A Systematic Assessment of Syntactic Generalization in Neural Language Models

Jennifer Hu[1], **Jon Gauthier**[1], **Peng Qian**[1], **Ethan Wilcox**[2], and **Roger P. Levy**[1]
[1]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
[2]Department of Linguistics, Harvard University
{jennhu,pqian,rplevy}@mit.edu
jon@gauthiers.net, wilcoxeg@g.harvard.edu

Abstract

While state-of-the-art neural network models continue to achieve lower perplexity scores on language modeling benchmarks, it remains unknown whether optimizing for broad-coverage predictive performance leads to human-like syntactic knowledge. Furthermore, existing work has not provided a clear picture about the model properties required to produce proper syntactic generalizations. We present a systematic evaluation of the syntactic knowledge of neural language models, testing 20 combinations of model types and data sizes on a set of 34 English-language syntactic test suites. We find substantial differences in syntactic generalization performance by model architecture, with sequential models underperforming other architectures. Factorially manipulating model architecture and training dataset size (1M–40M words), we find that variability in syntactic generalization performance is substantially greater by architecture than by dataset size for the corpora tested in our experiments. Our results also reveal a dissociation between perplexity and syntactic generalization performance.

1 Introduction

A growing body of work advocates that assessment of neural language models should include both information-theoretic metrics, such as perplexity, as well as targeted linguistic evaluation. Benchmarks such as GLUE (Wang et al., 2019a,b) have demonstrated that neural language models trained on naturalistic corpora for next-word prediction learn representations that can yield remarkable performance on many semantic tasks. Targeted syntactic evaluations have shown that these models also implicitly capture many **syntactic generalizations**, ranging from subject–verb agreement to long-distance filler–gap dependencies (Linzen et al., 2016; Marvin and Linzen, 2018; Futrell et al., 2018; Wilcox et al., 2019b). This paper aims to bring targeted evaluations of syntactic performance to scale, complementing similar developments in semantic evaluation (McCoy et al., 2019).

Because the most widespread currency of evaluation for language models is perplexity—how well, on average, a model predicts a word in its context—a primary focus of this paper is the relationship between a model's perplexity and its performance on targeted syntactic evaluations. As perplexity improves, can we expect more human-like syntactic generalization? How do training dataset size and model architecture jointly affect syntactic generalization? And what picture of models' syntactic generalization emerges when evaluation is brought to scale, across dozens of controlled syntactic tests?

In this paper we offer initial answers to these questions, systematically assessing the syntactic generalization abilities of neural language models on 34 targeted test suites (33 adapted from previously published work, and 1 novel) covering a wide range of syntactic phenomena. Test suites are written using a standard format that allows for flexible predictions which more closely resemble those used in psycholinguistic studies, specifically allowing for predictions about interactions among multiple testing conditions. Performance on each test suite is reported as a Syntactic Generalization (SG) score. We group test suites into six syntactic circuits based on the linguistic representations needed to achieve high performance on each suite.

We train four classes of neural models and one baseline n-gram model on four datasets derived from a newswire corpus, consisting of 1, 5, 14, and 42 million tokens. While previous work has compared model architectures for a fixed dataset size (e.g. Wilcox et al., 2019b) and network sizes for a fixed architecture (e.g. van Schijndel et al.,

Materials and code can be found at https://github.com/cpllab/syntactic-generalization.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1725–1744
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

2019), our controlled regime allows us to make an apples-to-apples comparison across model architectures on a range of sizes. In addition, we evaluate several off-the-shelf models which were trained on datasets ranging up to 2 billion tokens.

Our results address the three questions posed above: First, for the range of model architectures and dataset sizes tested, we find a substantial dissociation between perplexity and SG score. Second, we find a larger effect of model inductive bias than training data size on SG score, a result that accords with van Schijndel et al. (2019). Models afforded explicit structural supervision during training outperform other models: One structurally supervised model is able to achieve the same SG scores as a purely sequence-based model trained on $\sim$100 times the number of tokens. Furthermore, several Transformer models achieve the same SG score as a Transformer trained on $\sim$200 times the amount of data. Third, we find that architectures have different relative advantages across types of syntactic tests, suggesting that the tested syntactic phenomena tap into different underlying processing capacities in the models.

2 Background

2.1 Perplexity

Standard language models are trained to predict the next token given a context of previous tokens. Language models are typically assessed by their *perplexity*, the inverse geometric mean of the joint probability of words $w_1, \ldots, w_N$ in a held-out test corpus C:

$$\mathrm{PPL}(C) = p(w_1, w_2, \ldots w_N)^{-\frac{1}{N}} \qquad (1)$$

Models with improved perplexity have also been shown to better match various human behavioral measures, such as gaze duration during reading (Frank and Bod, 2011; Fossum and Levy, 2012; Goodkind and Bicknell, 2018; Wilcox et al., 2020). However, a broad-coverage metric such as perplexity may not be ideal for assessing human-like syntactic knowledge for a variety of reasons. In principle, a sentence can appear with vanishingly low probability but still be grammatically well-formed, such as *Colorless green ideas sleep furiously* (Chomsky, 1957). While perplexity remains an integral part of language model evaluation, fine-grained linguistic assessment can provide both more challenging and more interpretable tests to evaluate neural models.

2.2 Targeted tests for syntactic generalization

Alternatively, a language model can be evaluated on its ability to make human-like generalizations for specific syntactic phenomena (Linzen et al., 2016; Lau et al., 2017; Gulordava et al., 2018). The targeted syntactic evaluation paradigm (Marvin and Linzen, 2018; Futrell et al., 2019) incorporates methods from psycholinguistic experiments, designing sentences which hold most lexical and syntactic features of each sentence constant while minimally varying features that determine grammaticality or surprise characteristics of the sentence. For example, given the two strings *The keys to the cabinet are on the table* and **The keys to the cabinet is on the table*, a model that has learned the proper subject–verb number agreement rules for English should assign a higher probability to the grammatical plural verb in the first sentence than to the ungrammatical singular verb in the second (Linzen et al., 2016).

Although some targeted syntactic evaluations, such as the example discussed above, involve simple comparisons of conditional probabilities of a word in its context, other evaluations are more complex. We can demonstrate this with an evaluation of models' "garden-pathing" behavior (Futrell et al., 2019). For example, the sentence *The child kicked in the chaos found her way back home* yields processing disruption for humans at the word *found*. This is because, up to right before that word, the part-of-speech ambiguous *kicked* is preferentially interpreted as the main verb of the sentence, whereas it turns out to be a passive participle in a reduced relative clause modifying *child*. This garden-path disambiguation effect is ameliorated by replacing *kicked* with *forgotten*, which is not part-of-speech ambiguous (B below; Trueswell et al., 1994) or by using an unreduced relative clause (C below; Ferreira and Clifton, 1986). In probabilistic language models, these garden-path disambiguation effects are well captured by word negative log probabilities, or SURPRISALS (Hale, 2001): $S(w|C) = -\log_2 p(w|C)$, which are independently well-established to predict human incremental processing difficulty over several orders of magnitude in word probability (Smith and Levy, 2013). A targeted syntactic evaluation for garden-pathing is provided by comparing surprisals at the disambiguating word *found* in the set of four examples below (Futrell et al., 2019):

(A) The child kicked in the chaos **found** ...

(B) The child forgotten in the chaos **found** . . .
(C) The child who was kicked in the chaos **found** . . .
(D) The child who was forgotten in the chaos **found** . . .

Successful human-like generalization involves three criteria: (i) *found* should be less surprising (i.e., more probable) in B than A; (ii) *found* should be more probable in C than A; (iii) the C–D surprisal difference should be smaller than the A–B surprisal difference—a 2 × 2 *interaction effect* on surprisal—because the syntactic disambiguation effect of not reducing the relative clause was achieved by using a part-of-speech unambiguous verb.

We will use these controlled tests to help us describe and test for human-like syntactic knowledge in language models.

2.3 Related work

The testing paradigm presented here differs in several crucial ways from recent, related syntactic assessments and provides complementary insights. Unlike Warstadt et al. (2019a), our approach does not involve fine-tuning, but rather assesses what syntactic knowledge is induced from the language modeling objective alone. The most closely related work is the Benchmark of Linguistic Minimal Pairs (Warstadt et al., 2020), which is a challenge set of automatically-generated sentence pairs also designed to test language models on a large set of syntactic phenomena. Our approach differs in important ways: we compare critical sentence regions instead of full-sentence probabilities, and employ a 2 × 2 paradigm with a strict, multi-fold success criterion inspired by psycholinguistics methodology. This allows us to factor out as many confounds as possible, such as the lexical frequency of individual tokens and low-level n-gram statistics.

3 Methods

We designed a controlled paradigm for systematically testing the relationship between two design choices — model class and dataset size — and two performance metrics — perplexity and syntactic generalization capacity. Section 3.1 describes the test suites collected for our evaluation, and Sections 3.2 and 3.3 describe the datasets and model classes investigated.

3.1 Test suites

We assemble a large number of test suites inspired by the methodology of experimental sentence-processing and psycholinguistic research. Each test suite contains a number of ITEMS (typically between 20 and 30), and each item appears in several CONDITIONS: across conditions, a given item will differ only according to a controlled manipulation designed to target a particular feature of grammatical knowledge. Each test suite contains at least one PREDICTION, which specifies inequalities between surprisal values at pairs of regions/conditions that should hold if a model has learned the appropriate syntactic generalization.

We expect language models which have learned the appropriate syntactic generalizations from their input to satisfy these inequalities without further fine-tuning. We compute accuracy on a test suite as the proportion of items for which the model's behavior conforms to the prediction. Most of our test suites involve 2×2 designs and a success criterion consisting of a conjunction of inequalities across conditions, as in the garden-pathing example described in Section 2.2.[1] Random baseline accuracy varies by test suite and is ∼25% overall. Most of these test suites and criteria are designed so that n-gram models cannot perform above chance for $n = 5$ (sometimes greater).

Syntactic coverage In order to assess the coverage of our test suites, we manually inspected the phenomena covered in Carnie (2012), a standard introductory syntax textbook. Of the 47 empirical phenomena reviewed in the summary sections at the end of each chapter, our tests target 16 (∼34%). These are evenly distributed across the whole range of subject matter, with tests targeting phenomena in 11 of the 15 chapters (∼73%).[2]

Modifiers Five test suites include paired modifier versions, where extra syntactically irrelevant (but semantically plausible) content, such as a prepositional phrase or relative clause, is inserted before the critical region being measured. We use these paired test suites to evaluate models' stability to intervening content within individual syntactic tests.

Circuits The test suites are divided into 6 syntactic circuits, based on the type of algorithm required to successfully process each construction. We give a brief overview of each circuit below.[3]

- **Agreement** is a constraint on the feature values of two co-varying tokens. For example,

[1] The exception is Center Embedding, which features a 2-condition design with a single-inequality criterion.
[2] For more details on this analysis, see Appendix A.
[3] A full overview of our test suites is given in Appendix B.

the number feature of a verb must agree with the number feature of its upstream subject. We include 3 *Subject-Verb Number Agreement* suites from Marvin and Linzen (2018).

- **Licensing** occurs when a particular token must exist within the scope of an upstream licensor token. Scope is determined by the tree-structural properties of the sentence. Test suites include *Negative Polarity Item Licensing (NPI)* (4 suites) and *Reflexive Pronoun Licensing* (6 suites), both from Marvin and Linzen (2018).

- **Garden-Path Effects** are well-studied syntactic phenomena that result from tree-structural ambiguities that give rise to locally-coherent but globally implausible syntactic parses. Garden-path test suites include *Main Verb / Reduced Relative Clause (MVRR)* (2 suites) and *NP/Z Garden-paths (NPZ)* (4 suites), both from Futrell et al. (2018).

- **Gross Syntactic Expectation** is a processor's expectation for large syntactic chunks such as verb phrases or sentences, and are often set up by subordinating conjunctions such as *while*, *although* and *despite*. Our tests for gross syntactic expectation include *Subordination* (4 suites) from Futrell et al. (2018).

- **Center Embedding** sentences are sentences recursively nested within each other. Subject and verbs must match in a first-in-last-out order, meaning models must approximate a stack-like data-structure in order to successfully process them. Our 2 suites of *Center Embedding* sentences come from the items presented in Wilcox et al. (2019a).

- **Long-Distance Dependencies** are co-variations between two tokens that span long distances in tree depth. Test suites include *Filler-Gap Dependencies (FGD)* (6 suites) from Wilcox et al. (2018) and Wilcox et al. (2019b), and 2 novel *Cleft* suites, described in detail below.

Novel test suite: Cleft We introduce one novel test suite that assesses models' ability to process pseudo-cleft constructions, which are used to put a particular syntactic constituent into focus via passive transformation. Consider Example (1):

BLLIP sizes:	XS	SM	MD	LG
# sentences	40K	200K	600K	1.8M
# tokens	1M	4.8M	14M	42M
# non-UNK types	24K	57K	100K	170K
# UNK types	68	70	71	74

Table 1: Statistics of training set for each corpus size.

(1) a. What he did after coming in from the rain was **eat a hot meal**. [DO/VP]
b. *What he devoured after coming in from the rain was **eat a hot meal**. [LEX/VP]
c. *What he did after coming in from the rain was **a hot meal**. [DO/NP]
d. What he devoured after coming in from the rain was **a hot meal**. [LEX/NP]

When this constituent is a verb, it must be replaced in the wh-clause that heads the sentence with the DO verb, as in (1a), below. However, when it is a noun, the lexical verb for which it serves as an object must be preserved, as in (1d). If models have properly learned the pseudo-cleft construction, then DO verbs should set up expectations for VPs (the region in bold should have a lower surprisal in (1a) than in (1b)) and lexicalized verbs should set up expectations for NPs (the region in bold should have a lower surprisal in (1d) than in (1c)).

3.2 Model training data

Corpora We train and evaluate models on English newswire corpora of four different sizes, obtained by randomly sampling sections from the Brown Laboratory for Linguistic Information Processing 1987-89 Corpus Release 1 (BLLIP; Charniak et al., 2000). The corpora are sampled such that the training set of each corpus is a proper subset of each larger corpus. We call these four corpora BLLIP-XS (40K sentences, 1M tokens); BLLIP-SM (200K sentences, 5M tokens); BLLIP-MD (600K sentences, 14M tokens); and BLLIP-LG (2M sentences, 42M tokens). Table 1 summarizes statistics of the training set for each corpus.

To ensure consistency in perplexity evaluation across datasets, we report perplexity scores achieved by the models on a shared held-out test set. We additionally use a shared held-out validation for tuning and early stopping.

We use the NLTK implementation of the Penn Treebank tokenizer to process all datasets (Bird and Loper, 2004; Marcus et al., 1993).

	# layers	# hidden units	Embedding size
LSTM	2	256	256
ON-LSTM	3	1150	400
RNNG	2	256	256
GPT-2	12	768	768

Table 2: Size of neural models in our controlled experiments.

BLLIP sizes:	XS	SM	MD	LG
LSTM	13.4M	30.5M	52.2M	88.1M
ON-LSTM	30.8M	44.2M	61.2M	89.2M
RNNG	22.8M	48.4M	81.1M	134.9M
GPT-2	124.4M	124.4M	124.4M	124.4M

Table 3: Parameter counts for neural models in our controlled experiments.

Out-of-vocabulary tokens For each corpus, we designate a token as OOV if the token appears fewer than two times in the training set. Our larger training datasets thus contain larger vocabularies than our smaller training datasets. This allows larger-training-set models to learn richer word-specific information, but may also harm perplexity evaluation because they have vocabulary items that are guaranteed to not appear in the BLLIP-XS test set. This means that perplexity scores across training dataset sizes will not be strictly comparable: if a larger-training-set model does better than a smaller-training-set model, we can be confident that it has meaningfully lower perplexity, but the reverse is not necessarily the case. The exception to the above is GPT-2, which uses sub-words from byte-pair encoding and has no OOVs (see also Footnote 6).

Unkification We follow the convention used by the Berkeley parser (Petrov and Klein, 2007), which maps OOVs to UNK classes which preserve fine-grained information such as orthographic case distinctions and morphological suffixes (e.g. `UNK-ed`, `UNK-ly`). Before training, we verified that the UNK classes in the test and validation sets were all present in the training set.

3.3 Model classes

In order to study the effects of model inductive bias and dataset size, we trained a fleet of models with varying inductive biases on each corpus. Because many of our test suites exploit ambiguities that arise from incremental processing, we restrict evaluation to left-to-right language models; future work could involve evaluation of bidirectional models (Devlin et al., 2018; Yang et al., 2019) on an appropriate subset of our test suites, and/or adaptation of our suites for use with bidirectional models (Goldberg, 2019). Training ran until convergence of perplexity on a held-out validation set. Wherever possible, we trained multiple seeds of each model class and corpus size. We use the model sizes and training hyperparameters reported in the papers introducing each model (Table 2).[4] The full parameter counts and perplexity scores for each model × corpus combination are given in Tables 3 and 4, respectively.

BLLIP sizes:	XS	SM	MD	LG
LSTM	98.19	65.52	59.05	57.09
ON-LSTM	71.76	54.00	56.37	56.38
RNNG	122.46	86.72	71.12	69.57
GPT-2	529.90	183.10	37.04	32.14
n-gram	240.21	158.60	125.58	106.09

Table 4: Perplexity averages achieved by each controlled model on each corpus. Perplexity scores across training dataset sizes are not always strictly comparable (see Section 3.2).

LSTM Our baseline neural model is a vanilla long short-term memory network (LSTM; Hochreiter and Schmidhuber, 1997) based on the boilerplate PyTorch implementation (Paszke et al., 2017).

Ordered-Neurons We consider the Ordered-Neurons LSTM architecture (ON-LSTM; Shen et al., 2019), which encodes an explicit bias towards modeling hierarchical structure.

RNNG Recurrent neural network grammars (RNNG; Dyer et al., 2016) model the joint probability of a sequence of words and its syntactic structure. RNNG requires labeled trees that contain complete constituency parses, which we produce for BLLIP sentences with an off-the-shelf constituency parser (Kitaev and Klein, 2018).[5] To compute surprisals from RNNG, we use word-synchronous beam search (Stern et al., 2017) to approximate the conditional probability of the current word given the context.

[4] Due to computational constraints, we performed only minimal tuning past these recommended hyperparameters.

[5] While the BLLIP corpus already contains Treebank-style parses, we strip the terminals and re-parse in order to obtain more accurate, up-to-date syntactic parses.

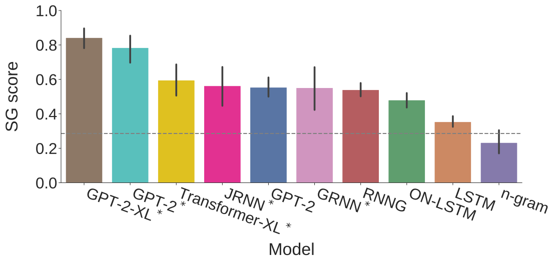

Figure 1: Average SG score by model class. Asterisks denote off-the-shelf models. Error bars denote bootstrapped 95% confidence intervals of the mean.

Transformer Transformer models (Vaswani et al., 2017) have recently gained popularity in language processing tasks. We use GPT-2 (Radford et al., 2019) as a representative Transformer model and train it from scratch on our BLLIP corpora.[6]

n-gram As a baseline, we consider a 5-gram model with modified Kneser-Ney smoothing.

3.4 Off-the-shelf models

We also test five off-the-shelf models: GRNN, trained on 90M tokens from Wikipedia (Gulordava et al., 2018); JRNN, trained on 800M tokens from the 1 Billion Word Benchmark (Jozefowicz et al., 2016); Transformer-XL, trained on 103M tokens from WikiText-103 (Dai et al., 2019); and the pre-trained GPT-2 and GPT-2-XL, trained on 40GB of web text (Radford et al., 2019). These models are orders of magnitude larger than our controlled ones in parameter count and/or training set size.

4 Results

Figure 1 shows the average accuracy of all models on the complete set of SG test suites. Asterisks denote off-the-shelf models. All neural models achieve a SG score significantly greater than a random baseline (dashed line). However, the range within neural models is notable, with the best-performing model (GPT-2-XL) scoring over twice as high as the worst-performing model (LSTM). Also notable are the controlled GPT-2 and RNNG models, which achieve comparable performance to Transformer-XL and JRNN, despite being trained on significantly smaller data sizes.

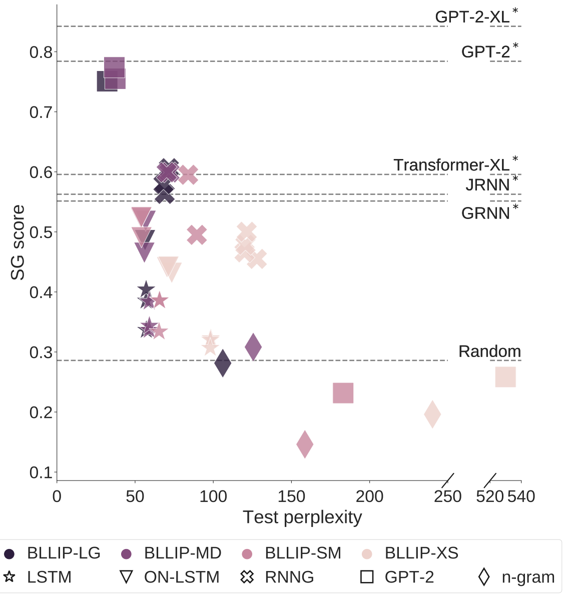

Figure 2: Relationship between SG score and perplexity on our held-out BLLIP test set for each model.

We now return to the three major issues presented in Section 1. In 4.1 we present evidence that SG score is dissociated from perplexity. In 4.2 we argue that model architecture accounts for larger gains in SG score than amount of training data. And in 4.3 we show that this cross-architecture difference is due largely to variance on a handful of key test suites.

4.1 Syntactic generalization and perplexity

Figure 2 shows the relationship between SG score and perplexity on the BLLIP test set across models and training set sizes. As expected, n-gram models never rise appreciably above chance in SG score. Among neural models, GPT-2 achieves both the worst (BLLIP-XS and BLLIP-SM) and best (BLLIP-MD and BLLIP-LG) performance; the impressive performance of these latter models comes with the caveat that the sub-words come from the pre-trained GPT-2 model, tacitly importing information from a larger training dataset (see further discussion in Section 4.5). For the remaining neural models, there is no simple relationship between perplexity and SG score, especially once training dataset size is controlled for (comparing points in Figure 2 of the same color). For example, there is a remarkable amount of variance in the SG score of models trained on BLLIP-LG not explained by perplexity. This suggests that targeted syntactic evaluation can reveal information that may be orthogonal to perplexity.

[6] Our GPT-2 code is based on `nshepperd/gpt-2`. The model vocabulary consists of byte-pair encoded sub-words extracted from the GPT-2 pre-trained model, not from the BLLIP training corpora. To calculate GPT-2 perplexities, we divide the sum of all sub-word conditional log-probabilities by the total number of words in the corpus.

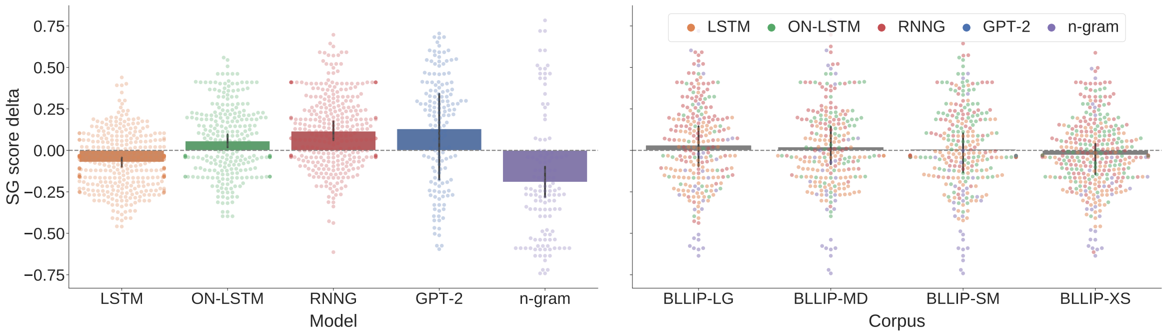

Figure 3: Main results of our controlled evaluation of model class and dataset size. SG score varies more by model class (left) than by training dataset size (right).

4.2 Inductive bias and data scale

In order to decouple the effects of model class and data scale from test suite difficulty, we represent a particular trained model's performance on each test suite as a delta relative to the average performance of all models on this test suite. Unless noted otherwise, the remainder of the figures in this section plot a score delta, aggregating these deltas within model classes or corpus types.

Figure 3 tracks the influence of model class and data scale across the model types tested in our experiments, with SG score deltas on the y-axis. The left-hand panel shows the difference in SG score by model class. We find that model class clearly influences SG score: for example, the error bars (bootstrapped 95% confidence intervals of the mean) for RNNG and LSTM do not overlap. The right-hand panel shows the difference in SG score delta by training dataset, and shows a much more minor increase in mean SG score as training data increases.

We tested the influence of these factors quantitatively using a linear mixed-effects regression model, predicting suite-level performance as a feature of model architecture and training dataset size (represented as log-number of words). Both features made statistically significant contributions to SG score (both $p < 0.001$). However, predictor ablation indicates that architecture affects regression model fit more (AIC=–581 when dataset size is ablated; AIC=–574 when architecture is ablated).[7]

Beyond the above analysis, our GPT-2 results offer another striking example of the influence of model architecture relative to data scale. Figure 2 shows that our controlled BLLIP-MD and BLLIP-LG GPT-2 models achieve roughly the same SG score as the pre-trained GPT-2 model, despite being trained on less than 1% of the data used by the pre-trained model. This suggests diminishing returns to training data scale for syntactic generalization performance.

4.3 Circuit-level effects on SG score

Figure 4 shows the breakdown at the circuit level by model architecture (left) and training dataset size (right). The right panel demonstrates little effect of dataset size on SG score delta within most circuits, except for Agreement, on which the models trained on our smallest dataset fare poorly. In the left panel we find substantial between-circuit differences across architectures. Linear mixed-effects analyses support this finding: interactions with circuit are significant for both training dataset size and model architecture, but stronger for the latter (AIC=–654 and AIC=–623 when size and architecture are respectively ablated).

While model inductive biases separate clearly in performance on some circuits, they have little effect on performance on Licensing. This minimally suggests that Licensing taps into a distinct syntactic process within language models. One potential explanation for this is that the interactions tested by Licensing involve tracking two co-varying tokens where the downstream token is optional (see e.g. Hu et al., 2020).

We show the circuit-level breakdown of absolute SG scores for all models (including off-the-shelf) in Figure 5. In general, the models that obtain high SG scores on average (as in Figure 1) also perform well across circuits: pre-trained GPT-2 and GPT-

[7] n-grams and/or GPT-2 could arguably be expected to have qualitatively different sensitivity to training dataset size (the latter due to byte-pair encoding), so we repeated the analyses here and in Section 4.3 excluding both architectures individually as well as simultaneously. In all cases the same qualitative patterns described in the main text hold.

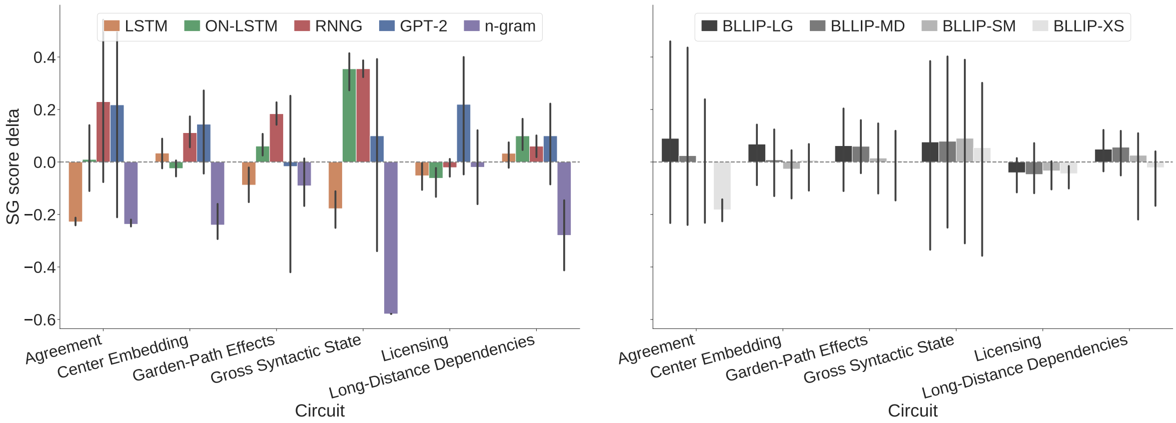

Figure 4: Controlled evaluation results, split across test suite circuits. Circuit-level differences in SG score vary more by model class (left) than by training dataset size (right).

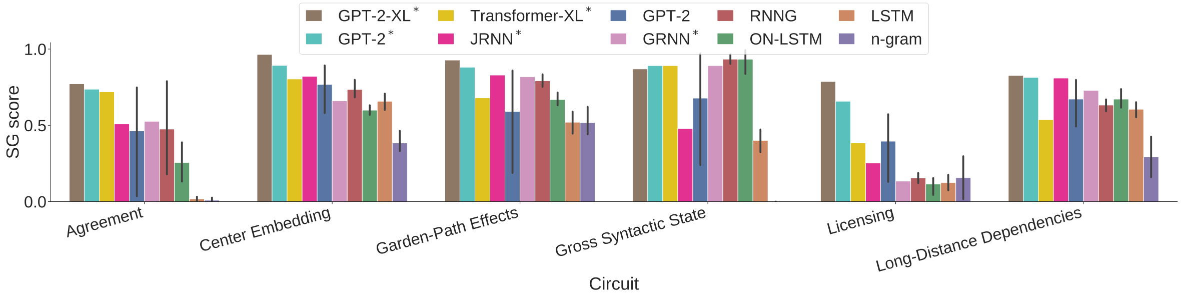

Figure 5: Evaluation results on all models, split across test suite circuits.

2-XL outperform all other models on each circuit, including Licensing, on which JRNN, GRNN, and most of our custom-trained models perform particularly poorly. Again, we highlight the impressive performance of RNNG: it achieves comparable average performance to GRNN on all circuits, despite being trained on a fraction of the data size.

4.4 Stability to modifiers

We separately investigate the degree to which models' syntactic generalizations are robustly stored in memory. For five test suites (Center Embedding, Cleft, MVRR, NPZ-Ambiguous, NPZ-Object), we designed minimally edited versions where syntactically irrelevant intervening content was inserted before the critical region. An ideal model should robustly represent syntactic features of its input across these modifier insertions.

In Figure 6 we plot models' average scores on these five test suites (dark bars) and their minimally edited versions (light bars), evaluating how robust each model is to intervening content. Among models in our controlled experiments, we see that model class clearly influences the degree to which predictions are affected by intervening content (compare e.g. the stability of RNNG to that of ON-LSTM). Some off-the-shelf models, such as GPT-2-XL, perform near ceiling on the original five test suites and are not affected at all by intervening content.

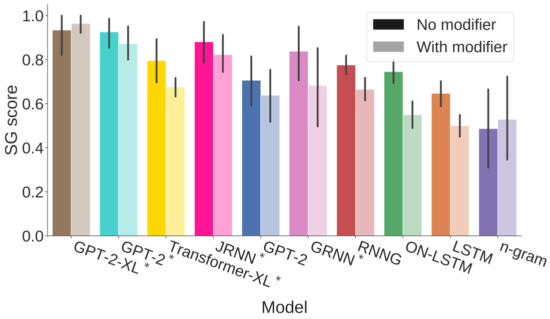

Figure 6: SG score on the pairs of test suites with and without intervening modifiers: Center Embedding, Cleft, MVRR, NPZ-Ambiguous, and NPZ-Object.

4.5 Effects of model pre-processing

The GPT-2 models trained and evaluated in this paper use a sub-word vocabulary learned by byte-pair encoding (BPE; Sennrich et al., 2016) to represent their inputs, while all other models represent and compute over word-level inputs. This byte-pair encoding was taken from the pre-trained GPT-2 model trained on a much larger corpus. The results reported for these models thus conflate a choice of model class (a deep Transformer architecture) and preprocessing standard (sub-word tokenization computed on a larger corpus). Some preliminary work suggests that sub-word tokenization is indeed responsible for much of the larger GPT-2 models' success: we find that GPT-2 models trained on word-level representations of BLLIP-LG and BLLIP-MD achieve good perplexity measures, but degrade sharply in SG score.

Peculiarities of the GPT-2 training regime may be responsible for its particularly bad performance on the smaller corpora. Its sub-word vocabulary was held constant across training corpora, meaning that the model vocabulary size also remained constant across corpora, unlike the other models tested. The poor performance of GPT-2 models trained on smaller corpora may thus be due to overparameterization, and not due to fundamental problems with the model architecture at small data scales. We leave a thorough investigation of the role of sub-word tokenization to future work.

5 Discussion

This work addresses multiple open questions about syntactic evaluations and their relationship to other language model assessments. Our results dissociate model perplexity and performance in syntactic generalization tests, suggesting that the two metrics capture complementary features of language model knowledge. In a controlled evaluation of different model classes and datasets, we find model architecture plays a more important role than training data scale in yielding correct syntactic generalizations. Our circuit-level analysis reveals consistent failure on Licensing but inconsistent behavior on other circuits, suggesting that different syntactic circuits make use of different underlying processing capacities. In addition to the insight these results provide about neural NLP systems, they also bear on questions central to cognitive science and linguistics, putting lower bounds on what syntactic knowledge can be acquired from string input alone.

Targeted syntactic evaluation is just one in a series of complementary methods being developed to assess the learning outcomes of neural language processing models. Other methods include classifying sentences as grammatical or ungrammatical (Warstadt et al., 2019b), decoding syntactic features from a model's internal state (Belinkov et al., 2017; Giulianelli et al., 2018), or transfer learning to a strictly syntactic task such as parsing or POS tagging (Hewitt and Manning, 2019). As each task brings an explicit set of assumptions, complementary assessment methods can collectively provide greater insight into models' learning outcomes.

Although this paper, together with Warstadt et al. (2020), report what is to our knowledge the largest-scale targeted syntactic evaluations to date, we emphasize that they are only first steps toward a comprehensive understanding of the syntactic capabilities of contemporary language models. This understanding will be further advanced by new targeted-evaluation test suites covering a still wider variety of syntactic phenomena, additional trained models with more varied hyperparameters and randomization seeds, and new architectural innovations. Humans develop extraordinary grammatical capabilities through exposure to natural linguistic input. It remains to be seen to just what extent contemporary artificial systems do the same.

Acknowledgments

The authors would like to thank the anonymous reviewers and Samuel R. Bowman for their feedback, Miguel Ballesteros for advice and technical guidance, and Tristan Thrush for technical assistance. J.H. is supported by the NIH under award number T32NS105587 and an NSF Graduate Research Fellowship. J.G. is supported by an Open Philanthropy AI Fellowship. R.P.L. gratefully acknowledges support from the MIT-IBM Watson AI Lab, a Google Faculty Research Award, and a Newton Brain Science Award.

References

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872.

Tom Bever. 1970. The cognitive basis for linguistic structures. In J.R. Hayes, editor, *Cognition and* *the Development of Language*, pages 279–362. New York: John Wiley & Sons.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Kathryn Bock and Carol A. Miller. 1991. Broken agreement. *Cognitive Psychology*, 23:45–93.

Andrew Carnie. 2012. *Syntax: A generative introduction*, volume 18. John Wiley & Sons.

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.

Rui P. Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics*.

Noam Chomsky. 1957. *Syntactic structures*. Walter de Gruyter.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA.

Stephen Crain and Janet Dean Fodor. 1985. How can grammars help parsers? In David Dowty, Lauri Kartunnen, and Arnold M. Zwicky, editors, *Natural Language Parsing: Psycholinguistic, Computational, and Theoretical Perspectives*, pages 940–128. Cambridge: Cambridge University Press.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Fernanda Ferreira and Charles Clifton, Jr. 1986. The independence of syntactic processing. *Journal of Memory and Language*, 25:348–368.

Victoria Fossum and Roger P. Levy. 2012. Sequential vs. hierarchical syntactic models of human incremental sentence processing. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics*, pages 61–69.

Stefan L Frank and Rens Bod. 2011. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22(6):829–834.

Lyn Frazier and Keith Rayner. 1982. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42.

Anastasia Giannakidou. 2011. Negative and positive polarity items: Variation, licensing, and compositionality. In *Semantics: An international handbook of natural language meaning*, volume 3, pages 1660–1712. Berlin: Mouton de Gruyter.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248.

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

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics*, pages 10–18.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the second* *meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8.

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.

Francis Roger Higgins. 1973. *The Pseudo-Cleft Construction in English*. Ph.D. thesis, MIT.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jennifer Hu, Sherry Yong Chen, and Roger P. Levy. 2020. A closer look at the performance of neural language models on reflexive anaphor licensing. In *Proceedings of the Meeting of the Society for Computation in Linguistics*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

William Ladusaw. 1979. *Polarity Sensitivity as Inherent Scope Relations*. Ph.D. thesis, University of Texas at Austin.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 5:1202–1247.

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

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

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.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users. In R. Duncan Luce, Robert R. Bush, and Eugene Galanter, editors, *Handbook of Mathematical Psychology*, volume II, pages 419–491. New York: John Wiley & Sons, Inc.

Don C. Mitchell. 1987. Lexical guidance in human parsing: Locus and processing characteristics. In Max Coltheart, editor, *Attention and Performance XII: The psychology of reading*. London: Erlbaum.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Neural Information Processing Systems Autodiff Workshop*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York. Association for Computational Linguistics.

Martin J. Pickering and Matthew J. Traxler. 1998. Plausibility and recovery from garden paths: An eye-tracking study. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 24(4):940–961.

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

Tanya Reinhart. 1981. Definite NP anaphora and c-command domains. *Linguistic Inquiry*, 12(4):605–635.

John Robert Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.

Marten van Schijndel and Tal Linzen. 2018. Modeling garden path effects without explicit hierarchical syntax. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5835–5841.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Nathaniel J. Smith and Roger P. Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128:302–319.

Adrian Staub. 2007. The parser doesn't ignore intransitivity, after all. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 33(3):550–569.

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.

Laurie A Stowe. 1986. Parsing wh-constructions: Evidence for on-line gap location. *Language & Cognitive Processes*, 1(3):227–245.

Patrick Sturt, Martin J. Pickering, and Matthew W. Crocker. 1999. Structural change and reanalysis difficulty in language comprehension. *Journal of Memory and Language*, 40:136–150.

John C. Trueswell, Michael K. Tanenhaus, and Susan M. Garnsey. 1994. Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language*, 33:285–318.

Shravan Vasishth, Sven Brüssow, Richard L Lewis, and Heiner Drenhaus. 2008. Processing polarity: How the ungrammatical intrudes on the grammatical. *Cognitive Science*, 32(4):685–712.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

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*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019a. CoLA: The Corpus of Linguistic Acceptability (with added annotations).

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ethan Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. Evaluating neural networks as models of human online language processing. In *Proceedings of the 42nd Meeting of the Cognitive Science Society (CogSci 2020)*. To appear.

Ethan Wilcox, Roger P. Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Ethan Wilcox, Roger P. 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*.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballestros, and Roger P. Levy. 2019b. 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*, pages 3302–3312, Minneapolis, Minnesota.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*.

A Syntactic coverage of test suites

In order to assess the coverage of our syntactic tests, we manually inspected the "Ideas, Rules and Constraints introduced in this Chapter" section for each chapter in Carnie (2012), a standard introductory syntax textbook. We included entries from these sections which are theory-neutral and refer to observable linguistic data. For example, we do not include *affix lowering* (Chapter 7) or *theta criterion* (Chapter 8) because these phenomena presuppose a commitment to one particular syntactic analysis.

We found that our tests covered 16 of the 47 phenomena presented (∼34%). Of the 15 chapters surveyed, our tests assessed phenomena in 11

Chapter 1: Generative Grammar	Lexical gender	
	Number	✓
	Person	
	Case	
Chapter 2: Parts of Speech	Parts of Speech	✓
	Plurality	✓
	Count vs. Mass Nouns	
	Argument Structure of Verbs	✓
Chapter 3: Constituency, Trees, Rules	Constituency Tests	
	Hierarchical Structure	✓
Chapter 4: Structural Relations	c-command	✓
	Government	
Chapter 5: Binding Theory	R-expression vs. Pronominals	
	Anaphoric expressions and their antecedents	✓
	Co-reference and co-indexation	
	Binding Principles (A, B, C)	✓
	Locality Constraints	✓
Chapter 6: X-Bar Theory	One Replacement	
	Do-so Replacement	
Chapter 7: Extending X-Bar Theory to Functional Categories	Fundamental Phrase Types of DP/CP/TP	
	Genitives: of-genitives and 's genitives	
	Subjects and Predicates	
	Clausal Embedding	✓
	Clausal	
	Tense/Finiteness and its restrictions	
	Yes/No Questions	
	Subject-Auxilliary Inversion	
Chapter 8: Constraining X-Bar Theory: The Lexicon	Thematic Relations	✓
	Internal Theta role vs. External Theta Roles	
	Expletive Pronouns and Expletive Insertion	
	Extended Projection Principle	
Chapter 9: Head-to-Head Movement	V → T Movement	
	T → C movement	✓
	Do-Support	
Chapter 10: DP Movement	Passive Constructions	✓
	DP-Raising	
Chapter 11: Wh-Movement	Wh-Movement	✓
	Structural Constraints on Wh-Movement (Island Constraints)	✓
	Wh in-Situ and Echo Questions	
Chapter 12: A Unified Theory of Movement	Universal Quantifiers vs. Existential Quantifiers	
	Quantificational Scope and Quantifier Raising	
Chapter 13: Extended VPs	Light Verbs	
	Object Shift (and end weight)	
	Ellipsis	
	Pseudogapping	
Chapter 14: Raising Control and Empty Categories	Control, Subject-to-Subject and Subject-to-Object Raising (ECM)	
Chapter 15: Advanced Topics in Binding Theory	Binding Principle A and B	✓

Table 5: Test suite coverage of syntactic phenomena presented in Carnie (2012).

(∼73%). We did not assess coverage from the last two chapters of the book, which explore alternative syntactic formalisms. The outcome of our manual inspection is given in Table 5.

A ✓indicates that some aspect of that phenomena was tested in one or more of our suites. ✓does not necessarily mean that the test suite was designed explicitly for the purpose of testing that phenomena, but merely that the phenomena was implicated in model success. For example, we place a ✓next to *Parts of Speech* because differentiation between verbs and nouns is necessary for models to succeed in the *Cleft Structure* tests.

B Description of test suites

In this work we have assembled a large number of test suites inspired by the methodology of experimental sentence-processing and psycholinguistic research. Each test suite contains a number of ITEMS, and each item appears in several CONDITIONS: across conditions, a given item will differ only according to a controlled manipulation designed to target a particular feature of grammatical knowledge. For each suite we define a SUCCESS CRITERION, which stipulates inequalities among conditional probabilities of sentence substrings.

In the main paper, a model's accuracy for a test suite is computed as the percentage of the test suite's items for which it satisfies the criterion. In this appendix, we briefly describe each test suite and the criterion used to determine whether a given model succeeds on each item of the test suite.

B.1 Notation

B.1.1 Sentence status

Following and building on linguistic traditions, we annotate examples as follows. Examples marked with a * violate a well-established grammatical constraint, and are ungrammatical. Examples marked with ? or ?? are not necessarily ungrammatical, but are marginal: for example, they may require an unusual interpretation of a word in order for the sentence to be grammatical. (More ?'s is roughly intended to indicate more severe marginality). Examples marked with ! are not ungrammatical, but induce severe processing difficulty that is measurable in real-time human sentence processing. For all test suites, we include references to established literature on the relevant grammatical and/or sentence-processing phenomena.

B.1.2 Success criteria

Criteria involve inequalities among conditional probabilities of sentence substrings given the complete sentence context preceding the substring. In describing criteria, we use $P(\cdot)$ for raw probabilities and $S(\cdot)$ for surprisals (negative log-probabilities), and leave the conditioning on preceding context implicit. For concision, we use subscripts on P and S to indicate the variant of the sentence within the test suite that we are referring to. In the first described test suite, CENTER EMBEDDING B.2, we show the criterion in both concise and fully spelled-out forms, to help clarify the conventions we are using in the concise form. All items within a given test suite share the same criterion for success.

We provide chance accuracy on the assumption that the order of probabilities among conditions for a given item is random. In some cases, exactly determining chance accuracy may require further assumptions about the distribution of these probabilities; in this case we provide an upper bound on chance accuracy.

B.2 Center embedding

Center embedding, the ability to embed a phrase in the middle of another phrase of the same type, is a hallmark feature of natural language syntax. Center-embedding creates NESTED SYNTACTIC DEPENDENCIES, which could pose a challenge for some language models. To succeed in generating expectations about how sentences will continue in the context of multiple center embedding, a model must maintain a representation not only of what words appear in the preceding context but also of the order of those words, and must predict that upcoming words occur in the appropriate order. In this test suite we use verb transitivity and subject–verb plausibility to test model capabilities in this respect. For example, A below is a correct center-embedding, but B is not:

(A) The painting$_{\mathrm{N}_1}$ that the artist$_{\mathrm{N}_2}$ painted$_{\mathrm{V}_2}$ deteriorated$_{\mathrm{V}_1}$. [correct]

(B) ??The painting$_{\mathrm{N}_1}$ that the artist$_{\mathrm{N}_2}$ deteriorated$_{\mathrm{V}_1}$ painted$_{\mathrm{V}_2}$. [incorrect]

Here, N_i and V_i correspond to matched subject–verb pairs.

In the WITH-MODIFIER version of the test suite, we postmodify N_2 with a relative clause to increase the linear distance over which the nested dependencies must be tracked, potentially leading to a harder test suite:

(A) The painting$_{\mathrm{N}_1}$ that the artist$_{\mathrm{N}_2}$ who lived long ago painted$_{\mathrm{V}_2}$ deteriorated$_{\mathrm{V}_1}$. [correct]

(B) #The painting$_{\mathrm{N}_1}$ that the artist$_{\mathrm{N}_2}$ who lived long ago deteriorated$_{\mathrm{V}_1}$ painted$_{\mathrm{V}_2}$. [incorrect]

Criterion The probability of the verb sequence in the correct variant should be higher than the probability of the verb sequence in the incorrect variant:

$$P_{\mathrm{A}}(\mathrm{V}_2\mathrm{V}_1) > P_{\mathrm{B}}(\mathrm{V}_1\mathrm{V}_2)$$

In full form, this criterion for the example item in the no-modifier version of this test suite would be:

$$P(\text{painted deteriorated}|\text{The painting that the artist}) > P(\text{deteriorated painted}|\text{The painting that the artist})$$

Chance performance on these center-embedding test suites would be 50%.

References Miller and Chomsky (1963);Wilcox et al. (2019a)

B.3 Pseudo-clefting

The pseudo-cleft construction involves (i) an extraction of a TARGETED CONSTITUENT from a sentence and (ii) a constituent that provides the semantic contents of the targeted constituent and must match it in syntactic category, where (i) and (ii) are linked by the copula. The pseudo-cleft construction can target both NPs and VPs; in the latter case, the VP of the free relative becomes an inflected form of *do*. This means that a free relative subject plus the copula can set up a requirement for the syntactic category that comes next. If the free relative clause has a *do* VP without a direct object, then the main-clause postcopular predicate can be a VP (A below). Otherwise, the postcopular predicate must be an NP (C below):

(A) What the worker did was $\overbrace{\text{board the plane}}^{\text{VP}}$.

(B) ?What the worker did was $\overbrace{\text{the plane}}^{\text{NP}}$.

(C) What the worker repaired was $\overbrace{\text{the plane}}^{\text{NP}}$.

(D) *What the worker repaired was $\overbrace{\text{board the plane}}^{\text{VP}}$.

Criterion The postcopular predicate should be more surprising when its syntactic category mismatches the cleft, averaging across VP and NP postcopular predicates:

$$S_{\mathrm{D}}(\mathrm{VP}) + S_{\mathrm{B}}(\mathrm{NP}) > S_{\mathrm{C}}(\mathrm{NP}) + S_{\mathrm{A}}(\mathrm{VP})$$

Chance is 50%. A more stringent criterion would be to apply this requirement separately for each of NP and VP postcopular predicates:

$$S_{\mathrm{D}}(\mathrm{VP}) > S_{\mathrm{A}}(\mathrm{VP}) \wedge S_{\mathrm{B}}(\mathrm{NP}) > S_{\mathrm{C}}(\mathrm{NP})$$

However, it is often possible to use an NP postcopular predicate with a *do* cleft through semantic coercion (e.g., in B "did" can be interpreted as "fixed" or "was responsible for"), so we felt that this latter criterion might be too stringent.

References Higgins (1973)

B.4 Filler–gap dependencies

Consider the following sentence, in which all arguments and adjuncts appear "in situ" (in the syntactic position at which they are normally interpreted semantically):

I know that our uncle grabbed the food
in front of the guests at the holiday party.

A FILLER–GAP DEPENDENCY can be created by EXTRACTING any of a number of elements from the subordinate clause, including *our uncle* (subject extraction), *the food* (object extraction) or *the guests* (extraction from a prepositional phrase). These possibilities serve as the basis for several test suites on filler–gap dependencies.

References Ross (1967); Crain and Fodor (1985); Stowe (1986); Wilcox et al. (2018); Chowdhury and Zamparelli (2018); Chaves (2020)

B.4.1 Subject extractions

(A) I know that $\overbrace{\text{our uncle}}^{\alpha}$ grabbed the food in front of the guests at the holiday party. [THAT, NO GAP]

(B) *I know who $\overbrace{\text{our uncle}}^{\alpha}$ grabbed the food in front of the guests at the holiday party. [WH, NO GAP]

(C) *I know that $\overbrace{\text{grabbed}}^{\beta}$ the food in front of the guests at the holiday party. [THAT, GAP]

(D) I know who $\overbrace{\text{grabbed}}^{\beta}$ the food in front of the guests at the holiday party. [WH, GAP]

Criterion We require that a model successfully pass a two-part criterion for each item: the *wh*-filler should make the unextracted subject α more surprising in the NO-GAP conditions and should make the post-gap material β less surprising in the GAP conditions:

$$S_{\text{B}}(\alpha) > S_{\text{A}}(\alpha) \wedge S_{\text{C}}(\beta) > S_{\text{D}}(\beta)$$

Chance is 25%.

B.4.2 Object extractions

The logic of this test suite is the same as that for subject extraction above. Note that we use obligatorily transitive embedded verbs, so that omitting a direct object should be highly surprising when there is no filler, as in C.

(A) I know that our uncle grabbed $\overbrace{\text{the food}}^{\alpha}$ in front of the guests at the holiday party. [THAT, NO GAP]

(B) *I know what our uncle grabbed $\overbrace{\text{the food}}^{\alpha}$ in front of the guests at the holiday party. [WH, NO GAP]

(C) ??I know that our uncle grabbed $\overbrace{\text{in front of}}^{\beta}$ the guests at the holiday party. [THAT, GAP]

(D) I know what our uncle grabbed $\overbrace{\text{in front of}}^{\beta}$ in front of the guests at the holiday party. [WH, GAP]

Criterion

$$S_{\text{B}}(\alpha) > S_{\text{A}}(\alpha) \wedge S_{\text{C}}(\beta) > S_{\text{D}}(\beta)$$

B.4.3 Extraction from prepositional phrases

The logic of this test suite is the same as that for subject and object extractions above.

(A) I know that our uncle grabbed the food in front of $\overbrace{\text{the guests}}^{\alpha}$ at the holiday party. [THAT, NO GAP]

(B) *I know who our uncle grabbed the food in front of $\overbrace{\text{the guests}}^{\alpha}$ at the holiday party. [WH, NO GAP]

(C) *I know that our uncle grabbed the food in front of $\overbrace{\text{at the holiday party}}^{\beta}$. [THAT, GAP]

(D) I know who our uncle grabbed the food in front of $\overbrace{\text{at the holiday party}}^{\beta}$. [WH, GAP]

Criterion

$$S_{\text{B}}(\alpha) > S_{\text{A}}(\alpha) \wedge S_{\text{C}}(\beta) > S_{\text{D}}(\beta)$$

B.4.4 Tests for unboundedness

Filler–gap dependencies are "unbounded" in the sense that there is no limit to how many clausal levels above the gap the filler can be extracted. This serves as the basis for harder versions of the object-extracted test suites, involving three or four levels of clausal embedding. Example [THAT, NO GAP] sentences are given below:

> I know that our mother said her friend remarked that the park attendant reported your friend threw the plastic into the trash can. [3 levels of embedding]
>
> I know that our mother said her friend remarked that the park attendant reported the cop thinks your friend threw the plastic into the trash can. [4 levels of embedding]

These base sentences give rise to 4-condition test suites using the same manipulations as for the basic object-extraction test suite (Section B.4.2), and the criterion for success is the same.

B.5 Main-verb/reduced-relative garden-path disambiguation

This is one of the best-studied instances of syntactic garden-pathing in the psycholinguistics literature. An example 4-condition item is given below:

(A) !The child kicked in the chaos $\overbrace{\text{found}}^{\mathrm{V}^*}$ her way back home. [REDUCED, AMBIG]

(B) The child who was kicked in the chaos $\overbrace{\text{found}}^{\mathrm{V}^*}$ her way back home.

(C) The child forgotten in the chaos $\overbrace{\text{found}}^{\mathrm{V}^*}$ her way back home.

(D) The child who was forgotten in the chaos $\overbrace{\text{found}}^{\mathrm{V}^*}$ her way back home.

Criterion Relative to the [REDUCED, AMBIG] condition, not reducing the relative clause should make V* less surprising, as should changing the participial verb to one that is the same form as a simple past-tense verb. Additionally, the effect of not reducing the relative clause on V* surprisal should be smaller for unambiguous participial verbs than for participial verbs:

$$S_{\text{A}}(\text{V}^*) > S_{\text{B}}(\text{V}^*) \wedge S_{\text{A}}(\text{V}^*) > S_{\text{C}}(\text{V}^*) \wedge \\ S_{\text{A}}(\text{V}^*) - S_{\text{B}}(\text{V}^*) > S_{\text{C}}(\text{V}^*) - S_{\text{D}}(\text{V}^*)$$

Chance is somewhere below 25%.

References Bever (1970); Ferreira and Clifton (1986); Trueswell et al. (1994); van Schijndel and Linzen (2018); Futrell et al. (2019)

B.6 Negative Polarity Licensing

The words *any* and *ever*, in their most common uses, are “negative polarity items” (NPIs): they can only be used in an appropriate syntactic-semantic environment—to a first approximation, in the scope of negation. For example, the determiner *no* can license NPIs, but its NP has to structurally command the NPI. Below, A and D are acceptable, because *no* is the determiner for the subject noun *managers*. There is no negation in C so the NPI is unlicensed and the sentence is unacceptable; crucially, however, B is unacceptable despite the presence of *no* earlier in the sentence, because *no* is embedded inside a modifier of the main-clause subject and thus does not command the NPI.

(A) No managers that respected the guard have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [+NEG,−DISTRACTOR]

(B) *The managers that respected no guard have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [−NEG,+DISTRACTOR]

(C) *The managers that respected the guard have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [−NEG,−DISTRACTOR]

(D) No managers that respected no guard have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [+NEG,+DISTRACTOR]

In the above test suite, the “distractor” position for *no* is inside a subject-extracted relative clause modifying the main-clause subject. We also used a variant test suite in which these relative clauses are object-extracted:

(A) No managers that the guard respected have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [+NEG,−DISTRACTOR]

(B) *The managers that no guard respected have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [−NEG,+DISTRACTOR]

(C) *The managers that the guard respected have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [−NEG,−DISTRACTOR]

(D) No managers that no guard respected have had $\overbrace{\text{any}}^{\text{NPI}}$ luck. [+NEG,+DISTRACTOR]

The above two test suites use *any* as the NPI; we also use test suites with *ever* as the NPI. Subject-extracted relative clause example:

(A) No managers that respected the guard have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [+NEG,−DISTRACTOR]

(B) *The managers that respected no guard have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [−NEG,+DISTRACTOR]

(C) *The managers that respected the guard have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [−NEG,−DISTRACTOR]

(D) No managers that respected no guard have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [+NEG,+DISTRACTOR]

Object-extracted relative clause example:

(A) No managers that the guard respected have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [+NEG,−DISTRACTOR]

(B) *The managers that no guard respected have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [−NEG,+DISTRACTOR]

(C) *The managers that the guard respected have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [−NEG,−DISTRACTOR]

(D) No managers that no guard respected have $\overbrace{\text{ever}}^{\text{NPI}}$ gotten old. [+NEG,+DISTRACTOR]

Criterion Changing the main-clause subject’s determiner from *The* to *No* should increase the probability of the NPI where it appears, regardless of whether there is a distractor *no* in the subject-modifying relative clause. Furthermore, when there is exactly one *no* in the sentence, the NPI should be higher-probability when it is in a licensing position rather than in a distractor position:

$$P_{\text{A}}(\text{NPI}) > P_{\text{C}}(\text{NPI}) \wedge P_{\text{D}}(\text{NPI}) > P_{\text{B}}(\text{NPI}) \wedge \\ P_{\text{A}}(\text{NPI}) > P_{\text{B}}(\text{NPI})$$

Chance is $\frac{5}{32}$.

References Ladusaw (1979); Vasishth et al. (2008); Giannakidou (2011); Marvin and Linzen (2018); Futrell et al. (2018)

B.7 NP/Z garden-path ambiguity

This is another well-studied syntactic garden-pathing configuration. In A below, the NP *the waters* introduces a local syntactic ambiguity: it could be (1) the direct object of *crossed*, in which case the sentence-initial subordinate clause has not yet ended, or (2) the subject of the main clause, in which case *crossed* is used intransitively and is the last word of the sentence-initial subordinate clause. (This was dubbed “NP/Z” by Sturt et al. (1999) because the subordinate-clause verb might have either an NP object or a Z(ero), i.e. null, object.) The next word, *remained*, is only compatible with (2); the ruling out of (1) generally yields increased processing difficulty for human comprehenders. Marking the end of the subordinate clause with a comma, as in B, makes the sentence easier at V*, as does an obligatorily intransitive subordinate-clause verb, as in C.

(A) !As the ship crossed the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm. [TRANS,NO COMMA]

(B) As the ship crossed, the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm. [TRANS,COMMA]

(C) As the ship drifted the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm. [INTRANS,NO COMMA]

(D) As the ship drifted, the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm. [INTRANS,COMMA]

Criterion Similar to the main-verb/reduced-relative garden-pathing ambiguity, a model must pass a three-part criterion. Relative to A, either marking the subordinate-clause end with a comma or using an obligatorily intransitive verb in the subordinate clause should reduce the surprisal of V*. Furthermore, the surprisal-reduction effect of the comma should be smaller when the subordinate-clause verb is intransitive than when it is transitive:

$$S_{\text{A}}(\text{V}^*) > S_{\text{B}}(\text{V}^*) \wedge S_{\text{A}}(\text{V}^*) > S_{\text{C}}(\text{V}^*) \wedge$$
$$S_{\text{A}}(\text{V}^*) - S_{\text{B}}(\text{V}^*) > S_{\text{C}}(\text{V}^*) - S_{\text{D}}(\text{V}^*)$$

We also use an NP/Z test suite where the second means of disambiguation is not changing the subordinate-clause verb to an intransitive, but rather giving the transitive subordinate-clause verb an overt direct object. For the above example item, the first two conditions are the same and the other two conditions would be:

(C) As the ship crossed the sea the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm.

(D) As the ship crossed the sea, the waters $\overbrace{\text{remained}}^{\text{V}^*}$ blue and calm.

The success criterion remains the same.

Finally, we create harder versions of both the above test suites by adding a postmodifier to the main-clause subject (in the above example, *the waters* becomes *the waters of the Atlantic Ocean*).

References Frazier and Rayner (1982); Mitchell (1987); Pickering and Traxler (1998); Sturt et al. (1999); Staub (2007)

B.8 Subject–verb number agreement

This task tests a language model for how well it predicts the number marking on English finite present-tense verbs (whether it should be the third-person *singular* form, or the non-third-person-singular form, generally referred to as the *plural* form for simplicity, although technically this is the form for first- and second-person singular as well). In controlled, targeted versions of this test, multiple NP precede the verb: the verb’s actual subject, as well as a DISTRACTOR NP with number that is different from that of the subject. A successful language model should place higher probability on the verbform matching that of the subject, not the distractor. We have three versions of this test suite: one where the distractor is in a prepositional phrase postmodifier of the subject:

(A) The farmer near the clerks knows$_{\text{V}_{\text{sg}}}$ many people.

(B) *The farmer near the clerks know$_{\text{V}_{\text{pl}}}$ many people.

(C) The farmers near the clerk know$_{\text{V}_{\text{pl}}}$ many people.

(D) *The farmers near the clerk knows$_{\text{V}_{\text{sg}}}$ many people.

one in which the distractor is in a subject-extracted relative clause postmodifier of the subject:

(A) The farmer that embarrassed the clerks knows$_{\text{V}_{\text{sg}}}$ many people.

(B) *The farmer that embarrassed the clerks know$_{\mathrm{V}_{\mathrm{pl}}}$ many people.

(C) The farmers that embarrassed the clerk know$_{\mathrm{V}_{\mathrm{pl}}}$ many people.

(D) *The farmers that embarrassed the clerk knows$_{\mathrm{V}_{\mathrm{sg}}}$ many people.

and one in which the distractor is in an object-extracted relative clause postmodifier of the subject:

(A) The farmer that the clerks embarrassed knows$_{\mathrm{V}_{\mathrm{sg}}}$ many people.

(B) *The farmer that the clerks embarrassed know$_{\mathrm{V}_{\mathrm{pl}}}$ many people.

(C) The farmers that the clerk embarrassed know$_{\mathrm{V}_{\mathrm{pl}}}$ many people.

(D) *The farmers that the clerk embarrassed knows$_{\mathrm{V}_{\mathrm{sg}}}$ many people.

Criterion Following Linzen et al. (2016) and Marvin and Linzen (2018), we require successful discrimination of the preferred upcoming verbform of the given lemma (rather than, for example, successful discrimination of the better context given a particular verbform). For success we require that a model successfully predicts the preferred verbform for *both* the singular- and plural-subject versions of an item:

$$P_{\mathrm{A}}(\mathrm{V}_{\mathrm{sg}}) > P_{\mathrm{B}}(\mathrm{V}_{\mathrm{pl}}) \wedge P_{\mathrm{C}}(\mathrm{V}_{\mathrm{pl}}) > P_{\mathrm{D}}(\mathrm{V}_{\mathrm{sg}})$$

Chance performance is thus 25%, though a context-insensitive baseline that places different probabilities on V_{sg} and V_{pl} would score 50%.

References Bock and Miller (1991); Linzen et al. (2016); Marvin and Linzen (2018)

B.9 Reflexive pronoun licensing

The noun phrase with which a reflexive pronoun (*herself*, *himself*, *themselves*) corefers must command it in a sense similar to that relevant for negative-polarity items (Section B.6). In the below example, the reflexive pronoun ending the sentence can only corefer to the subject of the sentence, *author*, with which it must agree in number: a singular subject requires a singular reflexive R_{sg}, and a plural subject requires a plural reflexive R_{pl}.

(A) The author next to the senators hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(B) *The authors next to the senator hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(C) The authors next to the senator hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

(D) *The authors next to the senator hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

We generated a pair of test suites—one in which the singular reflexive is *herself*, and another where the singular reflexive is *himself*, on the template of the above example, where the distractor NP is in a prepositional-phrase postmodifier of the subject NP. We also generated a similar pair of test suites where the distractor NP is inside a subject-extracted relative clause modifying the subject:

(A) The author that liked the senators hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(B) *The authors that liked the senator hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(C) The authors that liked the senator hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

(D) *The authors that liked the senator hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

and a pair of test suites where the distractor NP is inside an object-extracted relative clause modifying the subject:

(A) The author that the senators liked hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(B) *The authors that the senator liked hurt herself$_{\mathrm{R}_{\mathrm{sg.fem}}}$.

(C) The authors that the senator liked hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

(D) *The authors that the senator liked hurt themselves$_{\mathrm{R}_{\mathrm{pl}}}$.

Criterion For each item in each test suite, we require that for both the singular and the plural versions of the reflexive pronoun the model assign higher conditional probability in the correct licensing context than in the incorrect licensing context:

$$P_{\mathrm{A}}(\mathrm{R}_{\mathrm{sg}}) > P_{\mathrm{B}}(\mathrm{R}_{\mathrm{sg}}) \wedge P_{\mathrm{C}}(\mathrm{R}_{\mathrm{pl}}) > P_{\mathrm{D}}(\mathrm{R}_{\mathrm{pl}})$$

Chance is 25%.

References Reinhart (1981); Marvin and Linzen (2018)

B.10 Subordination

Beginning a sentence with *As*, *When*, *Before*, *After*, or *Because*, implies that an immediately following clause is not the main clause of the sentence, as would have otherwise been the case, but instead is a SUBORDINATE CLAUSE that must be followed by the main clause. Ending the sentence without a main clause, as in B, is problematic. Conversely, following an initial clause with a second clause MC (without linking it to the initial clause with *and*, *but*, *despite*, or a similar coordinator or subordinator), as in C below, is unexpected and odd.

(A) The minister praised the building $\overbrace{\quad}^{\textsc{End}}$.

(B) *After the minister praised the building $\overbrace{\quad}^{\textsc{End}}$.

(C) ??The minister praised the building, $\overbrace{\text{it started to rain.}}^{\text{MC}}$

(D) After the minster praised the building, $\overbrace{\text{it started to rain.}}^{\text{MC}}$

In addition to the base test suite exemplified by the item above, we include three versions with longer and more complex initial clauses, which may make the test suite more difficult. In the first of these versions, we postmodify both the subject and object of the initial clauses with prepositional phrases:

the minister praised the building
↓
the minister in the dark suit and white tie praised the new building on the town's main square

In the second of these versions, the postmodifiers are subject-extracted relative clauses:

the minister praised the building
↓
the minister who wore a black suit praised the new building that was built by the square

In the third of these versions, the postmodifiers are object-extracted relative clauses:

the minister praised the building
↓
the minister who the mayor had invited praised the new building that the businessman had built downtown

Criterion Introducing a subordinator at the beginning of the sentence should make an ending without a second clause less probable, and should make a second clause more probable:

$$P_{\text{A}}(\textsc{End}) > P_{\text{B}}(\textsc{End}) \wedge P_{\text{D}}(\text{MC}) < P_{\text{C}}(\text{MC})$$

References Futrell et al. (2018)