

How to Plant Trees 🌳 in LMs: Data and Architectural Effects on the Emergence of Syntactic Inductive Biases

Aaron Mueller
Johns Hopkins University
amueller@jhu.edu

Tal Linzen
New York University
linzen@nyu.edu

Abstract

Accurate syntactic representations are essential for robust generalization in natural language. Recent work has found that pre-training can teach language models to rely on hierarchical syntactic features—as opposed to incorrect linear features—when performing tasks after fine-tuning. We test what aspects of pre-training are important for endowing encoder-decoder Transformers with an inductive bias that favors hierarchical syntactic generalizations. We focus on architectural features (depth, width, and number of parameters), as well as the genre and size of the pre-training corpus, diagnosing inductive biases using two syntactic transformation tasks: question formation and passivization, both in English. We find that the number of parameters alone does not explain hierarchical generalization: model depth plays greater role than model width. We also find that pre-training on simpler language, such as child-directed speech, induces a hierarchical bias using an order-of-magnitude less data than pre-training on more typical datasets based on web text or Wikipedia; this suggests that in cognitively plausible language acquisition settings, neural language models may be more data-efficient than previously thought.

1 Introduction

Accurate syntactic representations are necessary for robust generalization to new natural language inputs and for the generation of correct outputs. Consider the problem of identifying the subject of “said” in the following sentence:

- (1) Can **you** repeat what the **senator** next to the **cats** **said**?

Typical language models (LMs), which receive linear sequences of words as input, could conceivably rely on a linear or positional feature that usually, but does not always, identifies the correct subject of a verb. An LM could learn, for example, that

the first noun in the sentence is always the subject. This heuristic works for many simple sentences, but fails in Ex. (1): here, the first noun is “**you**”, and so this heuristic would lead the LM to incorrectly interpret the sentence as meaning “Can you repeat what **you** said?” The LM could also learn that the subject of the verb is the noun closest to the verb in the linear order of the sentence, in which case it would interpret Ex. (1) as “Can you repeat what the **cats** said?” By contrast, an LM that represents the sentence as hierarchically structured will correctly identify **senator** as the subject of the embedded clause that contains the verb **said**. This example demonstrates that a preference for syntactic features over linear features is *required* for robust linguistic generalization.

The success of large-scale pre-training across fine-tuning tasks suggests that exposure to natural language may teach models to rely on appropriate syntactic features instead of heuristics (even though models still often rely on heuristics; [McCoy et al. 2019](#)). This hypothesis is supported by the finding that, given minimal pairs of grammatical and ungrammatical sentences, the probability distribution over sentences defined by LMs often favors the grammatical sentence ([Marvin and Linzen, 2018](#); [Hu et al., 2020](#)). A related line of work has shown that, through pre-training, LMs can under some circumstances acquire syntactic inductive biases which are then applied to fine-tuning tasks, whereas models which have not been pre-trained do not have such inductive biases ([Warstadt and Bowman 2020](#); [Warstadt et al. 2020b](#); [Lovering et al. 2021](#); [Mueller et al. 2022](#)).

When does pre-training endow LMs with a syntactic inductive bias? In this study, we address two specific sub-questions: (1) Which architectural features make a syntactic inductive bias more likely to emerge in a Transformer LM? (2) How is the inductive bias affected by the genre and size of the pre-training corpus? We investigate these

questions by evaluating a range of Transformer encoder-decoder models based on T5 (Raffel et al., 2020). We evaluate both existing publicly available models and models that we pre-train ourselves; we explore a variety of model widths (embedding and hidden dimension, feed-forward layer size) and depths (number of layers), and pre-train on corpora of varying genres and sizes. We then evaluate models’ inductive biases by observing their out-of-distribution generalization when fine-tuned on syntactic transformations tasks (§4). We find that **depth matters more than width for the acquisition of hierarchical biases** (§5), and that **pre-training on simpler language induces hierarchical biases using far less data** (§6 and §7). This last finding suggests that in language acquisition settings in which the training corpus more closely resembles the language that children are exposed to, Transformers may be more sample-efficient than previously thought.

Our code is available on GitHub.¹

2 Background and Motivation

Every finite training set is consistent with multiple generalizations. We use the term **inductive bias** to refer to the set of assumptions that a model relies on when generalizing to new data. Our usage includes any factor that leads the model to generalize in one way rather than another (Mitchell, 1980); this can include not only the model’s architecture, but also representations learned from prior or concurrent training on tasks that are related to the target task (Caruana, 1997), and in particular self-supervised pre-training (Lovering et al., 2021).

We can infer a model’s inductive bias by observing how it generalizes out of distribution after training on a dataset that is compatible with multiple generalizations. Applying this methodology, McCoy et al. (2018), McCoy et al. (2020), and Petty and Frank (2021) find that LSTM and Transformer encoder-decoder models trained from scratch (without pre-training) on syntactic transformations, such as converting a declarative sentence into a question (§3), do not generalize in a hierarchical manner. By contrast, Mueller et al. (2022) find that certain **pre-trained** encoder-decoder models—including T5 and BART (Lewis et al., 2020)—do generalize hierarchically after fine-tuning. Warstadt and Bowman (2020) and Warstadt et al. (2020b) re-

port similar results for the pre-trained masked LM RoBERTa (Liu et al., 2019), though in their study a robust syntactic inductive bias only emerged when the training corpus was much larger than a human might be exposed to.

Previous work on the effect of training corpus size and genre on syntactic generalization includes Huebner et al. (2021), who find that masked LMs show stronger syntactic abilities after training on a few million words of child-directed speech than a similar amount of Wikipedia or news text; they do not, however, explore whether similar abilities arise from training on a larger amount of Wikipedia text. Van Schijndel et al. (2019) report experimental results suggesting that scaling the training corpus or model size is unlikely to result in human-like syntactic abilities for LSTM LMs, but they only vary model width and only train on Wikipedia data. We fill the gap between these studies by investigating the influence of multiple component of the Transformer architecture and by training on corpora of varying genres and sizes.

Our work is related more broadly to the syntactic LM evaluation literature. In this style of work, evaluation is typically performed using minimal pairs, where a grammatical and ungrammatical sentence or completion are provided to a model, and the model is expected to assign a higher probability to the grammatical variant. Syntactic evaluations have found that LSTM- (Hochreiter and Schmidhuber, 1997) and Transformer-based (Vaswani et al., 2017) LMs are sensitive to grammatical number and gender in subject-verb agreement and reflexives (Hu et al., 2020; Marvin and Linzen, 2018; Goldberg, 2019; Lakretz et al., 2021; Gauthier et al., 2020). LMs are also sensitive to filler-gap dependencies (Wilcox et al., 2018) and, to a lesser extent, negative polarity items (Marvin and Linzen, 2018; Warstadt et al., 2020a). This holds across languages (Mueller et al., 2020; Ravfogel et al., 2018) and across grammatical/typological features (Ravfogel et al., 2019).

Overall, prior work has shown that pre-training can impart hierarchical inductive biases to LMs. The goal of this study is to examine which aspects of pre-training—specifically, architecture and training data—contribute to the emergence of this bias.

3 Syntactic Transformations

To evaluate the linguistic inductive biases of our models, we employ the *poverty of the stimulus*

¹<https://github.com/aaronmueller/emergent-syntax>

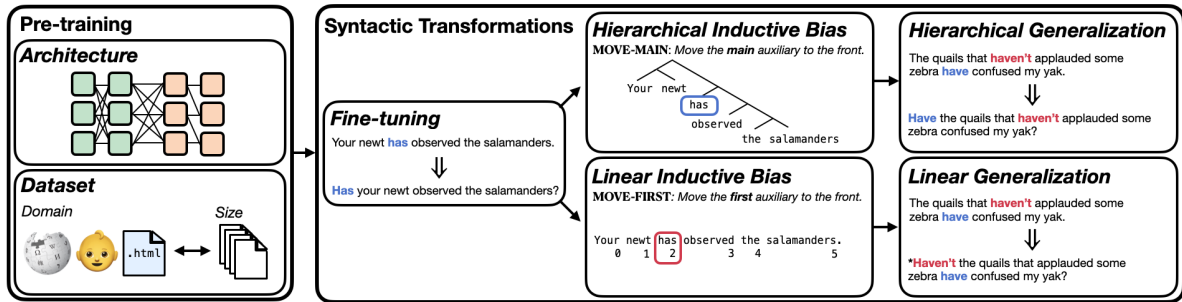


Figure 1: The syntactic transformations paradigm. A pre-trained model is fine-tuned on examples that are consistent with both syntactic (hierarchical) and positional/word order (linear) explanations. Then, it is evaluated on examples where only a model with a syntactic inductive bias will produce the correct output. We investigate which components of pre-training induce hierarchical inductive biases. Adapted from Warstadt et al. (2020b) and Mueller et al. (2022).

design (Wilson, 2006): we fine-tune a model on ambiguous data and then evaluate it on out-of-distribution examples where only the desired inductive bias will result in the correct outputs. Here, we use the **syntactic transformations** paradigm (Frank and Mathis, 2007) summarized in Figure 1, and observe whether models generalize according to hierarchical linguistic rules or according to surface heuristics based on word position or relative word ordering. We evaluate on English question formation and passivization, using the English datasets of Mueller et al. (2022) (themselves based on McCoy et al. 2020).

3.1 Question Formation

Here, the task is to transform a declarative sentence into a polar yes/no question by moving the auxiliary verb to the start of the sentence. The competing hypotheses are MOVE-FIRST and MOVE-MAIN (see Figure 1 for examples). We train the models on sentences that are consistent with both hypotheses, where the main auxiliary is always the linearly first auxiliary in the input sentence. Then, in the generalization examples, we append a relative clause (RC) to the subject, such that the main auxiliary is now the linearly *second* auxiliary in the input. A model that acquired MOVE-MAIN—that is, one that has a hierarchical inductive bias—will correctly identify the main auxiliary verb and move it to the front, meaning that it should still produce the correct output. A model that learned MOVE-FIRST will move the first auxiliary to the front, resulting in ungrammatical outputs (Figure 1).

3.2 Passivization

In this task, the goal is to transform an active sentence into a passive one. This requires various

insertions, deletions, reinfections, and movements, making this task a potentially more difficult one than question formation. Here, we evaluate the movement of the object to the subject position. The competing hypotheses here are MOVE-SECOND and MOVE-MAIN. We train the models on sentences where the object is always the linearly second noun in the sentence. Then, in the generalization examples, we append a prepositional phrase (PP) to the subject, such that the object is now the linearly *third* noun. If a model acquires the generalization MOVE-MAIN (consistent with a hierarchical inductive bias), it will detect the object and move it to the front, producing the correct output. If it acquires MOVE-SECOND, it will move the linearly second noun phrase even in the generalization examples (where, again, the correct noun to move is actually the linearly third one), and as such will output ungrammatical sequences. For example:

(2) Passivization

- a. *Training*: The raven observed the **newts** (near the yak). \Rightarrow The **newts** (near the yak) were observed by the raven.
- b. *Generalization*: The salamander behind the **ravens** applauded the **peacock**. \Rightarrow ?
- c. MOVE-MAIN (correct): The **peacock** was applauded by the salamander behind the **ravens**.
- d. MOVE-SECOND (incorrect): The **ravens** were applauded by the salamander.

3.3 Evaluation Metrics

For both syntactic transformations, we evaluate models' outputs using two metrics. The first is **sequence accuracy**, which measures the percentage of inputs for which the model's full output se-

quence is exactly correct. This is a strict metric that does not capture solely the syntactic phenomenon we investigate, but also penalizes the model for other errors, such as word substitution errors. We also report more targeted metrics for each of the tasks: **main auxiliary accuracy** for question formation, which measures how often the first word of the output sentence is the main auxiliary; and **object accuracy** for passivization, which measures how often the noun that gets moved to the start of the sentence is the object.

4 Overview of Experimental Paradigm

All of our experiments involve fine-tuning variants of T5, a Transformer encoder-decoder model pre-trained using a span denoising objective: contiguous token sequences are masked in the input sequence and then reconstructed in the output sequence. We either use the publicly available pre-trained “efficient” T5 models released by [Tay et al. \(2022\)](#),² or pre-train models ourselves using the transformers library ([Wolf et al., 2020](#)).

The syntactic transformation datasets we fine-tune on are the English datasets of [Mueller et al. \(2022\)](#), which consist of 100,000 training examples; 10,000 in-distribution test examples, which test whether the models have learned the task; and 10,000 out-of-distribution generalization examples, which reveal models’ inductive biases.

We adopt [Mueller et al.](#)’s hyperparameters (App. A). We fine-tune for 10 epochs (approximately 7500 training steps), and every 500 steps we save a checkpoint and evaluate it. Across models, accuracy on the in-distribution test set generally reaches 100% within 500 steps (the first checkpoint) and remains 100% throughout fine-tuning. Because in-distribution test set accuracy may not correlate with generalization accuracy, it is unclear which checkpoint would yield the best accuracy on the generalization set; we therefore report the mean generalization accuracy across all checkpoints.

5 Architectural Effects

Which architectural features contribute to hierarchical generalization? Given that language is struc-

²The term “efficient” here contrasts the models of [Tay et al. \(2022\)](#) with the original T5 models of [Raffel et al. \(2020\)](#), which are Pareto-inefficient with respect to downstream performance and number of parameters. The “efficient” models achieve similar performance across tasks using fewer parameters by using a deeper (more layers) and narrower (smaller hidden size/feed-forward size) architecture.

	Tiny	Mini	Small	Base
# Parameters	16M	31M	60M	220M
# Layers (NL)	4	4	6	12
Feedforward layer dimension (FF)	1024	1536	2048	3072
Embedding and hidden dimension (DM)	256	384	512	768
Key/value projection matrix dim. (KV)	32	32	32	64
# Heads per layer (NH)	4	8	8	12

Table 1: Architectural details for T5 variants from [Tay et al. \(2022\)](#). # Layers is the number of layers in each of the encoder and the decoder (multiply this number by 2 to obtain the total number of layers in the model).

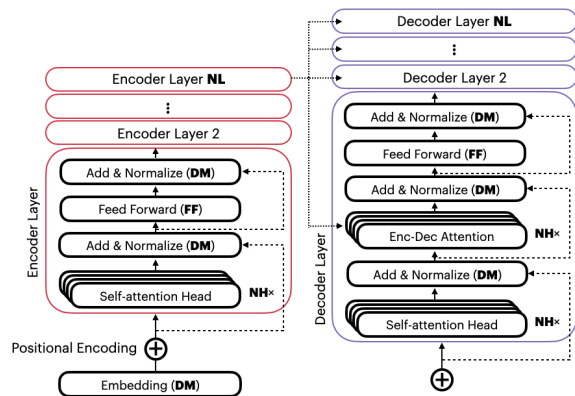


Figure 2: The Transformer encoder-decoder architecture, annotated with the architectural hyperparameters we vary.

ured hierarchically, we hypothesize that model depth (number of layers) will be the most important component, as deeper structure could more easily allow for representations of deeper hierarchical structures (e.g., more complex syntax trees), with recursive syntactic operations applied successively across layers ([Murty et al., 2023](#)).

5.1 Models

We fine-tune pre-trained models from [Tay et al. \(2022\)](#), available on HuggingFace. We train two sets of models. The first set is google/t5-efficient- $\{tiny, mini, small, base\}$; see Table 1 for the hyperparameters of these models, and Figure 2 for a diagram of the Transformer architecture that illustrates these hyperparameters. Note that multiple hyperparameter values change at the same time when moving from, e.g., T5_{small} to T5_{base}.

The second set of models we use from [Tay et al. \(2022\)](#) were derived from T5_{base} by changing exactly one hyperparameter value. For these more controlled variants, we adopt [Tay et al.](#)’s nomenclature, which is based on the particular hyperparameter that is being changed, and its new value;

for example, $T5_{\text{base}}\text{-DM512}$ (which we abbreviate here to DM512) is identical to $T5_{\text{base}}$, except the embedding/hidden dimension (DM) is reduced from 768 to 512. All of these models are trained on approximately 34B words from the Colossal Cleaned Common Crawl (C4) web text corpus.

5.2 Depth, not Scale, Predicts Syntactic Bias

We start by asking whether scale alone can explain hierarchical generalization: Is there a monotonic relationship between the number of parameters and in generalization accuracy? We find that the answer is no (Figure 3). For question formation, the Spearman rank-order correlation between the number of parameters and accuracy is 0.51 (sequence) and 0.58 (main auxiliary); for passivization, 0.75 (sequence) and 0.43 (object). While these are significant correlations ($p < .05$, except for object accuracy), if syntactic bias were predicted by scale alone, we would expect these to be close to 1. Thus, **number of parameters alone is not sufficient to explain the acquisition of a hierarchical bias**. This suggests that certain architectural components, which may be correlated with scale, are more important than others.

Indeed, we find that increasing model depth has a much stronger impact on accuracy than scaling the model up by increasing the value of other architectural hyperparameters (Figure 4): in a least squares linear regression where the dependent variable is sequence accuracy and independent variable is number of parameters (normalized to the same range as the accuracy values), the slope of the fitted line is 0.70 when varying over number of layers, but only 0.13 for embedding/hidden size, and 0.25 for feed-forward layer width. In particular, the wide and shallow NL4 has more parameters than the narrow and deep DM256, but achieves similar performance as DM256 on question formation and significantly worse performance on passivization (as a reminder, NL4 is $T5_{\text{base}}$ with 4 encoder layers and 4 decoder layers, and DM256 is $T5_{\text{base}}$ with embedding/hidden size 256). This suggests that **when scaling the architecture, model depth is more important than other components for enabling hierarchical generalization**.

Is encoder depth or decoder depth more important for hierarchical generalization, or is total depth alone responsible for the patterns we find? We investigate this in App. B, with mixed results: for passivization, reducing the depth of either compo-

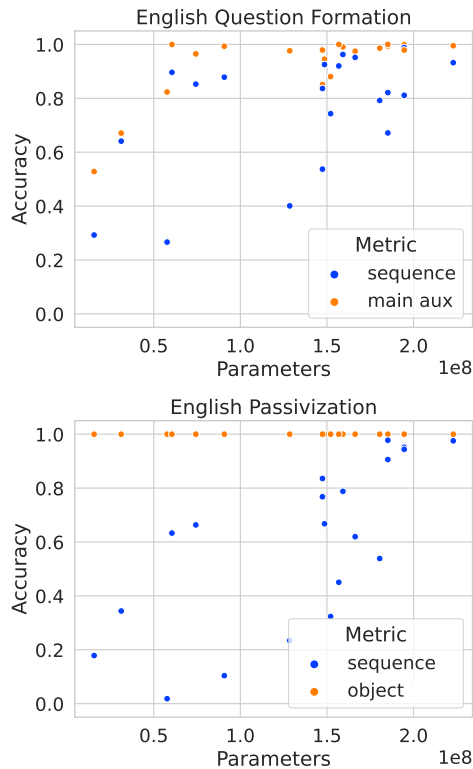


Figure 3: Generalization accuracies on question formation (top) and passivization (bottom) using $T5_{\{\text{tiny,mini,small,base}\}}$, as well as variants of $T5_{\text{base}}$ where we vary the number of layers, number of attention heads per layer, embedding/hidden dimension, feed-forward width, or key-value projection dimension. There is a positive correlation between the number of parameters and accuracy, but the trend is not monotonic. Certain architectural features may therefore play a more important role than others.

nent leads to similar drops in generalization accuracy, but for question formation, decoder depth has a greater effect than encoder depth.

5.3 Syntactic Bias Correlates with Downstream Performance

How well does syntactic generalization accuracy correlate with performance on other tasks? We address this question by correlating main auxiliary accuracy with validation perplexity, question answering accuracy on SQuAD (Rajpurkar et al., 2016), and scores on the SuperGLUE collection of natural language understanding tasks (Wang et al., 2019), all provided by Tay et al. (2022). We do not report correlations with passivization accuracy, as most models achieve 100% accuracy on this task, which leaves little explainable variance.

We obtain Spearman correlations of 0.57 ($p < .1$) for SuperGLUE, 0.34 ($p > .1$) for SQuAD, and 0.67

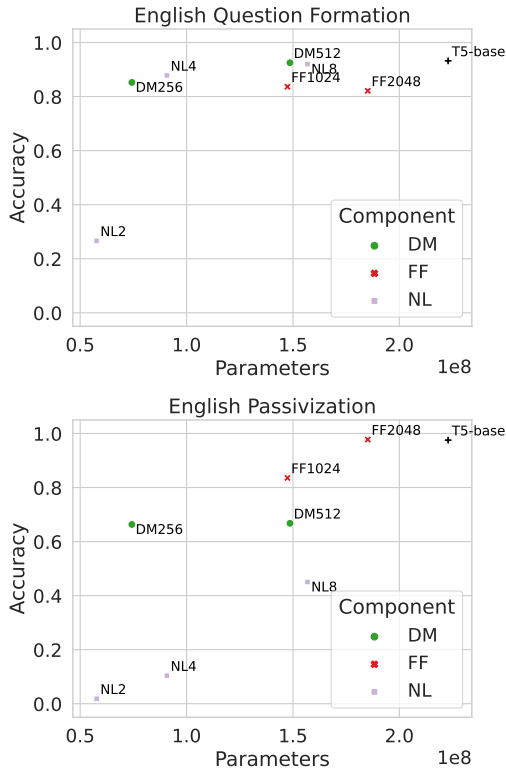


Figure 4: Generalization sequence accuracies on question formation (top) and passivization (bottom) using architecturally modified versions of $T5_{base}$. Decreasing model depth (NL) results in the greatest drop in performance, suggesting that model depth is more important for learning syntax than other components. See App. C for results when varying the number of attention heads or key/value projection dimension.

($p < .05$) for negative validation perplexity. In other words, the correlation is weak but significant with average SuperGLUE accuracy (Tay et al. do not report accuracy for individual SuperGLUE tasks); not significant with question answering; and relatively strong and significant with language modeling performance more broadly. We note that since the number of models is relatively modest, correlations need to be quite strong to reach the statistical significance threshold.

These correlations do not indicate that syntactic abilities are *causally* implicated in the models’ improved performance on other tasks, but they do show that **the emergence of syntactic abilities often co-occurs with better language modeling performance and downstream performance**. Future work could employ causal analysis methods to better understand how the emergence of syntactic preferences affects (or does not affect) performance across NLP tasks.

6 Corpus Genre

Large LMs are typically pre-trained on web text and/or Wikipedia data—genres that are distinct from the type of language that humans are exposed to during childhood. Could the domain of pre-training corpora explain why LMs require much more data than humans to reach similar syntactic abilities (Warstadt et al., 2020b)? Huebner et al. (2021) report experiments that support this hypothesis: they find that the RoBERTa masked LM achieves higher accuracies on linguistic acceptability judgment benchmarks when it is pre-trained on child-directed speech as opposed to a similar amount of Wikipedia data. In this section, we investigate whether this applies to our paradigm by pre-training encoder-decoder models on child-directed speech and a similar amount of text drawn from the English Wikipedia.

6.1 Models

We train models based on the T5 architecture and objective (see §4) on the English portion of CHILDES (MacWhinney, 2000), a 5M-word child-directed speech corpus, and on an English Wikipedia corpus from Huebner et al. (2021), which consists of a similar number of sentences as CHILDES. As Wikipedia sentences are longer, the total number of words in the Wikipedia training set we use here is approximately 10M.

We train models with eight hyperparameter configurations on each dataset (Table 2): we either vary the number of layers ($NL \in \{2, 4, 8, 16\}$), keeping other hyperparameters, such as embedding/hidden dimension and number of heads, constant; or we keep the number of layers at 8 and vary other hyperparameters. While we only pre-train each configuration once, we fine-tune each configuration five times, with a different random seed each time.

Following Huebner et al., we modify the training hyperparameters to better suit the smaller and simpler child-directed speech corpus: we reduce the maximum sequence length to 128 and train a SentencePiece tokenizer (Kudo and Richardson, 2018) with a reduced vocabulary size of $2^{13} = 8192$; this is motivated by children’s vocabulary size of approximately 5,000–6,000 lemmas at age 6 (Biemiller, 2003). For the Wikipedia corpus, we train SentencePiece tokenizers using vocab sizes $\in \{8192, 32768\}$ and take the best-performing

Model	Parameters	NL	FF	DM	KV	NH	Data	
							CHILDES	Wikipedia
Tiny	23M	8	1024	256	32	4	0.62 (.06)	0.07 (.02)
Mini	50M	8	1536	384	32	8	0.68 (.07)	0.35 (.08)
Small	75M	8	2048	512	32	8	0.73 (.04)	0.46 (.10)
Base	157M	8	3072	768	64	12	0.61 (.07)	0.45 (.09)
Large	268M	8	4096	1024	64	16	0.57 (.09)	0.26 (.09)
Small	31M	2	2048	512	32	8	0.49 (.04)	0.08 (.01)
Small	46M	4	2048	512	32	8	0.58 (.05)	0.35 (.08)
Small	75M	8	2048	512	32	8	0.73 (.04)	0.46 (.10)
Small	134M	16	2048	512	32	8	0.70 (.06)	0.48 (.08)

Table 2: Main auxiliary accuracies averaged across 5 fine-tuning random seeds (standard deviation across seeds in parentheses) on the question formation generalization dataset for various encoder-decoder models pre-trained from scratch on 5M words of transcribed child-directed speech or 10M words of Wikipedia text.

model for each hyperparameter configuration,^{3,4} as it is not clear *a priori* whether a smaller vocabulary would be beneficial for Wikipedia’s more complex and diverse language. We use sequence packing, where we concatenate multiple sentences from the corpus into a single example such that the total length of each training example is approximately equal to the maximum sequence length.

When pre-training on child-directed speech, we checkpoint every 10K training steps and find that the best performance on our syntactic transformations tasks is achieved at 130K steps. We train on the Wikipedia corpus for the same number of steps.

6.2 Results

We find that **pre-training on child-directed speech generally results in a greater ability to detect the main verb, as compared to pre-training on Wikipedia** (Table 2). This holds across model sizes and across model depths. The CHILDES-pre-trained 8-layer variant of T5_{small} performs best. When fixing NL at 8 and varying other components according to each model size’s default settings (as in Table 1), we find that T5_{small} performs best. In the following experiment, we therefore focus on T5_{small} modified to have 8 encoder layers and 8 decoder layers.

7 How Much Data Leads to the Emergence of a Syntactic Bias?

The next experiment we report has two goals. First, we aim to replicate the finding that sim-

³32768 is the vocab size for T5 (Raffel et al., 2020).

⁴The best vocab size varied depending on model size and corpus size. Vocab size 8192 tends to work better for smaller corpora and smaller models on average, and 32768 tends to work better for larger corpora and larger models.

pler language gives rise to a stronger syntactic bias. Second, we expand the range of corpus sizes for the genres where larger corpora are available; our goal is to determine how much data is necessary to induce a hierarchical bias from each genre. In addition to child-directed speech and English Wikipedia, which we included in the previous experiment, we also pre-train models on the Colossal Cleaned Common Crawl (C4) web text corpus (Raffel et al., 2020) and on Simple Wikipedia, which contains text from the same domain as English Wikipedia, but with a more limited vocabulary and simpler sentence structures.

7.1 Method

We collect English Wikipedia data using Wikidumps.⁵ We use the witokit library⁶ to preprocess the data. We pre-train on {1M, 10M, 100M, 1B} words of English Wikipedia data, where words are counted before being divided into subwords by the tokenizer. Our {1M, 10M}-word data is from Huebner et al. (2021); our {100M, 1B}-word data is a concatenation of their 10M-word dataset with the Wikidump data that we download and preprocess. For C4, we randomly shuffle the HuggingFace version of the dataset⁷ and sample individual examples until we have reached 1B words. We then create {1M, 10M, 100M, 1B}-word datasets by uniformly subsampling the data, ensuring that smaller datasets are subsamples of larger datasets. For CHILDES, we only have access to 5M words, so we pre-train on {1M, 5M} words, where the 1M-word dataset is a uniform subsample of the 5M-word dataset.

We also download Simple Wikipedia Wikidumps,⁸ and follow the same preprocessing pipeline we used for the English Wikipedia. Since we only have access to approximately 300M words of Simple Wikipedia, we only pre-train on {1M, 10M, 100M} words, where smaller datasets are uniform subsamples of larger datasets.

For all genres and sizes, we use the best-performing architecture from §6 (T5_{small} with 8 encoder layers and 8 decoder layers), as well as the best training hyperparameters from that experiment. We tune over vocabulary size for each corpus style and size. See App. A for details.

⁵<https://dumps.wikimedia.org/enwiki/>

⁶<https://github.com/akb89/witokit>

⁷<https://huggingface.co/datasets/c4>

⁸<https://dumps.wikimedia.org/simplewiki/>

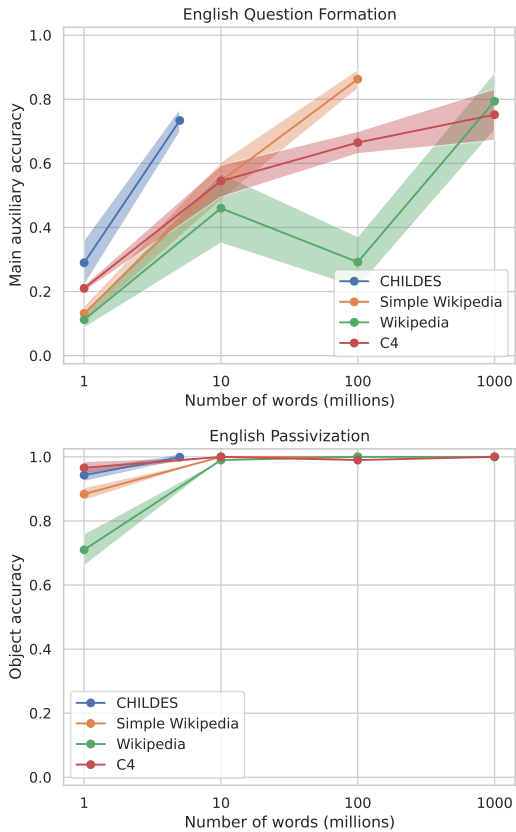


Figure 5: Generalization accuracies for question formation (top) and passivization (bottom) when pre-training a small T5-like model (8 encoder layers and 8 decoder layers) on corpora of various sizes and domains. Simpler language induces syntactic generalization with less data: CHILDES outperforms other datasets, and Simple Wikipedia outperforms Wikipedia. Accuracies (points) and standard deviations (shaded regions) are measured across 5 random seeds of fine-tuning.

7.2 Results

Replicating and extending our results from §6, we find that **pre-training on simpler language induces hierarchical generalization using less data** (Figure 5). For question formation, transcribed child-directed speech, the simplest language style we use, induces hierarchical generalization in well over 50% of question formation generalization examples using just 5M words. For Simple Wikipedia and C4, 100M words are required to reach this accuracy level; for Wikipedia, 1B words. Models pre-trained on Simple Wikipedia generalize in a much more syntax-sensitive manner than models pre-trained on a similar amount of Wikipedia data.

For passivization, generalization accuracies are generally much higher, though the qualitative trends we observe for question formation still hold: child-directed speech induces hierarchical gener-

Dataset	# Words	Main auxiliary accuracy	Object accuracy
Wikipedia	100M	0.27 (.08)	0.98 (.01)
Wikipedia + CHILDES	105M	0.20 (.05)	0.99 (.01)
C4	100M	0.67 (.04)	1.00 (.00)
C4 + CHILDES	105M	0.60 (.05)	1.00 (.00)
CHILDES	5M	0.73 (.04)	0.99 (.00)

Table 3: Generalization accuracy for question formation and passivization using the 100M-word versions of Wikipedia and C4, before and after concatenating CHILDES. Accuracy is averaged over five fine-tuning seeds (standard deviation over seeds in parentheses).

alization using less data, and Simple Wikipedia induces hierarchical generalization using less data than Wikipedia.

Could we narrow the gap between Wikipedia/C4 and CHILDES by simply concatenating CHILDES to these datasets? The answer appears to be no: performance does not significantly change when concatenating CHILDES to Wikipedia, nor when concatenating CHILDES to C4 (Table 3). Perhaps the style of the different datasets is too dissimilar for the model to form consistent generalizations when exposed to both distributions simultaneously. It could be more beneficial to run a two-phase pre-training procedure, where we expose the model to the simpler CHILDES dataset first, and then expose it to Wikipedia or C4 only after it has acquired the hierarchical inductive bias. We discuss this hypothesis in more detail in §8.

8 Discussion

Why does depth facilitate the emergence of a syntactic bias? Our first set of experiments suggests that depth is the most important architectural factor contributing to hierarchical generalization in Transformers. This finding is consistent with the suggestion of [Tay et al. \(2022\)](#), who advocate for deeper and narrower architectures for the best performance across NLP tasks. Why are deeper models better in practice for many tasks and linguistic evaluations, when in theory an arbitrarily wide model can approximate any function with only two layers ([Hornik et al., 1989](#))?

One natural hypothesis is that Transformers generalize hierarchically on the basis of tree-structured representations organized across layers, such that higher layers represent larger constituents, and recursive syntactic operations are applied across successive layers; such a strategy arises more naturally

in a deeper model. In recent work, [Murty et al. \(2023\)](#) find evidence that the internal organization of Transformer representations across layers becomes more tree-like over the course of training on some tasks, and that this property predicts the model’s compositional generalization. While they fail to find a correlation between model depth and the degree to which representations are tree-shaped, this may be because they train relatively small models from scratch on synthetic datasets. In future work, methods such as those of [Murty et al. \(2023\)](#) may be used to measure the tree-likeness of Transformers’ representations throughout pre-training on natural language, and the degree to which the tree-likeness of the pre-trained model correlates with its syntactic inductive bias for fine-tuning.

Why does simpler language teach syntax more effectively? We find that pre-training on simpler language, such as child-directed speech or Simple Wikipedia, enables hierarchical generalization from far less pre-training data than more complex language. Our findings from encoder-decoder models are consistent with previous findings from encoder-only masked LMs ([Huebner et al., 2021](#)), and with work on language understanding from speech ([Gelderloos et al., 2020](#)). The advantage of child-directed speech may be attributable to reduced lexical complexity, reduced syntactic complexity, or both ([Soderstrom, 2007](#)). Lower lexical complexity—in this case, fewer word types—may make it possible to learn the distribution of, say, parts of speech from a smaller corpus, as the same words would recur more often in different contexts. Lower syntactic complexity could result in a higher proportion of short sentences with unambiguous syntactic structure, which could help bootstrap syntactic learning. These two features are correlated in natural child-directed speech, but could be disentangled in future work by independently manipulating the lexical and syntactic distributions.

Simpler language can be leveraged for more efficient pre-training. Our experiments show that not all pre-training data is created equal, and motivate further research on data curation for pre-training, and in particular on curriculum learning ([Bengio et al., 2009](#)). We conjecture that robust syntactic inductive biases will play a role not only in fine-tuning but also in pre-training, making it possible for models to use additional pre-training sentences more efficiently. This motivates a two-phase

“starting small” approach ([Elman, 1993](#)), where the model is first exposed to a model to child-directed speech until syntactic inductive biases emerge, and then pre-training on a larger corpus proceeds as usual afterwards. This approach is related to, but distinct from, the single-phase simple-to-complex approach, where a pre-training dataset is sorted from the simplest inputs to the most complex and then presented to a model in order. The single-phase approach has demonstrated mixed results ([Campos, 2021](#); [Surkov et al., 2022](#)), but to our knowledge, a syntax-focused two-phase approach has not yet been attempted.

Transformers may be more data-efficient than previously thought. Our findings about the amount of pre-training data required for the acquisition of syntactic biases also have implications for cognitive modeling research. Humans learn language from far fewer words than contemporary LMs, and at the same time generalize their linguistic knowledge to new settings more robustly; conversely, standard NLP evaluations, which do not take the pre-training corpus into consideration, implicitly reward architectures that learn well from vast amounts of data, raising the concern that those architectures are suboptimal for cognitive modeling ([Linzen, 2020](#)). Our evaluation setup and empirical results go some way towards addressing these concerns: we show that pre-training on a developmentally plausible amount of data can induce human-like inductive biases that improve out-of-distribution generalization. This suggests that Transformers, when trained in cognitively relevant regimes, may serve as fruitful models of human language acquisition and processing (see also [Hosseini et al. 2022](#)).

9 Conclusions

We have analyzed the architectural and data features that contribute to the acquisition of syntactic inductive biases during the pre-training of encoder-decoder Transformers. We find that model depth matters more for hierarchical generalization than other model components (§5); that models more quickly learn that language is hierarchical given simpler language (§6); and that it takes orders-of-magnitude more data to induce hierarchical inductive biases when pre-training on genres such as Wikipedia or web text, compared to simpler data such as child-directed speech (§7).

Acknowledgements

We thank the authors of [Tay et al. \(2022\)](#) for facilitating academic research on language model scaling by releasing a large range of model checkpoints. We also thank Alexandra DeLucia, Nathaniel Weir, and Daniel Khashabi for their thoughtful feedback on earlier versions of this paper. This material is based upon work supported by the National Science Foundation (NSF) under Grant No. BCS-2114505. Aaron Mueller was supported by a National Science Foundation Graduate Research Fellowship (Grant #1746891). This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

Limitations

Our analyses are based on models with T5-like architectures and span denoising training objectives. Thus, our findings may not generalize to other types of encoder-decoder models (e.g., BART), nor encoder-only and decoder-only models. We believe this is unlikely, given that similar findings have been shown for models with architectures and objectives that differ significantly from T5's ([Huebner et al., 2021](#); [Warstadt and Bowman, 2020](#)). Nonetheless, it cannot be ruled out.

Our analyses are also based entirely in English, and only leverage two syntactic transformations. It is possible that our findings will not generalize to other languages, given that certain grammatical features (e.g., more extensive case marking) induce more syntax-sensitive behavior given a similar amount of training data across languages ([Mueller et al., 2020](#); [Ravfogel et al., 2019](#)); thus, perhaps less Wikipedia or C4 data is needed in these languages for models to acquire hierarchical preferences. It is also possible that, within a language, a model could adopt a hierarchical inductive bias for one type of transformation, but not another—especially if one transformation is much more frequent than the other. Indeed, the frequency of particular words positively correlates with syntactic evaluation accuracies ([Wei et al., 2021](#); [Newman et al., 2021](#)), and it would be reasonable to expect a similar trend for the frequency of syntactic transformations. Thus, future work should investigate more transformations in more languages to ensure that these findings are consistent.

References

- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, page 41–48, New York, NY, USA. Association for Computing Machinery.
- Andrew Biemiller. 2003. Vocabulary: Needed if more children are to read well. *Reading Psychology*, 24(3-4):323–335.
- Daniel Campos. 2021. [Curriculum learning for language modeling](#). *Computing Research Repository*, arXiv:2108.02170.
- Rich Caruana. 1997. [Multitask learning](#). *Machine Learning*, 28(1):41–75.
- Jeffrey L. Elman. 1993. [Learning and development in neural networks: The importance of starting small](#). *Cognition*, 48(1):71–99.
- Robert Frank and Donald Mathis. 2007. [Transformational networks](#). In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*. Cognitive Science Society.
- Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. [SyntaxGym: An online platform for targeted evaluation of language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.
- Lieke Gelderloos, Grzegorz Chrupała, and Afra Alishahi. 2020. [Learning to understand child-directed and adult-directed speech](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1–6, Online. Association for Computational Linguistics.
- Yoav Goldberg. 2019. [Assessing BERT's syntactic abilities](#). *Computing Research Repository*, arXiv:1901.05287.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. [Multilayer feedforward networks are universal approximators](#). *Neural Networks*, 2(5):359–366.
- Eghbal A. Hosseini, Martin Schrimpf, Yian Zhang, Samuel Bowman, Noga Zaslavsky, and Evelina Fedorenko. 2022. [Artificial neural network language models align neurally and behaviorally with humans even after a developmentally realistic amount of training](#). *bioRxiv*.
- Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. [A systematic assessment of syntactic generalization in neural language models](#). In *Proceedings of the 58th Annual Meeting of*

- the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.
- Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. [BabyBERTa: Learning more grammar with small-scale child-directed language](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021. [Mechanisms for handling nested dependencies in neural-network language models and humans](#). *Cognition*, 213:104699. Special Issue in Honour of Jacques Mehler, Cognition’s founding editor.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Tal Linzen. 2020. [How can we accelerate progress towards human-like linguistic generalization?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *Computing Research Repository*, arXiv:1907.11692.
- Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. [Predicting inductive biases of pre-trained models](#). In *International Conference on learning representations*.
- Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.
- Rebecca Marvin and Tal Linzen. 2018. [Targeted syntactic evaluation of language models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.
- R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. [Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks](#). In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*, pages 2096–2101, Madison, Wisconsin. Cognitive Science Society.
- R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. [Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks](#). *Transactions of the Association for Computational Linguistics*, 8:125–140.
- 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. Association for Computational Linguistics.
- Tom M Mitchell. 1980. *The need for biases in learning generalizations (Rutgers Computer Science Tech. Rept. CBM-TR-117)*. Rutgers University.
- Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. [Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.
- Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. [Cross-linguistic syntactic evaluation of word prediction models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.
- Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2023. [Characterizing intrinsic compositionality in transformers with tree projections](#). In *International Conference on Learning Representations*.
- Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. [Refining targeted syntactic evaluation of language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.
- Jackson Petty and Robert Frank. 2021. [Transformers generalize linearly](#). *Computing Research Repository*, arXiv:2109.12036.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.

- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. [Studying the inductive biases of RNNs with synthetic variations of natural languages](#). 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 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. [Can LSTM learn to capture agreement? the case of Basque](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.
- Melanie Soderstrom. 2007. Beyond babytalk: Re-evaluating the nature and content of speech input to preverbal infants. *Developmental Review*, 27(4):501–532.
- Maxim Surkov, Vladislav Mosin, and Ivan Yamshchikov. 2022. [Do data-based curricula work?](#) In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 119–128, Dublin, Ireland. Association for Computational Linguistics.
- Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022. [Scale efficiently: Insights from pretraining and finetuning transformers](#). In *International Conference on Learning Representations*.
- 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 5831–5837, Hong Kong, China. Association for Computational Linguistics.
- 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*, volume 30. Curran Associates, Inc.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [SuperGLUE: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Alex Warstadt and Samuel R. Bowman. 2020. [Can neural networks acquire a structural bias from raw linguistic data?](#) In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, Online. Cognitive Science Society.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.
- Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. [Learning which features matter: RoBERTa acquires a preference for linguistic generalizations \(eventually\)](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.
- Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. [Frequency effects on syntactic rule learning in transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. [What do RNN language models learn about filler-gap dependencies?](#) In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.
- Colin Wilson. 2006. [Learning phonology with substantive bias: An experimental and computational study of velar palatalization](#). *Cognitive Science*, 30(5):945–982.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Hyperparameters

When fine-tuning models on syntactic transformations, we use settings from [Mueller et al. \(2022\)](#): batch size 128, window size 128, initial learning rate of 5×10^{-5} , fine-tune for 10 epochs (≈ 7500 training steps), checkpoint and evaluate every 500 steps.

When pre-training models from scratch, we train for 130K training steps, batch size 16 (except for the 1B-word datasets, where we use batch size 128 such that the model sees the entire dataset at least once). We tune over the vocabulary size $\in \{8192, 32768\}$ for each dataset and dataset size.

B Is the Encoder or Decoder More Important for Hierarchical Generalization?

In §5, we found that model depth is more important than model width for enabling LMs to acquire a hierarchical inductive bias. Here, we specifically investigate whether the encoder or decoder of the model is more important by varying the depth of the encoder and decoder individually and observing changes in generalization patterns. As in §5, our models are based on the $T5_{\text{base}}$ architecture, which has 12 encoder and 12 decoder layers.

In our results (Figure 6), we observe that decreasing the depth of either component leads to similar losses in accuracy on passivization, though decreasing decoder depth results in consistently lower accuracies for question formation. Thus, total depth may be the most important factor, regardless of where it is concentrated. Nonetheless, we observe preliminary evidence for the decoder being slightly more important for acquiring a hierarchical inductive bias—or at least generating outputs that are consistent with this bias for question formation. Future work could investigate other transformations and other languages to test the consistency of these findings.

C All Architectural Variation Results

In §5 and App. B, we show that model depth is more important than model width. However, we did not show the performance of models where we vary the number of attention heads, nor the key-value projection matrix dimension. Here, we show the full results (Figure 7).

Overall, varying the number of attention heads has little effect on the performance of the model.

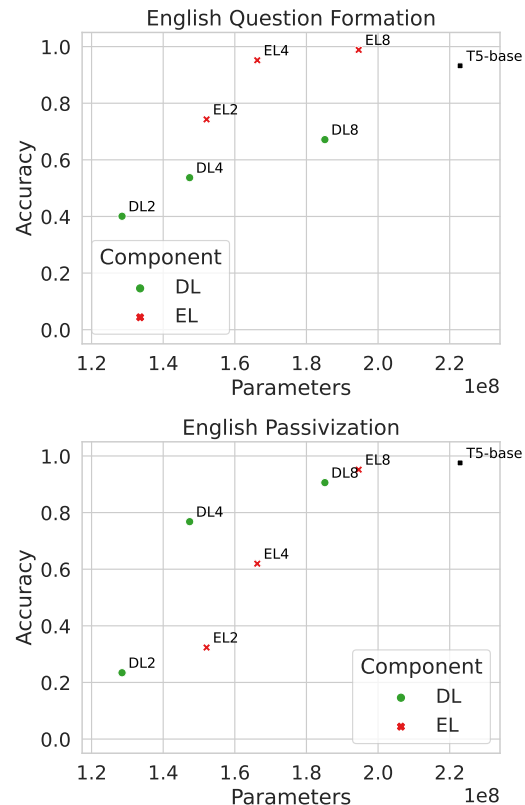


Figure 6: Generalization sequence accuracies on question formation (top) and passivization (bottom) using $T5_{\text{base}}$, as well as variants of $T5_{\text{base}}$ where we vary the encoder depth/decoder depth. Here, EL2 refers to an architecture identical to $T5_{\text{base}}$, except it has 2 encoder layers (instead of 12). Likewise, DL2 is created from $T5_{\text{base}}$ by modifying the number of decoder layers, keeping the number of encoder layers at the original 12.

We see the same trend for reductions in the size of the key/value projection matrix. Thus, model depth still appears to be the most important component in inducing hierarchy-sensitive generalizations.

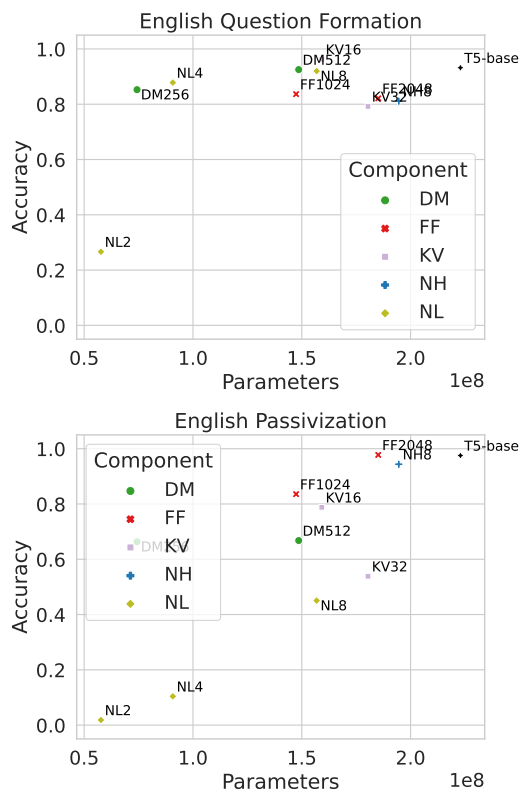


Figure 7: Generalization sequence accuracies on question formation (top) and passivization (bottom) using all architectural variants of $T5_{base}$, as well as $T5_{base}$ itself.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Limitations section at end of paper.
- A2. Did you discuss any potential risks of your work?
We evaluate the syntactic abilities of language models trained on small amounts of data. Our findings have implications for model evaluations, but do not change the models or their behavior.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Section 1.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 3.

- B1. Did you cite the creators of artifacts you used?
Section 3.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The datasets and models we use all use highly permissive licenses, and we are using each as originally intended for research purposes.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
See B2.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we use is synthetically generated using a CFG and contains only semantically implausible sentences. It contains no PII.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C Did you run computational experiments?

Sections 3, 4, 5.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

Table 1; Sections 3, 4, 5.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Sections 3, 4, 5.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 3.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Sections 3, 4, 5.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Not applicable. Left blank.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Not applicable. Left blank.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. Left blank.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Not applicable. Left blank.