not named long-distance dependencies (except the Long-Distance Dependencies circuit which includes filler-gap dependencies and cleft constructions), they all involve long-distance dependencies.

5 Conclusion

In this paper, we explored how much syntactic supervision is "good enough" to make language models more human-like. Specifically, we performed the syntactic ablation experiments where RNNGs were gradually ablated from full syntactic supervision to zero syntactic supervision ($\approx$ unidirectional LSTM), and then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. The results demonstrated that the RNNG with mild syntactic supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL. We hope that the "good enough" approach to language processing (Ferreira et al., 2002; Ferreira and Patson, 2007) provides the promising direction for future research.

Limitations

There are several limitations with this paper. First, the evaluated models are limited; the syntactic ablation was applied to only one model (i.e. RNNG) and remains to be generalized to other models (cf. Sartran et al., 2022). Second, the evaluation datasets are also limited; our ablated RNNGs were evaluated against only one dataset (i.e. SyntaxGym) and remain to be extended to other datasets (cf. Warstadt et al., 2020). In addition, from engineering perspectives, our ablated RNNGs, though lightweight, still require some syntactic supervision, which may induce the scalability bottleneck.

Acknowledgements

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was also supported by JSPS KAKENHI Grant Numbers 20K19877 and 19H04990, and JST PRESTO Grant Number JPMJPR21C2.

References

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. BLLIP 1987-89 WSJ Corpus Release 1 LDC2000T43. Linguistic Data Consortium.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Fernanda Ferreira, Karl G.D. Bailey, and Vittoria Ferraro. 2002. Good-enough representations in language comprehension. *Current Directions in Psychological Science*, 11(1):11–15.

Fernanda Ferreira and Nikole D. Patson. 2007. The 'good enough' approach to language comprehension. *Language and Linguistics Compass*, 1(1-2):71–83.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

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

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.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th*

Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

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.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Hiroshi Noji and Yohei Oseki. 2021. Effective batching for recurrent neural network grammars. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. 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 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. *ArXiv*, abs/2006.01912.

Ryo Yoshida, Hiroshi Noji, and Yohei Oseki. 2021. Modeling human sentence processing with left-corner recurrent neural network grammars. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2964–2973, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.