

# BLiMP : A Benchmark of Linguistic Minimal Pairs for English

Alex Warstadt<sup>1</sup>, Alicia Parrish<sup>1</sup>, Haokun Liu<sup>2</sup>, Anhad Mohananey<sup>2</sup>,  
Wei Peng<sup>2</sup>, Sheng-Fu Wang<sup>1</sup>, Samuel R. Bowman<sup>1,2,3</sup>



<sup>1</sup>Dept. of Linguistics  
New York University

<sup>2</sup>Dept. of Computer Science  
New York University

<sup>3</sup>Center for Data Science  
New York University

Correspondence: [warstadt@nyu.edu](mailto:warstadt@nyu.edu)


## Abstract



We introduce The Benchmark of Linguistic Minimal Pairs (shortened to BLiMP, or ) , a challenge set for evaluating what language models (LMs) know about major grammatical phenomena in English.  consists of 67 sub-datasets, each containing 1000 minimal pairs isolating specific contrasts in syntax, morphology, or semantics. The data is automatically generated according from expert-crafted grammars, and aggregate human agreement with the labels is 96.4%. We use it to evaluate  $n$ -gram, LSTM, and Transformer (GPT-2 and Transformer-XL) LMs. We find that state-of-the-art models identify morphological contrasts reliably, but they struggle with semantic restrictions on the distribution of quantifiers and negative polarity items and subtle syntactic phenomena such as extraction islands.



## 1 Introduction


Current neural networks for language understanding rely heavily on unsupervised pretraining tasks like language modeling. However, it is still an open question to what degree different linguistic phenomena are represented by state-of-the-art language models (LMs). Many recent studies (e.g., Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018) have advanced our understanding in this area by evaluating LMs’ preferences between *minimal pairs* of sentences, as in Example (1). However, these studies have used vastly different analysis metrics and focused on a small set of linguistic paradigms, making a big-picture comparison between these studies limited.

- (1) a. The cat annoys Tim. (grammatical)  
b. The cat annoy Tim. (ungrammatical)

We introduce the Benchmark of Linguistic Minimal Pairs (shortened to BLiMP or just ) a

linguistically-motivated benchmark for assessing language models’ knowledge across a wide variety of English phenomena, encapsulating both previously studied and novel contrasts.  consists of 67 datasets automatically generated from expert-crafted grammars, each containing 1000 minimal pairs. Human validation with crowd workers shows that humans overwhelmingly agree with the contrasts in .

We use  to evaluate several pretrained LMs: Transformer-based LMs GPT-2 (Radford et al., 2019) and Transformer-XL (Dai et al., 2019), an LSTM LM trained by (Gulordava et al., 2019), and a  $n$ -gram LM. This experiment gives a sense of which phenomena are hard or easy learn for LMs in general, and the extent to which unrelated models perform poorly with the same phenomena. We conclude that current neural LMs robustly learn agreement phenomena and even some subtle syntactic phenomena such as ellipsis and control/raising. They perform comparatively worse (and well below human baseline) on minimal pairs related to argument structure and the licensing of negative polarity items and quantifiers. All models perform at or near chance on extraction islands, which we conclude is the most challenging phenomenon covered by . Overall, we note that all models we evaluate fall short of human performance by a wide margin. GPT-2, which performs the best, does match (even just barely exceeds) human performance on some grammatical phenomena, but remains 8 percentage points below human performance overall.

We conduct additional experiments to investigate the effect of training size on LSTM model performance on . We show that learning trajectories differ, sometimes drastically, across different paradigms in the dataset, with some phenomena like anaphor agreement showing consistent improvement as training size increases, and other phenomena remaining near chance despite

Phenomenon	N	Acceptable Example	Unacceptable Example
Anaphor agreement	2	<i>Many girls insulted <u>themselves</u>.</i>	<i>Many girls insulted <u>herself</u>.</i>
Argument structure	9	<i>Rose wasn't <u>disturbing</u> Mark.</i>	<i>Rose wasn't <u>boasting</u> Mark.</i>
Binding	7	<i>It's <u>himself</u> who <u>Robert</u> attacked.</i>	<i>It's <u>himself</u> who <u>attacked</u> Robert.</i>
Control/Raising	5	<i>Kevin isn't <u>irritating</u> to work with.</i>	<i>Kevin isn't <u>bound</u> to work with.</i>
Determiner-Noun agr.	8	<i>Rachelle had bought that <u>chair</u>.</i>	<i>Rachelle had bought that <u>chairs</u>.</i>
Ellipsis	2	<i>Anne's doctor cleans <u>one important</u> book and Stacey cleans a few.</i>	<i>Anne's doctor cleans one book and Stacey cleans a few <u>important</u>.</i>
Filler-Gap	7	<i>Brett knew <u>what</u> many waiters find.</i>	<i>Brett knew <u>that</u> many waiters find.</i>
Irregular forms	2	<i>Aaron <u>broke</u> the unicycle.</i>	<i>Aaron <u>broken</u> the unicycle.</i>
Island effects	8	<i>Which <u>bikes</u> is John fixing?</i>	<i>Which is John fixing <u>bikes</u>?</i>
NPI licensing	7	<i>The truck has <u>clearly</u> tipped over.</i>	<i>The truck has <u>ever</u> tipped over.</i>
Quantifiers	4	<i>There was <u>a</u> cat annoying Alice.</i>	<i>There was <u>each</u> cat annoying Alice.</i>
Subject-Verb agr.	6	<i>These casseroles <u>disgust</u> Kayla.</i>	<i>These casseroles <u>disgusts</u> Kayla.</i>

Table 1: Minimal pairs from each of the twelve linguistic phenomenon categories covered by 🧠. Minimal differences are underlined.  $N$  is the number of 1000-example minimal pair paradigms within each broad category.

increases in training size, like with NPIs and extraction islands. We also compare overall sentence probability to two other built-in metrics coded on 🧠 and find that the chosen metric changes how we evaluate relative model performance.

## 2 Background & Related Work

### 2.1 Language Models

The objective of a language model is to give a probability distribution over the possible strings of a language. Language models can be built on neural network models or non-neural network models. Due to their unsupervised nature, they can be trained without external annotations. More recently, neural network based language modeling has been shown to be a strong pretraining task (Howard and Ruder, 2018; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019) for natural language understanding tasks. Some recent models, such as BERT (Devlin et al., 2019) use closely related tasks such as masked language modeling.

In the last decade, we have seen two major paradigm shifts in the state of the art for language modeling. The first major shift for language modeling was the movement from statistics-based methods based on  $n$ -grams (see Chen and Goodman, 1999) to neural methods such as LSTMs (Sundermeyer et al., 2012), which directly optimize on the task of predicting the next word. More recently, transformer based architectures (Radford et al., 2018) employ attention as a tool, treating generative language modeling as a pretraining task. Although it is reasonably clear that these

shifts have resulted in stronger language models, the primary metric of performance is perplexity, which cannot give detailed insight into these models' linguistic knowledge. Evaluation on downstream task benchmarks (Wang et al., 2018, 2019a) is more informative, but might not present a broad enough challenge or represent grammatical distinctions at a sufficiently fine-grained level.

### 2.2 Evaluating Linguistic Knowledge

A large number of recent studies has used acceptability judgments to reveal what neural networks know about grammar. One branch of this literature has focused on using minimal pairs to infer whether LMs learn about specific linguistic phenomena. Table 2 gives a summary of work that has studied linguistic phenomena in this way. For instance, Linzen et al. (2016) look closely at minimal pairs contrasting subject-verb agreement. Marvin and Linzen (2018) look at a larger set of phenomena, including negative polarity item licensing and reflexive licensing. However, a relatively small set of phenomena is covered by these studies, to the exclusion of well-studied phenomena in linguistics such as control and raising, ellipsis, distributional restrictions on quantifiers, and countless others. This is likely due to the labor-intensive nature of collecting examples that exhibit informative grammatical phenomena and their acceptability judgments.

A related line of work uses acceptability judgments to evaluate what neural networks know about about grammatical phenomena in general. Corpora of sentences and their grammaticality are

Phenomenon	Relevant work
Anaphora/binding	Marvin and Linzen (2018), Futrell et al. (2018), Warstadt et al. (2019b)
Subject-verb agreement	Linzen et al. (2016), Futrell et al. (2018), Gulordava et al. (2019), Marvin and Linzen (2018), An et al. (2019), Warstadt et al. (2019b)
Negative polarity items	Marvin and Linzen (2018), Futrell et al. (2018), Jumelet and Hupkes (2018), Wilcox et al. (2019), Warstadt et al. (2019a)
Filler-gap dep./Islands	Wilcox et al. (2018), Warstadt et al. (2019b), Chowdhury and Zamparelli (2018, 2019) Chaves (to appear), Da Costa and Chaves (to appear)
Argument structure	Kann et al. (2019), Warstadt et al. (2019b), Chowdhury and Zamparelli (2019)


Table 2: Summary of related work organized by linguistic phenomena tested. All studies analyze neural networks using acceptability judgments on minimal pairs. Some studies appear multiple times.

collected for this purpose in a number of computational studies on grammaticality judgment (Heilman et al., 2014; Lau et al., 2017; Warstadt et al., 2019b). The most recent and comprehensive corpus is CoLA (Warstadt et al., 2019b), which contains around 10k sentences covering a wide variety of linguistic phenomena from 23 linguistic papers and textbooks. CoLA, which is included in the GLUE benchmark (Wang et al., 2018), has been used to track advances in the general grammatical knowledge of reusable sentence understanding models. Current models like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2019) can be trained to give acceptability judgments that agree with CoLA about as much as individual humans, if not more.

While CoLA can also be used to evaluate phenomenon-specific knowledge of models, this method is limited by the need to train a supervised classifier on CoLA data prior to evaluation. Warstadt and Bowman (2019) compare the CoLA performance of pretrained sentence understanding models: an LSTM, GPT (Radford et al., 2018), and BERT. They find that all models have good performance on sentences involving marked argument structure, and struggle with sentences with long-distance dependencies like those found in questions, though the Transformers have a noticeable advantage. However, evaluating supervised classifiers prevents making strong conclusions about the models themselves, since biases in the CoLA training data may affect the results. For instance, relatively strong performance on a phenomenon might be due to a model’s implicit knowledge or to frequent occurrence of similar examples in the training data. Evaluating LMs on minimal pairs evades this problem by eschewing

supervised training on acceptability judgments. It becomes possible to use the probability an LM assigns to a sentence as a proxy for acceptability because other factors impacting a sentences probability such as length and lexical content are controlled for.

### 3 Data

The  dataset<sup>1</sup> consists of 67 paradigms of 1000 sentence pairs. Each paradigm is annotated for the unique contrast it isolates and the broader category of phenomena it is part of. The data is automatically generated according to expert-crafted grammars, and our automatic labels are validated with crowd-sourced human judgments.

#### 3.1 Data generation procedure

To create minimal pairs exemplifying a wide array of linguistic contrasts, it is necessary to artificially generate all datasets. This ensures both that we have sufficient unacceptable examples, and that the data is fully controlled, allowing for repeated isolation of a single linguistic phenomenon in each paradigm (Ettinger et al., 2018). The data generation scripts use a basic template to create each paradigm, pulling from a vocabulary of over 3000 words annotated for morphological, syntactic, and semantic features needed to create grammatical and semantically felicitous sentences. Examples (2) and (3) show one such template for the ‘acceptable’ and ‘unacceptable’ sentences within a pair: the sole difference between them is the underlined critical word, which differs *only* in whether the anaphor agrees in number with its antecedent. Our

<sup>1</sup><https://github.com/alexwarstadt/blimp>

generation codebase is freely available.<sup>2</sup>

- (2) DP1 V1 refl\_match .  
The cats licked themselves .
- (3) DP1 V1 refl\_mismatch .  
The cats licked itself .

This generation procedure is not without limitations, and despite the very detailed vocabulary we use, implausible sentences are occasionally generated (e.g., ‘Sam ran around some glaciers’). In these cases, though, both the acceptable and unacceptable sentences will be equally implausible given world knowledge, so any difference in the probability assigned to them is still due to the grammatical contrast that they are designed to isolate.

### 3.2 Coverage

The paradigms that are covered by 🍌 represent well-established contrasts in English morphology, syntax, and semantics. Each individual paradigm is grouped into one of 12 phenomena, shown in Table 1. These phenomena are selected with the constraint that the phenomenon can be illustrated with minimal pairs of equal sentence length and that it is of a form that could be written as a template, like in (2) and (3). While this dataset has broad coverage, it is not exhaustive – it is not possible to include every grammatical phenomenon of English, and there is no agreed-upon set of core phenomena. However, we consider frequent inclusion of a phenomenon in a syntax/semantics textbook as an informal proxy for what linguists consider to be core phenomena. We survey several syntax textbooks (Sag et al., 2003; Adger, 2003; Sportiche et al., 2013), and find that nearly all of the phenomena in 🍌 are discussed in some source, and most of the topics that repeatedly appear in textbooks and can easily be represented with minimal pairs (e.g. agreement, argument selection, control/raising, wh-extraction/islands, binding) are represented in 🍌. Because the generation code is reusable, future experiments that need a paradigm not included in 🍌 will be able to generate their own datasets.

### 3.3 Comparison to Related Resources

With over 3000 words, 🍌 has by far the widest lexical variability of any related generated dataset.

<sup>2</sup>[https://github.com/alexwarstadt/data\\_generation](https://github.com/alexwarstadt/data_generation)

The vocabulary includes verbs with 11 different subcategorization frames, including verbs that select for prepositional phrases, infinitival verb phrases, and embedded clauses and questions. For instance Ettinger et al. (2018) and Marvin and Linzen (2018) each use a vocabulary of well under 200 items.

### 3.4 Data validation

To verify that the generated sentences represent a real contrast in acceptability, we conduct human validation via Amazon Mechanical Turk. Twenty separate validators rated five pairs from each of the 67 paradigms, for a total of 6700 judgments. We restricted validators to individuals currently located in the US who self-reported as native speakers of English. To assure that our validators were making a genuine effort on the task, each HIT included an attention check item and a hidden field question to catch bot-assisted humans. For each acceptable-unacceptable pair of sentences, 20 different individuals completed a forced-choice task that mirrors the task done by the LMs; the human-determined “acceptable” sentence was calculated via majority vote of annotators. By this metric, we estimate aggregate human agreement with our annotations to be 96.4% overall. As a threshold of inclusion in 🍌, the majority of validators needed to agree with 🍌 on at least 4/5 examples from each paradigm. Thus, all 67 paradigms in the public version of 🍌 passed this validation, and only two additional paradigms had to be rejected on this criterion. We also estimate *individual* human agreement to be 88.6% overall using the approximately 100 annotations from each paradigm.<sup>3</sup> Figure 3 reports these individual human results (alongside model results) as a conservative measure of human agreement.

## 4 Models & Methods


### 4.1 Models

**GPT-2** GPT-2 (Radford et al., 2019) is a large-scale language model using the Transformer architecture (Vaswani et al., 2017). We use the large version of GPT-2, which contains 24 layers and 345M parameters. The model is pre-trained on Radford et al.’s custom-built WebText dataset, which contains 40GB of text from various domains that is extracted from web pages and


<sup>3</sup>A small number had to be excluded due to ineligible annotators.





Model	Overall	Ana. Agr	Arg. Str	Binding	Ctrl. Rais.	D-N Agr	Ellipsis	Filler. Gap	Irregular	Island	NPI	Quantifiers	S-V Agr
5-gram	60.5	47.9	71.9	64.4	68.5	70.0	36.9	58.1	79.5	53.7	45.5	53.5	60.3
LSTM	68.9	91.7	73.2	73.5	67.0	85.4	67.6	72.5	89.1	42.9	51.7	64.5	80.1
TXL	68.7	94.1	69.5	74.7	71.5	83.0	77.2	64.9	78.2	45.8	55.2	69.3	76.0
GPT-2	80.1	99.6	78.3	80.1	80.5	93.3	86.6	79.0	84.1	63.1	78.9	71.3	89.0
Human	88.6	97.5	90.0	87.3	83.9	92.2	85.0	86.9	97.0	84.9	88.1	86.6	90.9

Table 3: Percentage accuracy of four baseline models and raw human performance on  using a forced-choice task. A random guessing baseline would give expected accuracy of 50%.

filtered by humans. To our best knowledge, the WebText corpus is not publicly available. Assuming approximately 5-6 bytes per word on average, we estimate WebText contains approximately 8B tokens. The testing code for GPT-2 has been integrated into `jiant`, a codebase for training and evaluating sentence understanding models (Wang et al., 2019b).<sup>4</sup>


**Transformer-XL** Transformer-XL (Dai et al., 2019) is another multi-layer Transformer-based neural language model. We test a pretrained Transformer-XL model with 18 layers of Transformer decoders and 16 attention heads for each layer. The model is trained on WikiText-103 (Merity et al., 2016), a corpus of 103M tokens from high-quality Wikipedia articles. Code for testing Transformer-XL on  is also implemented in `jiant`.

**LSTM** We include a long-short term memory (LSTMs, Hochreiter and Schmidhuber, 1997) language model in our experiments. Specifically, we test a pretrained LSTM language model from (Gulordava et al., 2019) on . The model is trained on a 90M token corpus extracted from English Wikipedia. For investigating the effect of training size on models’  performance, We retrain a series of LSTM models with the same hyperparameters and the following training sizes: 64M, 32M, 16M, 8M, 4M, 2M, 1M, 1/2M, 1/4M, and 1/8M tokens. For each size, we train the model on five different random samples drawing from the original training data, which has a size of 83M tokens. We release our LSTM evaluation code.<sup>5</sup>


**5-gram** We build a 5-gram LM on the English Gigaword corpus (Graff et al., 2003), which con-


sists of 3.07B tokens. To efficiently query  $n$ -grams we use an implementation<sup>6</sup> based on (Heafield et al., 2013), which is shown to speed up estimation (Heafield, 2011). We release our  $n$ -gram evaluation code.<sup>7</sup>


## 4.2 Evaluation

We mainly evaluate the models by measuring whether the LM assigns a higher probability to the grammatical sentence within the minimal pair. This method, used by Marvin and Linzen (2018), is only meaningful for comparing sentences of similar length and lexical content, as overall sentence probability tends to decrease as sentence length increases or word frequencies decrease (see Lau et al., 2017). However, as discussed in Section 3 we design every paradigm in  to be compatible with this method.

## 5 Results

We report the 12-category accuracy results for all models and human evaluation in Table 3. Fine-grained results for individual paradigms can be accessed in supplementary materials available alongside .

**Overall Results** An LM’s overall performance on  can be measured simply by measuring the proportion of correct predictions across the 67,000 minimal pairs from all paradigms. GPT-2 achieves the highest score and the  $n$ -gram the lowest. Transformer-XL and the LSTM LM perform in the middle, and at roughly the same level as each other. All models perform well below our estimated human performance (as described in Section 3.4).


The  $n$ -gram model’s poor overall performance confirms  is not solvable from co-occurrence in-


<sup>4</sup><https://github.com/nyu-ml/jiant/tree/blimp-and-npi/scripts/blimp>

<sup>5</sup><https://github.com/sheng-fu/colorlessgreenRNNs>

<sup>6</sup><https://github.com/kpu/kenlm>

<sup>7</sup>[https://github.com/anhad13/blimp\\_ngram](https://github.com/anhad13/blimp_ngram)

formation alone. Rather, success at  is driven by the more abstract (and less interpretable) features learned by neural networks. Furthermore, there are no categories in which the  $n$ -gram comes close to human performance.

Because we evaluate pretrained models that differ in architecture, and training data quantity/domain, we can only speculate about what drives these differences (though see Section 6.3 for a controlled ablation study). Nonetheless, the results seem to indicate that access to training data is the main driver of performance on  for the neural models we evaluate. On purely architectural grounds, the similar performance of Transformer-XL and the LSTM is surprising since Transformer-XL is the state of the art on several LM training sets. However, they are both trained  $100\pm 10$ M tokens of Wikipedia text. GPT-2's advantage seems to come from the fact that it is trained on roughly two orders of magnitude more data. While it is unclear whether LSTMs trained on larger datasets could rival GPT-2, such experiments are impractical due to the difficulty of scaling LSTMs to this size.

**Phenomenon-Specific Results** The results also reveal considerable variation in performance across grammatical phenomena. Models generally perform best and closest to human level on morphological phenomena. This includes anaphor agreement, determiner-noun agreement, and subject-verb agreement. In each of these domains, GPT-2's performance is within 2.1 percentage points of humans. The set of challenging phenomena is somewhat more diverse. Islands are the hardest phenomenon by a wide margin. Only GPT-2 performs noticeably above chance, but it remains 20 points below humans. Some semantic phenomena, specifically those involving NPIs and quantifiers, are also challenging overall. Finally, all models show relatively weak performance on the argument structure contrasts.

Based on these results, we conclude that current SotA LMs have robust knowledge of basic facts of English agreement. We do not wish to suggest, however, that LMs will come close to human performance for all agreement phenomena. In Section 6.1 we discuss evidence that increased dependency length and the presence of agreement attractors of the kind investigated by Linzen et al. (2016) and Gulordava et al. (2019) reduce performance on agreement phenomena.

The exceptionally poor performance on islands is hard to reconcile with Wilcox et al.'s (2018) conclusion that LSTMs have knowledge of some island constraints. In part, this difference may come down to differences in metrics. Wilcox et al. compare a set of four related sentences to obtain the *wh*-licensing interaction as a metric of how strongly the LM identifies a filler-gap dependency in a single syntactic position, and they consider an island constraint to have been learned if this value is close to zero. We instead compare LM probabilities of sentences with similar lexical content but with gaps in different syntactic positions. These metrics target different forms of grammatical knowledge, though both are desirable properties to find in an LM. We also note that this does not imply poor knowledge of filler-gap dependencies in general, with all neural models perform above well above chance. This suggests that, while these models are able to establish long-distance dependencies in general, they are comparatively worse at identifying the syntactic domains in which these dependencies are blocked.

The semantic phenomena that models struggle with are usually attributed in current theories to a presupposition failure or contradiction arising from semantic composition or pragmatic reasoning (e.g., Chierchia, 2013; Ward and Birner, 1995; Geurts and Nouwen, 2007). This suggests these models may struggle to learn these underlying semantic and pragmatic factors. Marvin and Linzen similarly find that LSTMs largely fail to recognize NPI licensing contrasts. (Warstadt et al., 2019a) find that BERT (which is similar in scale to GPT-2) has only moderate sensitivity to unlicensed NPIs in an unsupervised setting.

The relatively weak performance on argument structure is somewhat surprising, since arguments are usually (though by no means always) local to their heads. Argument structure is closely related to semantic event structure (see Marantz, 2013), which may be comparatively difficult for LMs to learn. This finding also contradicts Warstadt and Bowman's (2019) conclusion that argument structure is one of the strongest domains for neural models. However, Warstadt and Bowman study supervised models with training on CoLA, which includes a large proportion of sentences related to argument structure.

**Correlation of Model & Human Performance** We also examine to what extent the models' per-

5-gram	0.34	0.39	0.58	0.59	1
LSTM	0.49	0.63	0.9	1	0.59
TXL	0.48	0.68	1	0.9	0.58
GPT-2	0.54	1	0.68	0.63	0.39
human	1	0.54	0.48	0.49	0.34
	human	GPT-2	TXL	LSTM	5-gram

Table 4: Heatmap showing the correlation between models’ accuracies in each of the 67 paradigms.

formances are similar to each other, and how they are similar to human evaluation in terms of which phenomena are comparatively difficult. Figure 4 shows the Pearson correlation between four models and human evaluation on their accuracies in 67 paradigms. Compared to humans, GPT-2 has the highest correlation, closely followed by Transformer-XL and LSTM, though the correlation is only moderate. The  $n$ -gram’s performance correlates with humans relatively weakly. Transformer-XL and LSTM are very highly correlated at 0.9, possibly reflecting their similar training data. We also observe that neural models all correlate with each other more strongly than with humans or the  $n$ -gram model. This suggests that neural networks share some biases that are not entirely human-like.

**Shallow Predictors of Performance** We also ask what factors aside from linguistic phenomena make a minimal pair harder or easier for an LM to distinguish. In particular, are shallow features like sentence length or overall sentence likelihood predictors of whether the LM will have the right preference? The results are shown in Figure 1. While sentence length, perplexity and the probability of the good sentence all seems to predict model performance to a certain extent, the predictive power is not strong, especially for GPT-2, which is much less influenced by greater perplexity of the good sentence than the other models.

## 6 Additional Experiments

### 6.1 Long-Distance Dependencies

The presence of intervening material that lengthens an agreement dependency lowers accuracy on that sentence in both humans and LMs. We study how the presence or absence of this intervening material affects the ability of LMs to de-

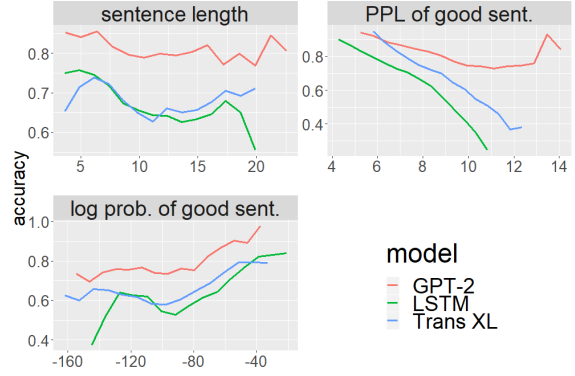


Figure 1: Models’ performance on as a function of sentence length, perplexity, and LM log probability of the good sentence.

tect mismatches in agreement in . First, we test for knowledge of determiner-noun agreement with and without an intervening adjective, as in Example (4). The results are plotted in Figure 2. The  $n$ -gram model is the most heavily impacted, performing on average 35 points worse. This is unsurprising, since the bigram consisting of a determiner and noun is far more likely to be observed than the trigram of determiner+noun+adjective. For the neural models, we find a weak but consistent effect, with all models performing on average between 5 and 3 points worse when there is an intervening adjective.

- (4) a. Ron saw that man/\*men.  
b. Ron saw that nice man/\*men.

Second, we test for sensitivity to mismatches in subject-verb agreement when an “attractor” noun of the opposite number intervenes. We compare attractors in relative clauses and as part of a relational noun (5). This follows experiments by (Linzen et al., 2016) and others. Again, we find an extremely large effect for the  $n$ -gram model, which performs over 50 points worse and performing well below chance when there is an attractor present, showing that the  $n$ -gram model is consistently fooled by the presence of the attractor. All of the neural models perform above chance with an attractor present, but GPT-2 and the LSTM perform 22 and 20 points worse when an attractor is present. Transformer-XL’s performance is harmed by only 5 points. Note however that GPT-2 still has the highest performance in both cases, and even outperforms humans in the relational noun case. Thus, we reproduce Linzen et al.’s (2016)

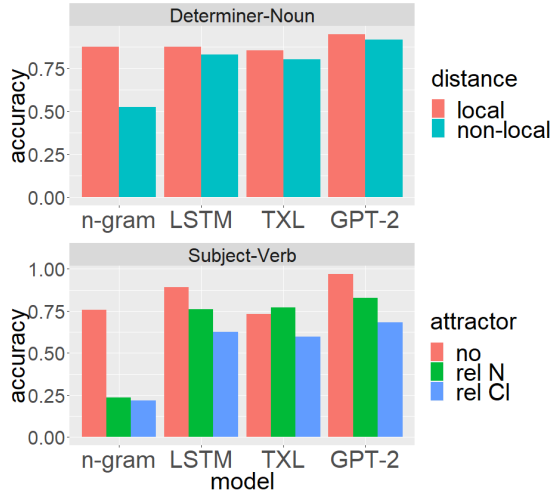


Figure 2: How models’ performance is affected by the locality of determiner-noun agreement (upper panel) and presence and type of agreement attractor (lower panel).

finding that attractors significantly reduce LSTM LMs’ sensitivity to mismatches in agreement, and find evidence that this holds true of Transformer LMs as well.

- (5) a. The sisters bake/\*bakes.
- b. The sisters who met Cheryl bake/\*bakes.
- c. The sisters of Cheryl bake/\*bakes.

## 6.2 Regular vs. Irregular Agreement

In the determiner-noun agreement and subject-verb agreement categories, we generate separate datasets for nouns with regular and irregular number marking, as in Example (6). All else being equal, only models with access to sub-word-level information should make any distinction between regular and irregular morphology.

- (6) a. Ron saw that nice kid/\*kids. (regular)
- b. Ron saw that nice man/\*men. (irregular)

Contrary to this prediction, the results in Figure 3 show that the sub-word-level models GPT-2 and Transformer-XL show little effect of irregular morphology: they perform less than 0.013 worse on irregulars than regulars. Given their high performance overall, this suggests they robustly encode number features without relying on segmental cues.<sup>8</sup>

<sup>8</sup>We also find that the LSTM LM, which has word-level tokens, performs on average 5.2 points worse on the irregular paradigms. However, this effect is not due to morphology, but

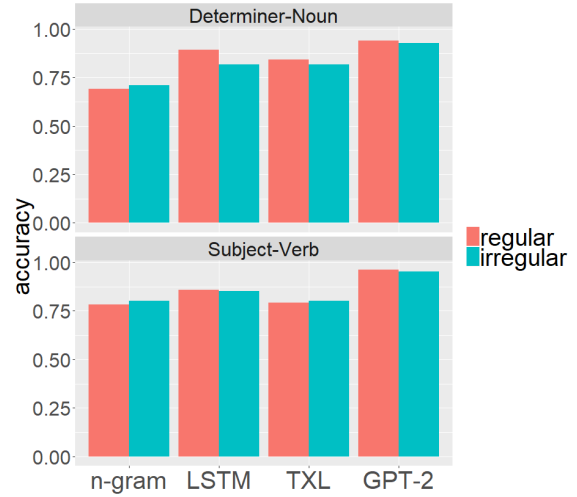


Figure 3: Models’ performance on regular vs. irregular agreement

## 6.3 Training size and 🧠 performance

🧠, with a wide range of linguistic phenomena, can also be used to track how a model’s knowledge of particular phenomena varies with the quantity of training data. We test this with the LSTM model and find that different phenomena in 🧠 have sharply different learning curves across different training sizes, as shown in Figure 4. Crucially, phenomena with similar results from the LSTM model trained on the full 83M tokens of training data may have very different learning curves. For example, the LSTM model performs well on both irregular forms and anaphor agreement, but the different learning curves suggest that more training data is required in the anaphor agreement case to achieve this same performance level. This is supported by the regression analysis showing that the best-of-fit line for anaphor agreement has the steepest slope (0.0623), followed by Determiner-Noun agreement (0.0426), Subject-Verb agreement (0.041), Irregular (0.039) and Ellipsis (0.0389). By contrast, Binding (0.016), Argument Structure (0.015), and Filler-Gap Dependency (0.0095) have shallower learning curves, showing a less strong effect of increases in training data size. The phenomena that showed the lowest performance overall, NPIs and Islands, also show the lowest effects of increases to training size, with slopes of 0.0078 and 0.0036, respectively. This indicates that, even given a substan-

to the higher proportion of out-of-vocabulary items among the irregular nouns.



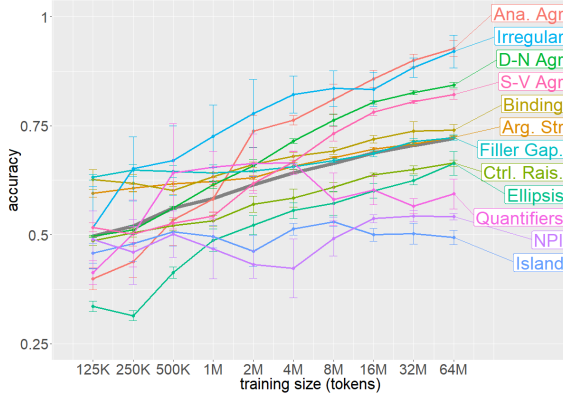


Figure 4: LSTM model performance as a function of training size and phenomena in 🗉. Error bars indicate standard deviations among five runs. The grey line shows the average across all phenomena.

tially larger amount training data, the LSTM is unlikely to achieve human-like performance on these phenomena – it simply fails to learn the necessary dependencies. It should be noted that these differences in learning curves show how 🗉 performance dissociates from perplexity, the standard measure of LM performance: while perplexity keeps decreasing as training size increases,<sup>9</sup> the performance in different 🗉 phenomena show very different learning curves.

#### 6.4 Alternate Evaluation Methods

There are several other techniques one can use to measure an LM’s “preference” between two minimally different sentences. So far, we have considered only the *full-sentence method*, advocated for by Marvin and Linzen (2018), which compares the LM likelihood of the full sentences. In a followup experiment, we use two “prefix methods” that evaluate the model’s preferences by comparing its prediction at a key point of divergence between the two sentences, each of which has appeared in prior work in this area. Subsets of 🗉 data—from the binding, determiner-noun agreement, and subject-verb agreement categories—are designed to be compatible with multiple methods, allowing us to conduct the first direct comparison. We find that all methods give broadly similar results when aggregating over a large set of paradigms, but some results diverge sharply for specific paradigms.

<sup>9</sup>Average ppl. on the Gulordava et al. (2019) test set: 595 at 0.125M, 212 at 1M, 92.8 at 8M, and 53 at 64M.

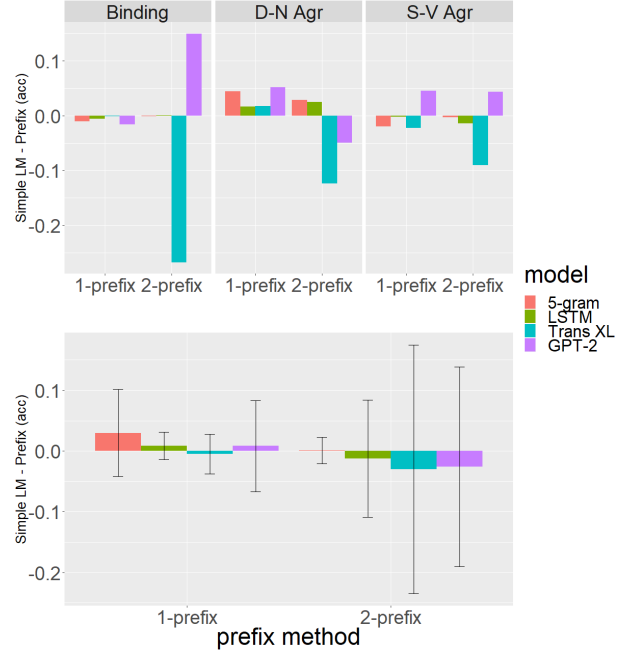


Figure 5: Comparison of models’ performance on the simple LM method and the 1- and 2-prefix methods. The upper panels show results from three phenomena that are compatible with both 1-prefix and 2-prefix methods. The lower panel shows the averages and standard deviations across all phenomena.

**One-prefix method** In the *one-prefix method*, used by Linzen et al. (2016), a pair of sentences share the same initial portion of a sentence, but differ in a critical word that make them differ in grammaticality (e.g., *The cat eats mice* vs. *The cat eat mice*). The model’s prediction is correct if it assigns a higher probability to the grammatical token given the shared prefix.

**Two-prefix method** In the *two-prefix method*, used by Wilcox et al. (2019), a pair of sentences have a different initial portion that differ in some critical way, but the grammaticality difference is only revealed when a shared critical word is included (e.g., *The cat eats mice* vs. *The cats eats mice*). For these paradigms, we evaluate whether the model assigns a higher probability to the grammatical sequence when it has only seen the initial portion plus the critical word. Note that the same pair of sentences cannot be compatible with both prefix methods, and that a pair may be compatible with the full-sentence method but neither prefix method.<sup>10</sup>

<sup>10</sup>This may occur when the two sentences are permutations of each other:

(7) a. It’s himself that John likes.

For both prefix methods, it is crucial that the grammaticality of the sentence is unambiguously predictable from the critical word, but not sooner. With simple LM probabilities, the probabilities of the rest of the word tokens in the sentence also affect the performance. For example, a model may decide that ‘The cat **ate** the mouse’ is more likely than ‘The cat **eaten** the mouse’ if the probability of a sequence like ‘**ate** the mouse’ is lower. Furthermore, for the ungrammatical case, it is unclear how a model assigns probabilities conditioned on an ungrammatical prefix, since ungrammatical sentences are largely absent from the training data. Using prefix probabilities allow us to exclude models’ use of this additional information and evaluate how the models perform when they have just enough information on grammaticality.

**Results** We find that models have generally comparable accuracies in prefix methods and the simple whole-sentence LM method. A deeper examination of the differences between these methods in each paradigm reveals some cases where a models’ performance fluctuates more between these methods. For example, Transformer-XL performs much worse at binding, determiner-noun agreement, and subject-verb agreement in the simple LM method, suggesting that the probabilities Transformer-XL assigns to the irrelevant part at the end of the sentence very often overturn the ‘judgment’ based on probability up to the critical word. On the other hand, GPT-2 benefits from reading the whole sentence for binding phenomena, as its performance is better in the simple LM method than in the prefix method. Overall, we observe that Transformer-XL and GPT-2 are more affected by evaluation methods than LSTM and  $n$ -gram when we compare the simple LM method and the *two-prefix method*. The results are summarized in Figure 5.

## 7 Discussion & Future Work

We have shown ways in which 🍪 can be used as tool to gain both high-level and fine-grained insight into the grammatical knowledge of language models. Like the GLUE benchmark (Wang et al., 2018), 🍪 assigns a single overall score to an LM which summarizes its general sensitivity to minimal pair contrasts. Thus, it can function as a

linguistically motivated benchmark for the evaluation of new language models that is complementary to more standard metrics like perplexity. 🍪 also provides a detailed breakdown of LM performance by linguistic phenomenon, which can be used to draw concrete conclusions about the kinds of grammatical knowledge acquired by a given model.

One question we leave unexplored is how well supervised acceptability classifiers built on top of pretrained models like BERT (Devlin et al., 2019) perform on 🍪. It would be possible to evaluate how well such classifiers generalize to unseen phenomena by training on a subset of paradigms in 🍪 and evaluating on the held-out sets, giving an idea of to what extent models are able to transfer knowledge in one domain to a similar one. Warstadt et al. (2019a) find that this method is potentially more revealing of implicit grammatical knowledge than purely unsupervised methods.

An important goal of linguistically-informed analysis of LMs is to better understand those empirical domains where current LMs appear to acquire some relevant knowledge, but still fall short of human performance. The results from 🍪 suggest that—in addition to relatively well-studied phenomena like filler-gap dependencies, NPIs, and binding—argument structure remains one area where there is much to uncover about what LMs learn. However, as language modeling techniques continue to improve, it will be useful to have large-scale tools like 🍪 to efficiently track changes in what these models do and do not know about grammar.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1850208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This project has also benefited from support to SB by Eric and Wendy Schmidt (made by recommendation of the Schmidt Futures program), by Samsung Research (under the project *Improving Deep Learning using Latent Structure*), by Intuit, Inc., and by NVIDIA Corporation (with the donation of a Titan V GPU).

b. \*It’s himself that likes John.

## References

- David Adger. 2003. *Core Syntax: A Minimalist Approach*. Oxford University Press Oxford.
- Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. 2019. Representation of constituents in neural language models: Coordination phrase as a case study. *arXiv preprint arXiv:1909.04625*.
- Rui P Chaves. to appear. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the third meeting of the Society for Computation in Linguistics (SCiL)*.
- Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.
- Gennaro Chierchia. 2013. *Logic in Grammar*. Oxford University Press.
- 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.
- Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un) grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212.
- Jillian K Da Costa and Rui P Chaves. to appear. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the third meeting of the Society for Computation in Linguistics (SCiL)*.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. *Transformer-XL: Attentive language models beyond a fixed-length context*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.
- Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801. Association for Computational Linguistics.
- Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.
- Bart Geurts and Rick Nouwen. 2007. 'At least' et al.: the semantics of scalar modifiers. *Language*, pages 533–559.
- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2019. Colorless green recurrent networks dream hierarchically. *Proceedings of the Society for Computation in Linguistics*, 2(1):363–364.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics.
- Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696.
- Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 174–180.

- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.
- Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of lstms to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231.
- Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. *Proceedings of the Society for Computation in Linguistics*, 2(1):287–297.
- Jey Han Lau, Alexander Clark, and Shalom Lapin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.
- 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.
- Alec Marantz. 2013. Verbal argument structure: Events and participants. *Lingua*, 130:152–168.
- 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.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. [Pointer sentinel mixture models](#). *CoRR*, abs/1609.07843.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *arXiv e-prints*.
- Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*, 2 edition. CSLI Publications.
- Dominique Sportiche, Hilda Koopman, and Edward Stabler. 2013. *An Introduction to Syntactic Analysis and Theory*. John Wiley & Sons.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.
- 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *33rd Conference on Neural Information Processing Systems*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP*



*Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Alex Wang, Ian F. Tenney, Yada Pruksachatkun, Katherin Yu, Jan Hula, Patrick Xia, Raghu Pappagari, Shuning Jin, R. Thomas McCoy, Roma Patel, Yinghui Huang, Jason Phang, Edouard Grave, Haokun Liu, Najoung Kim, Phu Mon Htut, Thibault F'evry, Berlin Chen, Nikita Nangia, Anhad Mohananey, Katharina Kann, Shikha Bordia, Nicolas Patry, David Benton, Ellie Pavlick, and Samuel R. Bowman. 2019b. *jiant* 1.2: A software toolkit for research on general-purpose text understanding models. <http://jiant.info/>.

Gregory Ward and Betty Birner. 1995. Definiteness and the English existential. *Language*, pages 722–742.

Alex Warstadt and Samuel R Bowman. 2019. Grammatical analysis of pretrained sentence encoders with acceptability judgments. *arXiv preprint arXiv:1901.03438*.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, et al. 2019a. Investigating BERT’s knowledge of language: Five analysis methods with NPIs. In *Proceedings of EMNLP*, pages 2870–2880.

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

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*, pages 3302–3312.