Does Syntax Need to Grow on Trees? Sources of Hierarchical Inductive Bias in Sequence-to-Sequence Networks

R. Thomas McCoy
Department of Cognitive Science
Johns Hopkins University
tom.mccoy@jhu.edu

Robert Frank
Department of Linguistics
Yale University
robert.frank@yale.edu

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

Abstract

Learners that are exposed to the same training data might generalize differently due to differing inductive biases. In neural network models, inductive biases could in theory arise from any aspect of the model architecture. We investigate which architectural factors affect the generalization behavior of neural sequence-to-sequence models trained on two syntactic tasks, English question formation and English tense reinflection. For both tasks, the training set is consistent with a generalization based on hierarchical structure and a generalization based on linear order. All architectural factors that we investigated qualitatively affected how models generalized, including factors with no clear connection to hierarchical structure. For example, LSTMs and GRUs displayed qualitatively different inductive biases. However, the only factor that consistently contributed a hierarchical bias across tasks was the use of a tree-structured model rather than a model with sequential recurrence, suggesting that human-like syntactic generalization requires architectural syntactic structure.

1 Introduction

Any finite training set is consistent with multiple generalizations. Therefore, the way that a learner generalizes to unseen examples depends not only on the training data but also on properties of the learner. Suppose a learner is told that a blue triangle is an example of a *blick*. A learner preferring shape-based generalizations would conclude that *blick* means ''triangle,'' while a learner preferring color-based generalizations would conclude that *blick* means ''blue object'' (Landau et al., 1988). Factors that guide a learner to choose one generalization over another are called **inductive biases**.

What properties of a learner cause it to have a particular inductive bias? We investigate this question with respect to sequence-to-sequence neural networks (Botvinick and Plaut, 2006; Sutskever et al., 2014). As a test case for studying differences in how models generalize, we use the syntactic task of **English question formation**, such as transforming (1a) into (1b):

(1) a. The zebra **does** chuckle.
b. **Does** the zebra chuckle?

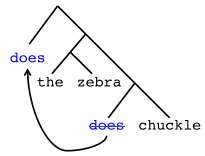
Following Chomsky's (1980) empirical claims about children's linguistic input, we constrain our training set to be consistent with two possible rules illustrated in Figure 1: MOVE-MAIN (a rule based on hierarchical syntactic structure) and MOVE-FIRST (a rule based on linear order). We then evaluate each trained model on examples where the rules make different predictions, such as (2): given (2a), MOVE-MAIN would generate (2b) while MOVE-FIRST would generate (2c):

(2) a. Your zebras that **don't** dance **do** chuckle.
b. **Do** your zebras that **don't** dance chuckle?
c. **Don't** your zebras that dance **do** chuckle?

Since no such examples appear in the training set, a model's behavior on them reveals which rule the model is biased toward. This task allows us to study a particular bias, namely, a bias for hierarchical generalization, which is important for models of

Transactions of the Association for Computational Linguistics, vol. 8, pp. 125–140, 2020. https://doi.org/10.1162/tacl_a_00304
Action Editor: Alexander Clark. Submission batch: 5/2019; Revision batch: 10/2019; Published 2020.

MOVE-MAIN: Move the main verb's auxiliary to the front of the sentence.



MOVE-FIRST: Move the linearly first auxiliary to the front of the sentence.

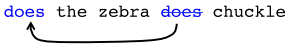


Figure 1: Two potential rules for English question formation.

language because it has been argued to underlie human language acquisition (Chomsky, 1965).

To test which models have a hierarchical bias, we use the question formation task and a second task: **tense reinflection**. For both tasks, our training set is ambiguous between a hierarchical generalization and a linear generalization. If a model chooses the hierarchical generalization for only one task, this preference is likely due to task-specific factors rather than a general hierarchical bias. On the other hand, a consistent preference for hierarchical generalizations across tasks would provide converging evidence that a model has a hierarchical bias. We find that all the factors we tested can qualitatively affect how a model generalizes on the question formation task. These factors are the type of recurrent unit, the type of attention, and the choice of sequential vs. tree-based model structure. Even though all these factors affected the model's decision between MOVE-MAIN and MOVE-FIRST, only the use of a tree-based model can be said to impart a hierarchical bias, since this was the only model type that chose a hierarchical generalization across both of our tasks. Specific findings that support these general conclusions include:

- Generalization behavior is profoundly affected by the type of recurrent unit and the type of attention, and also by the *interactions* between these factors.
- LSTMs and GRUs have qualitatively different inductive biases. The difference appears at least partly due to the fact that the values in GRU hidden states are bounded within a particular interval (Weiss et al., 2018).
- Only a model built around the correct tree structure displayed a robust hierarchical bias across tasks. Sequentially structured models failed to generalize hierarchically even when the input contained explicit marking of each sentence's hierarchical structure.

Overall, we conclude that many factors can qualitatively affect a model's inductive biases, but human-like syntactic generalization may require specific types of high-level structure, at least when learning from text alone.

2 The Question Formation Task

2.1 Background

The classic discussion of the acquisition of English question formation begins with two empirical claims: (i) disambiguating examples such as Example (2) rarely occur in a child's linguistic input, but (ii) all learners of English nevertheless acquire MOVE-MAIN rather than MOVE-FIRST. Chomsky (1965, 1980) uses these points to argue that humans must have an innate bias toward learning syntactic rules that are based on hierarchy rather than linear order (this argument is known as the *argument from the poverty of the stimulus*).

There has been a long debate about this line of argument. Though some have discussed the validity of Chomsky's empirical claims (Crain and Nakayama, 1987; Ambridge et al., 2008; Pullum and Scholz, 2002; Legate and Yang, 2002), most of the debate has been about which mechanisms could explain the preference for MOVE-MAIN. These mechanisms include an assumption of substitutability (Clark and Eyraud, 2007), a bias for simplicity (Perfors et al., 2011), exploitation of statistical patterns (Lewis and Elman, 2001; Reali and Christiansen, 2005), and semantic knowledge (Fitz and Chang, 2017); see Clark and Lappin (2010) for in-depth discussion.

These past works focus on the *content* of the bias that favors MOVE-MAIN (i.e., which types of generalizations the bias supports), but we instead focus on the *source* of this bias (i.e., which factors of the learner give rise to the bias). In the book *Rethinking Innateness*, Elman et al. (1998) argue that innate biases in humans must arise from architectural constraints on the neural connections in the brain rather than from constraints stated at the symbolic level, under the assumption that symbolic constraints are unlikely to be specified in the genome. Here we use artificial neural networks

	Training set, validation set, test set		Generalization set

	DECL	QUEST
No RC	the newts do see my yak by the zebra . → the newts do see my yak by the zebra .	the newts do see my yak by the zebra . → do the newts see my yak by the zebra ?
RC on object	the newts do see my yak who does fly . → the newts do see my yak who does fly .	the newts do see my yak who does fly . → do the newts see my yak who does fly ?
RC on subject	the newts who don't fly do see my yak . → the newts who don't fly do see my yak .	the newts who don't fly do see my yak . → do the newts who don't fly see my yak ?

Figure 2: The difference between the training set and generalization set. To save space, this table uses some words not present in the vocabulary used to generate the examples. RC stands for ''relative clause.''

to investigate whether syntactic inductive biases can emerge from architectural constraints.

2.2 Framing of the Task

Following Frank and Mathis (2007) and McCoy et al. (2018), we train models to take a declarative sentence as input and to either output the same sentence unchanged, or transform that sentence into a question. The sentences were generated from a context-free grammar containing only the sentence types shown in Figure 2 and using a 68-word vocabulary; the full grammar is at the project Web site.[1] The different types of sentences vary in the linear position of the main auxiliary, such that a model cannot identify the main auxiliary with a simple positional heuristic. The task to be performed is indicated by the final input token, as in Examples (3) and (4):

(3) a. *Input:* your zebra does read . DECL
b. *Output:* your zebra does read .

(4) a. *Input:* your zebra does read . QUEST
b. *Output:* does your zebra read ?

During training, all question formation examples are consistent with both MOVE-FIRST and MOVE-MAIN, such that there is no direct evidence favoring one rule over the other (see Figure 2).

To assess how models generalize, we evaluate them on a generalization set consisting of examples where MOVE-MAIN and MOVE-FIRST make different predictions due to the presence of a relative clause on the subject (see sentence (2a)).

[1]Our code is at `github.com/tommccoy1/rnn-hierarchical-biases`. Results for the over 3,500 models trained for this paper, with example outputs, are at `rtmccoy.com/rnn_hierarchical_biases.html`; only aggregate (median) results are reported here.

2.3 Evaluation Metrics

We focus on two metrics. The first is **full-sentence accuracy on the test set**. That is, for examples drawn from the same distribution as the training set, does the model get the output exactly right?

For testing generalization to the withheld example type, a natural metric would be full-sentence accuracy on the generalization set. However, in preliminary experiments we found that most models rarely produced the exact output predicted by either MOVE-MAIN or MOVE-FIRST, as they tend to truncate the output, confuse similar words, and make other extraneous errors. To abstract away from such errors, we use **first-word accuracy on the generalization set**. With both MOVE-FIRST and MOVE-MAIN, the first word of the question is the auxiliary that has been moved from within the sentence. If the auxiliaries in the relative and main clauses are distinct, this word alone is sufficient to differentiate the two rules. For example, in the bottom right cell of Figure 2, MOVE-MAIN predicts having *do* at the start, whereas MOVE-FIRST predicts *don't*.[2] Models almost always produced either the main auxiliary or the first auxiliary as the first word of the output (over 98% of the time for most models[3]), so a low first-word accuracy can be interpreted as high consistency with MOVE-FIRST.

2.4 Architecture

We used the sequence-to-sequence architecture in Figure 3 (Sutskever et al., 2014). This model consists of two neural networks: the **encoder** and

[2]We exclude from the generalization set cases where the two auxiliaries are the same. We also exclude cases where one auxiliary is singular and the other plural so that a model cannot succeed by using heuristics based on the grammatical number of the subject.

[3]The one exception is noted in the caption to Figure 4.

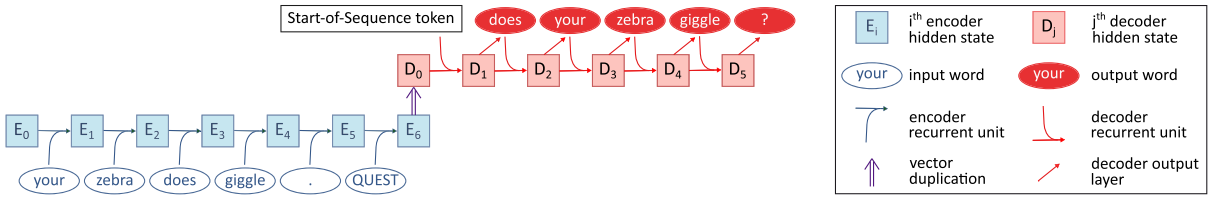


Figure 3: Sequential sequence-to-sequence model.

the **decoder**. The encoder is fed the input sentence one word at a time; after each word, the encoder updates its **hidden state**, a vector representation of the information encountered so far. After the encoder has been fed the entire input, its final hidden state (E_6 in Figure 3) is fed to the decoder, which generates an output sequence one word at a time based on its own hidden state, which is updated after each output word. The weights that the encoder and decoder use to update their hidden states and generate outputs are learned via gradient descent; for more details, see Appendix A.

2.5 Overview of Experiments

Holding the task constant, we first varied two aspects of the architecture that have no clear connection to question formation, namely, the recurrent unit and the type of attention; both of these aspects have been central to major advances in natural language processing (Sundermeyer et al., 2012; Bahdanau et al., 2015), so we investigate them here to see whether their contributions might be partially explained by linguistically relevant inductive biases that they impart. We also tested a more clearly task-relevant modification of the architecture, namely the use of tree-based models rather than the sequential structure in Figure 3.

3 Recurrent Unit and Attention

3.1 Recurrent Unit

The recurrent unit is the component that updates the hidden state after each word for the encoder and decoder. We used three types of recurrent units: simple recurrent networks (SRNs; Elman, 1990), gated recurrent units (GRUs; Cho et al., 2014), and long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997). In SRNs and GRUs, the hidden state is represented by a single vector, whereas LSTMs use two vectors (the hidden state and the cell state). In addition, GRUs and LSTMs both use *gates*, which control what information is retained across time steps, whereas SRNs do not; GRUs and LSTMs differ from each other in the number and types of gates they use.

3.2 Attention

In the basic model in Figure 3, the final hidden state of the encoder is the decoder's only source of information about the input. To avoid having such a bottleneck, many contemporary sequence-to-sequence models use **attention** (Bahdanau et al., 2015), a feature that enables the decoder to consider all encoder hidden states (E_0 through E_6 in Figure 3) when generating hidden state D_i. A model without attention has the only inputs to D_i being D_{i-1} and y_{i-1} (the previous output); attention adds a third input, $c_i = \sum_j \alpha_i[j]E_j$, which is a weighted sum of the encoder's hidden states (E_0 through E_n) using a weight vector α_i whose j^{th} element is denoted by $\alpha_i[j]$.

Implementations of attention vary in how the weights $\alpha_i[j]$ are derived (Graves et al., 2014; Chorowski et al., 2015; Luong et al., 2015). Attention can be solely **location-based**, where each α_i is determined solely from D_{i-1} (and potentially also y_{i-1}), so that the model chooses *where* to attend without first checking *what* it is attending to. Alternately, attention could be **content-based**, in which case each $\alpha_i[j]$ is determined from both D_{i-1} and E_j, such that the model does consider what it might attend to before attending to it. We test both location-based and content-based attention, and we also test models without attention.

3.3 Results

We trained models with all nine possible combinations of recurrent unit and attention type, using the hyperparameters and training procedure described in Appendix A. The results are in Figure 4.

The SRN without attention failed on the test set, mainly because it often confused words that had the same part of speech, a known weakness

	🚫	🌐	📖
SRN	0.00	0.93	1.00
GRU	0.88	0.77	1.00
LSTM	0.94	0.98	1.00

(a) Full-sentence accuracy on the test set

	🚫	🌐	📖
SRN		0.43	0.64
GRU	0.01	0.78	0.17
LSTM	0.02	0.05	0.01

(b) First-word accuracy on the generalization set

Figure 4: Results for each combination of recurrent unit and attention type. All numbers are medians over 100 initializations. 🚫 = no attention; 🌐 = location-based attention; 📖 = content-based attention. A grayed-out cell indicates that the architecture scored below 50% on the test set. In (b), the SRN 🌐 produced the first auxiliary 45% of the time; for all other models, the proportion of first-auxiliary outputs is almost exactly one minus the first-word accuracy (i.e., the proportion of main-auxiliary outputs).

	Unsquashed	Squashed
GRU 🌐	0.97	0.77
LSTM 🌐	0.98	0.98

(a) Full-sentence accuracy on the test set

	Unsquashed	Squashed
GRU 🌐	0.54	0.78
LSTM 🌐	0.05	0.43

(b) First-word accuracy on the generalization set

Figure 5: Effects of squashing. All numbers are medians across 100 initializations. The standard versions of the architectures are the squashed GRU and the unsquashed LSTM.

of SRNs (Frank and Mathis, 2007). Therefore, its generalization set behavior is uninformative. The other architectures performed strongly on the test set (>50% full-sentence accuracy), so we now consider their generalization set performance. The GRU with location-based attention and the SRN with content-based attention both preferred MOVE-MAIN, while the remaining architectures preferred MOVE-FIRST.[4] These results suggest that both the recurrent unit and the type of attention can qualitatively affect a model's inductive biases. Moreover, the *interactions* of these factors can have drastic effects: with SRNs, content-based attention led to behavior consistent with MOVE-MAIN while location-based attention led to behavior consistent with MOVE-FIRST; these types of attention had opposite effects with GRUs.

3.4 Differences between LSTMs and GRUs

One striking result in Figure 4 is that LSTMs and GRUs display qualitative differences, even though the two architectures are often viewed as interchangeable and achieve similar performance in applied tasks (Chung et al., 2014). One difference between LSTMs and GRUs is that a squashing function is applied to the hidden state of a GRU to keep its values within the range $(-1, 1)$, while the cell state of an LSTM is not bounded. Weiss et al. (2018) demonstrate that such squashing leads to a qualitative difference in how well these models generalize counting behavior. Such squashing may also explain the qualitative differences that we observe: Counting the input elements is equivalent to keeping track of their linear positions, so we might expect that a tendency to count would make the linear generalization more accessible.

To test whether squashing increases a model's preference for MOVE-MAIN, we created a modified LSTM that included squashing in the calculation of its cell state, and a modified GRU that did not have the squashing usually present in GRUs. See Appendix B for more details. Using the same training setup as before, we trained models with these modified recurrent units and with location-based attention. LSTMs and GRUs with squashing chose MOVE-MAIN more often than the corresponding models without squashing (Figure 5), suggesting that such squashing is one factor that causes GRUs to behave differently than LSTMs.

3.5 Hyperparameters and Random Seed

In addition to variation across architectures, we also observed considerable variation across multiple instances of the same architecture that differed only in random seed; the random seeds determined both the initial weights of each model and the order in which training examples

[4] We say that a model *preferred* generalization A over generalization B if it behaved more consistently with A than B.

were sampled. For example, the generalization set first-word accuracy for SRNs with content-based attention ranged from 0.17 to 0.90. Based on our exploration of hyperparameters, it also appears that the learning rate and hidden size can qualitatively affect generalization. The effects of these details are difficult to interpret systematically, and we leave the characterization of their effects for future work. Results for all individual re-runs are at the project Web site.

4 Tree Models

So far we have tested whether properties that are not interpretably related to hierarchical structure nevertheless affect how a model generalizes on a syntactic task. We now turn to a related but opposite question: when a model's design is meant to give it a hierarchical inductive bias, does this design succeed at giving the model this bias?

4.1 Tree Model that Learns Implicit Structure

The first hierarchical model that we test is the Ordered Neurons LSTM (ON-LSTM; Shen et al., 2019). This model is not given the tree structure of each sentence as part of its input. Instead, its processing is structured in a way that leads to the implicit construction of a soft parse tree. This implicit tree structure is created by imposing a stack-like constraint on the updates to the values in the cell state of an LSTM: The degree to which the i^{th} value is updated must always be less than or equal to the degree to which the j^{th} value is updated for all $j \leq i$. This hierarchy of cell-state values adds an implicit tree structure to the model, where each level in the tree is defined by a soft depth in the cell state to which that level extends.

We re-implemented the ON-LSTM and trained 100 instances of it using the hyperparameters specified in Appendix A. This model achieved a test set full-sentence accuracy of 0.93 but a generalization set first-word accuracy of 0.05, showing a strong preference for MOVE-FIRST over MOVE-MAIN, contrary to what one would expect from a model with a hierarchical inductive bias. This lack of hierarchical behavior might be explained by the findings of Dyer et al. (2019) that ON-LSTMs do not perform much better than standard LSTMs at implicitly recovering hierarchical structure, even though ON-LSTMs (but not standard LSTMs) were designed in a way intended to impart a hierarchical bias. According to Dyer et al. (2019), the ON-LSTM's apparent success reported in Shen et al. (2019) was largely due to the method used to analyze the model rather than the model itself.

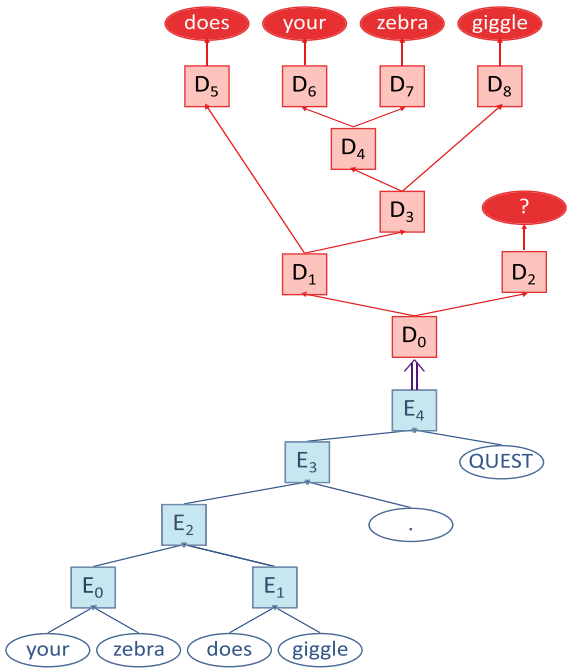


Figure 6: Sequence-to-sequence network with a tree-based encoder and tree-based decoder.

4.2 Tree Models Given Explicit Structure

The ON-LSTM results show that hierarchically structured processing alone is not sufficient to induce a bias for MOVE-MAIN, suggesting that constraints on *which* trees are used may also be necessary. We therefore tested a second type of hierarchical model, namely, Tree-RNNs, that were explicitly fed the correct parse tree. Parse trees can be used to guide the encoder, the decoder, or both; Figure 6 shows a model where both the encoder and decoder are tree-based. For the tree-based encoder, we use the Tree-GRU from Chen et al. (2017). This model composes the vector representations for a pair of sister nodes to generate a vector representing their parent. It performs this composition bottom-up, starting with the word embeddings at the leaves and ending with a single vector representing the root (E_4 in Figure 6); this vector acts as the encoding of the input. For the tree-based decoder, we use a model based on the Tree-LSTM decoder from Chen et al. (2018), but using a GRU instead of an LSTM, for consistency with the tree encoder.

Model	Full-sentence test acc.	First-word gen. acc.
Sequential/Sequential	0.88	0.01
Sequential/Tree	0.00	0.90
Tree/Sequential	0.96	0.13
Tree/Tree	0.96	0.99

Figure 7: Results with tree-based models (medians over 100 initializations). Model names indicate encoder/decoder; e.g., Sequential/Tree has a sequential GRU encoder and a tree-GRU decoder.

This tree decoder is the mirror image of the tree encoder: starting with the vector representation of the root node (D_0 in Figure 6), it takes the vector representation of a parent node and outputs two vectors, one for the left child and one for the right child, until it reaches a leaf node, where it outputs a word. We test models with a tree-based encoder and sequential decoder, a sequential encoder and tree-based decoder, or a tree-based encoder and tree-based decoder, all without attention; we investigate these variations to determine whether hierarchical generalization is determined by the encoder, the decoder, or both.

The results for these models are in Figure 7, along with the previous results of the fully sequential GRU (sequential encoder + sequential decoder) without attention for comparison. The model with a tree-based encoder and sequential decoder preferred MOVE-FIRST, like the fully sequential model. Only the models with a tree-based decoder preferred MOVE-MAIN, consistent with the finding of McCoy et al. (2019) that it is the decoder that determines an encoder-decoder model's representations. However, the model with a sequential encoder and a tree decoder failed on the test set, so the only model that both succeeded on the test set and showed a bias toward a MOVE-MAIN generalization was the fully tree-based model (Tree/Tree).[5] The behavior of this Tree/Tree model was striking in another way as well: Its generalization set *full-sentence* accuracy was 69%, while all other models—even those that achieved high *first-word* accuracy on the generalization set—had close to 0% generalization set full-sentence accuracy. The ON-LSTM and Tree-GRU results show that an architecture designed to have a certain inductive bias might, but will not necessarily, display the intended bias.

[5]We do not have an explanation for the failure of the Sequential/Tree model on the test set; most of its errors involved confusion among words that had the same part of speech (e.g., generating *my* instead of *your*).

5 Tense Reinflection

We have shown that several models reliably preferred MOVE-MAIN over MOVE-FIRST. However, this behavior alone does not necessarily mean that these models have a hierarchical bias, because a preference for MOVE-MAIN might arise not from a hierarchical bias but rather from some task-specific factors such as the prevalence of certain n-grams (Kam et al., 2008; Berwick et al., 2011). A true hierarchical bias would lead a model to adopt hierarchical generalizations across training tasks; by contrast, we hypothesize that other factors (such as a bias for focusing on n-gram statistics) will be more sensitive to details of the task and will thus be unlikely to consistently produce hierarchical preferences. To test the robustness of the hierarchical preferences of our models, then, we introduce a second task, **tense reinflection**.

5.1 Reinflection Task

The reinflection task uses English subject–verb agreement to illuminate a model's syntactic generalizations (Linzen et al., 2016). The model is fed a past-tense English sentence as input. It must then output that sentence either unchanged or transformed to the present tense, with the final word of the input indicating the task to be performed:

(5) my yak swam . PAST → my yak swam .

(6) my yak swam . PRESENT → my yak swims .

Because the past tense in English does not inflect for number (e.g., the past tense of *swim* is *swam* whether the subject is singular or plural), the model must determine from context whether each verb being turned to present tense should be singular or plural. Example (6) is consistent with two salient rules for determining which aspects of the context are relevant:

(7) AGREE-SUBJECT: Each verb should agree with its hierarchically determined subject.

(8) AGREE-RECENT: Each verb should agree with the linearly most recent noun.

Though these rules make the same prediction for (6), they make different predictions for other

examples, such as (9a), for which AGREE-SUBJECT predicts (9b) whereas AGREE-RECENT predicts (9c):

(9) a. my zebra by the yaks swam . PRESENT
b. my **zebra** by the yaks **swims** .
c. my zebra by the **yaks swim** .

Similar to the setup for the question formation experiments, we trained models on examples for which AGREE-SUBJECT and AGREE-RECENT made the same predictions and evaluated the trained models on examples where the rules make different predictions. We ran this experiment with all 9 sequential models ([SRN, GRU, LSTM] x [no attention, location-based attention, content-based attention]), the ON-LSTM, and the model with a tree-based encoder and tree-based decoder that were provided the correct parse trees, using the hyperparameters in Appendix A. The example sentences were generated using the same context-free grammar used for the question formation task, except with inflected verbs instead of auxiliary/verb bigrams (e.g., *reads* instead of *does read*). We evaluated these models on the full-sentence accuracy on the test set and also main-verb accuracy for the generalization set—that is, the proportion of generalization set examples for which the main verb was correctly predicted, such as when *swims* rather than *swim* was chosen in the output for (9a). Models usually chose the correct lemma for the main verb (at least 87% of the time for all tense reinflection models), with most main verb errors involving the correct verb but with incorrect inflection (i.e., being singular instead of plural, or vice versa). Thus, a low main-verb accuracy can be interpreted as consistency with AGREE-RECENT.

All sequential models, even the ones that generalized hierarchically with question formation, overwhelmingly chose AGREE-RECENT for this reinflection task (Figure 8), consistent with the results of a similar experiment done by Ravfogel et al. (2019). The ON-LSTM also preferred AGREE-RECENT. By contrast, the fully tree-based model preferred the hierarchical generalization AGREE-SUBJECT. Thus, although the question formation experiments showed qualitative differences in sequential models' inductive biases, this experiment shows that those differences cannot be explained by positing that there is a general hierarchical bias in some of our sequential models. What the relevant bias for these models *is* remains unclear; we only claim to show that it is not a hierarchical bias. Overall, the model with both a tree-based encoder and a tree-based decoder is the only model we tested that plausibly has a generic hierarchical bias, as it is the only one that behaved consistently with such a bias across both tasks.

Model	Full-sentence test acc.	Main-verb gen. acc.
SRN 🚫	0.00	
SRN 🌐	1.00	0.00
SRN 📖	1.00	0.00
GRU 🚫	0.90	0.04
GRU 🌐	0.81	0.00
GRU 📖	1.00	0.00
LSTM 🚫	0.96	0.04
LSTM 🌐	0.98	0.00
LSTM 📖	1.00	0.00
ON-LSTM 🚫	0.95	0.05
Tree/Tree 🚫	0.96	0.94

Figure 8: Reinflection results (medians over 100 initializations). 🚫 = no attention; 🌐 = location-based attention; 📖 = content-based attention.

6 Are Tree Models Constrained to Generalize Hierarchically?

It may seem that the tree-based models are constrained by their structure to make only hierarchical generalizations, rendering their hierarchical generalization trivial. In this section, we test whether they are in fact constrained in this way, and similarly whether sequential models are constrained to make only linear generalizations. Earlier, the training sets for our two tasks were ambiguous between two generalizations, but we now used training sets that unambiguously supported either a linear transformation or a hierarchical transformation.[6] For example, we used a MOVE-MAIN training set that included some examples like (10a), whereas the MOVE-FIRST training set included some examples like (10b):

(10) a. my yaks that do read don't giggle . QUEST
→ don't my yaks that do read giggle ?
b. my yaks that do read don't giggle . QUEST
→ do my yaks that read don't giggle ?

Similarly, for the tense reinflection task, we created an AGREE-SUBJECT training set and an AGREE-RECENT training set. For each of these four training

[6]The lack of ambiguity in each training set means that the generalization set becomes essentially another test set.

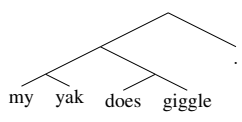
sets, we trained 100 sequential GRUs and 100 Tree/Tree GRUs, all without attention.

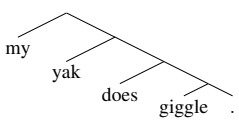
Each model learned to perform linear and hierarchical transformations with similar accuracy: On the MOVE-MAIN and MOVE-FIRST datasets, both the sequential and tree-based models achieved 100% first-word accuracy. On both the AGREE-SUBJECT and AGREE-RECENT datasets, the sequential model achieved 91% main-verb accuracy and the tree-based model achieved 99% main-verb accuracy. Thus, the fact that the tree-based model preferred hierarchical generalizations when the training set was ambiguous arose not from any constraint imposed by the tree structure but rather from the model's inductive biases—biases that can be overridden given appropriate training data.

7 Tree Structure vs. Tree Information

Our sequential and tree-based models differ not only in structure but also in the information they have been provided: The tree-based models have been given correct parse trees for their input and output sentences, while the sequential models have not been given parse information. Therefore, it is unclear whether the hierarchical generalization displayed by the tree-based models arose from the tree-based model structure, from the parse information provided to the models, or both.

To disentangle these factors, we ran two further experiments. First, we retrained the Tree/Tree GRU but using uniformly right-branching trees (as in (11b)) instead of correct parses (as in (11a)). Thus, these models make use of tree structure but not the kind of parse structure that captures linguistic information. Second, we retrained the sequential GRU without attention[7] but modified the input and output by adding brackets that indicate each sentence's parse; for example, (12a) would be changed to (12b). Thus, these models are provided with parse information in the input but such structure does not guide the neural network computation as it does with tree RNNs.

(11) a. 

b. 

[7]We chose this sequential model because the Tree/Tree model is also based on GRUs without attention.

	Not provided correct parse	Provided correct parse
GRU 🚫	0.01	0.38
Tree/Tree 🚫	0.05	0.99

(a) Question formation generalization set results.

	Not provided correct parse	Provided correct parse
GRU 🚫	0.04	0.00
Tree/Tree 🚫	0.07	0.94

(b) Tense reinflection generalization set results.

Figure 9: Disentangling tree structure and parse information. The GRU 🚫 that is not provided the correct parse is the same as GRU 🚫 in Figures 4 and 8. The Tree/Tree model that is provided the correct parse is the same as the Tree/Tree model in Figures 7 and 8. The other two conditions are new: The GRU 🚫 that was provided the correct parses was given these parses via bracketing, while the Tree/Tree model that was not provided the correct parses was instead given right-branching trees.

(12) a. my yak does giggle . QUEST
→ does my yak giggle ?
b. [[[my yak] [does giggle]] .] QUEST
→ [[does [[my yak] giggle]] ?]

We ran 100 instances of each experiment using different random seeds. For the experiment with bracketed input, the brackets significantly increased the lengths of the sentences, making the learning task harder; we therefore found it necessary to use a patience of 6 instead of the patience of 3 we used elsewhere, but all other hyperparameters remained as described in Appendix A.

For both tasks, neither the sequential GRU that was given brackets in its input nor the Tree/Tree model that was given right-branching trees displayed a hierarchical bias (Figure 9).[8] The lack of hierarchical bias in the sequential GRU with bracketed input indicates that simply providing parse information in the input and target output is insufficient to induce a model to favor hierarchical generalization; it appears that such parse information must be integrated into the model's structure to be effective. On

[8]Providing the parse with brackets did significantly improve the first-word accuracy of the sequential GRU, but this accuracy remained below 50%.

the other hand, the lack of a hierarchical bias in the Tree/Tree model using right-branching trees shows that simply having tree structure is also insufficient; it is necessary to have the *correct* tree structure.

8 Will Models Generalize Across Transformations?

Each experiment discussed so far involved a single linguistic transformation. By contrast, humans acquiring language are not exposed to phenomena in isolation but rather to a complete language encompassing many phenomena. This fact has been pointed to as a possible way to explain hierarchical generalization in humans without needing to postulate any innate preference for hierarchical structure. While one phenomenon, such as question formation, might be ambiguous in the input, there might be enough direct evidence among other phenomena to conclude that the language as a whole is hierarchical, a fact which learners can then extend to the ambiguous phenomenon (Pullum and Scholz, 2002; Perfors et al., 2011), under the nontrivial assumption that the learner will choose to treat the disparate phenomena in a unified fashion.

While our training sets are ambiguous with respect to whether the phenomenon underlying the mapping is structurally driven, they do contain other cues that the language is more generally governed by hierarchical regularities. First, certain structural units are reused across positions in a sentence; for example, prepositional phrases can appear next to subjects or objects. Such reuse of structure can be represented more efficiently with a hierarchical grammar than a linear one. Second, in the question formation task, subject–verb agreement can also act as a cue to hierarchical structure: For example, in the sentence *my* ***walrus*** *by the yaks* ***does*** *read*, the inflection of *does* depends on the verb's hierarchically determined subject (*walrus*) rather than the linearly closest noun (*yaks*).[9]

For the sequential RNNs we have investigated, it appears that these indirect cues to hierarchical structure were not sufficient to guide the models towards hierarchical generalizations. However, perhaps the inclusion of some more direct evidence for hierarchy would be more successful.

[9]Subject–verb agreement does not act as a cue to hierarchy in the tense reinflection task because all relevant sentences have been withheld to maintain the training set's ambiguity.

	Ambiguous question formation	Ambiguous tense reinflection
Single-task	0.01	0.04
Multi-task	0.09	0.17
+auxiliaries	0.01	0.99

Figure 10: Multi-task learning results for a GRU without attention. *Single-task* reports baselines from training on a single ambiguous task. *Multi-task* reports results from adding an unambiguous second task. *Multi-task + auxiliaries* reports results from adding an unambiguous second task and also adding overt auxiliaries to the tense reinflection sentences. The numbers give the generalization set performance on the ambiguous task.

To take a first step toward investigating this possibility, we use a multi-task learning setup, where we train a single model to perform both question formation and tense reinflection. We set up the training set such that one task was unambiguously hierarchical while the other was ambiguous between the hierarchical generalization and the linear generalization. This gave two settings: One where question formation was ambiguous, and one where tense reinflection was ambiguous. We trained 100 instances of a GRU without attention on each setting and assessed how each model generalized for the task that was ambiguous.

For both cases, generalization behavior in the multi-task setting differed only minimally from the single-task setting (Figure 10). One potential explanation for the lack of transfer across tasks is that the two tasks operated over different sentence structures: the question formation sentences always contained overt auxiliaries on their verbs (e.g., *my walrus does giggle*), while the tense reinflection sentences did not (e.g., *my walrus giggles*). To test this possibility, we reran the multi-task experiments but with overt auxiliaries added to the tense reinflection sentences (Figure 10, ''Multi-task + auxiliaries'' row). In this setting, the model still generalized linearly when it was question formation that was ambiguous. However, when it was tense reinflection that was ambiguous, the model generalized hierarchically.

We hypothesize that the directionality of this transfer is due to the fact that the question formation training set includes unambiguous long-distance subject–verb agreement as in (13), which

might help the model on generalization-set examples for tense reinflection such as Example (14):

(13) my **zebras** by the yak **do** read . DECL
→ my **zebras** by the yak **do** read .

(14) my zebras by the yak did read . PRESENT
→ my **zebras** by the yak **do** read .

By contrast, the tense reinflection training set does not contain any outputs of the type withheld from the question formation training set. If this explanation is correct, it would mean that the improvement on the tense reinflection task derived not from the question formation *transformation* but rather from the subject–verb agreement incidentally present in the question formation *dataset*. Therefore, even the single potential case of generalization across transformations is likely spurious.

Recent NLP work has also found that neural networks do not readily transfer knowledge across tasks; e.g., pretrained models often perform worse than non-pretrained models (Wang et al., 2019). This lack of generalization across tasks might be due to the tendency of multi-task neural networks to create largely independent representations for different tasks even when a shared representation could be used (Kirov and Frank, 2012). Therefore, to make cross-phenomenon generalizations, neural networks may need to be given an explicit bias for sharing processing across phenomena.

9 Discussion

We have found that all factors we tested can qualitatively affect a model's inductive biases but that a hierarchical bias—which has been argued to underlie children's acquisition of syntax—only arose in a model whose inputs and computations were governed by syntactic structure.

9.1 Relation to *Rethinking Innateness*

Our experiments were motivated in part by the book *Rethinking Innateness* (Elman et al., 1998), which argued that humans' inductive biases must arise from constraints on the wiring patterns of the brain. Our results support two conclusions from this book. First, those authors argued that ''Dramatic effects can be produced by small changes'' (p. 359). This claim is supported by our observation that low-level factors, such as the size of the hidden state, qualitatively affect how models generalize (Section 3.5). Second, they argued that ''[w]hat appear to be single events or behaviors may have a multiplicity of underlying causes'' (p. 359); in our case, we found that a model's generalization behavior results from some combination of factors that interact in hard-to-interpret ways; for example, changing the type of attention had different effects in SRNs than in GRUs.

The dramatic effects of these low-level factors offer some support for the claim that humans' inductive biases can arise from fine-grained architectural constraints in the brain. However, this support is only partial. Our only model that robustly displayed the kind of preference for hierarchical generalization that is necessary for language learning did not derive such a preference from low-level architectural properties but rather from the explicit encoding of linguistic structure.

9.2 Relation to Human Language Acquisition

Our experiments showed that some tree-based models displayed a hierarchical bias, although non-tree-based models never displayed such a bias, even when provided with strong cues to hierarchical structure in their input (through bracketing or multi-task learning). These findings suggest that the hierarchical preference displayed by humans when acquiring English requires making explicit reference to hierachical structure, and cannot be argued to emerge from more general biases applied to input containing cues to hierarchical structure. Moreover, because the only successful hierarchical model was one that took the correct parse trees as input, our results suggest that a child's set of biases includes biases governing which specific trees will be learned. Such biases could involve innate knowledge of likely tree structures, but they do not need to; they might instead involve innate tendencies to bootstrap parse trees from other sources, such as prosody (Morgan and Demuth, 1996) or semantics (Pinker, 1996). With such information, children might learn their language's basic syntax before beginning to acquire question formation, and this knowledge might then guide their acquisition of question formation.

There are three important caveats for extending our conclusions to humans. First, humans may have a stronger bias to share processing across

phenomena than neural networks do, in which case multi-task learning would be a viable explanation for the biases displayed by humans even though it had little effect on our models. Indeed, this sort of cross-phenomenon consistency is similar in spirit to the principle of systematicity, and it has long been argued that humans have a strong bias for systematicity whereas neural networks do not (e.g., Fodor and Pylyshyn, 1988; Lake and Baroni, 2018). Second, some have argued that children's input actually does contain utterances unambiguously supporting a hierarchical transformation (Pullum and Scholz, 2002), whereas we have assumed a complete lack of such examples. Finally, our training data omit many cues to hierarchical structure that are available to children, including prosody and real-world grounding. It is possible that, with data closer to a child's input, more general inductive biases might succeed.

However, there is still significant value in studying what can be learned from strings alone, because we are unlikely to understand how the multiple components of a child's input interact without a better understanding of each component. Furthermore, during the acquisition of abstract aspects of language, real-world grounding is not always useful in the absence of linguistic biases (Gleitman and Gleitman, 1992). More generally, it is easily possible for learning to be harder when there is more information available than when there is less information available (Dupoux, 2018). Thus, our restricted experimental setup may actually make learning easier than in the more informationally-rich scenario faced by children.

9.3 Practical Takeaways

Our results leave room for three possible approaches to imparting a model with a hierarchical bias. First, one could search the space of hyperparameters and random seeds to find a setting that leads to the desired generalization. However, this may be ineffective: At least in our limited exploration of these factors, we did not find a hyperparameter setting that led to hierarchical generalization across tasks for any non-tree-based model.

A second option is to add a pre-training task or use multi-task learning (Caruana, 1997; Collobert and Weston, 2008; Enguehard et al., 2017), where the additional task is designed to highlight hierarchical structure. Most of our multi-task experiments only achieved modest improvements over the single-task setting, suggesting that this approach is also not very viable. However, it is possible that further secondary tasks would bring further gains, making this approach more effective.

A final option is to use more interpretable architectures with explicit hierachical structure. Our results suggest that this approach is the most viable, as it yielded models that reliably generalized hierarchically. However, this approach only worked when the architectural bias was augmented with rich assumptions about the input to the learner, namely that it provided correct hierarchical parses for all sentences. We leave for future work an investigation of how to effectively use tree-based models without providing correct parses.

Acknowledgments

For helpful comments we thank Joe Pater, Paul Smolensky, the JHU Computation and Psycholinguistics lab, the JHU Neurosymbolic Computation lab, the Computational Linguistics at Yale (CLAY) lab, the anonymous reviewers, and audiences at the University of Pavia Center for Neurocognition, Epistemology, and Theoretical Syntax, the Penn State Department of Computer Science and Engineering, and the MIT Department of Brain and Cognitive Sciences. Any errors are our own.

This material is based upon work supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under grant no. 1746891, and by NSF grant nos. BCS-1920924 and BCS-1919321. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Our experiments were conducted with resources from the Maryland Advanced Research Computing Center (MARCC).

A Architecture and Training Details

We used a word embedding size of 256 (with word embeddings learned from scratch), a hidden size of 256, a learning rate of 0.001, and a batch size of 5. Models were evaluated on a validation set after every 1,000 training batches, and we halted training if the model had been trained for at least 30,000 batches and had shown no

improvement over 3 consecutive evaluations on the validation set (the number 3 in this context is called the patience). The training set contained 100,000 examples, while the validation, test, and generalization sets contained 10,000 examples each. The datasets were held constant across experiments, but models sampled from the training set in different orders across experiments. During training, we used teacher forcing on 50% of examples.

B Equations for Squashing Experiments

The equations governing a standard LSTM are:

$$i_t = \sigma(W_i[h_{t-1}, w_t] + b_i) \quad \text{(B.1)}$$
$$f_t = \sigma(W_f[h_{t-1}, w_t] + b_f) \quad \text{(B.2)}$$
$$g_t = \tanh(W_g[h_{t-1}, w_t] + b_g) \quad \text{(B.3)}$$
$$o_t = \sigma(W_o[h_{t-1}, w_t] + b_o) \quad \text{(B.4)}$$
$$c_t = f_t * c_{t-1} + i_t * g_t \quad \text{(B.5)}$$
$$h_t = o_t * \tanh(c_t) \quad \text{(B.6)}$$

To create a new LSTM whose cell state exhibits squashing, like the hidden state of the GRU, we modified the LSTM cell state update in (B.5) to (B.7), where the new coefficients now add to 1:[10]

$$c_t = \frac{f_t}{f_t + i_t} * c_{t-1} + \frac{i_t}{f_t + i_t} * g_t \quad \text{(B.7)}$$

The equations governing a standard GRU are:

$$r_t = \sigma(W_r[h_{t-1}, w_t] + b_r) \quad \text{(B.8)}$$
$$z_t = \sigma(W_z[h_{t-1}, w_t] + b_z) \quad \text{(B.9)}$$
$$\tilde{h} = \tanh(W_x[r_t * h_{t-1}, w_t] + b_x) \quad \text{(B.10)}$$
$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h} \quad \text{(B.11)}$$

The GRU's hidden state is squashed because its update gate z merges the functions of the input and forget gates (i and f) of the LSTM (cf. Equations (B.5) and (B.11)). As a result, the input and forget weights are tied in the GRU but not the LSTM. To create a non-squashed GRU, we added an input gate i and changed the hidden state update (Equation (B.11)) to Equation (B.13) to make z act solely as a forget gate:

$$i_t = \sigma(W_i[h_{t-1}, w_t] + b_i) \quad \text{(B.12)}$$
$$h_t = z_t * h_{t-1} + i_t * \tilde{h} \quad \text{(B.13)}$$

[10] We modified the structure of the gates rather than adding a squashing nonlinearity to avoid vanishing gradients.

References

Ben Ambridge, Caroline F. Rowland, and Julian M. Pine. 2008. Is structure dependence an innate constraint? New experimental evidence from children's complex-question production. *Cognitive Science*, 32(1):222–255.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations*.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242.

Matthew M. Botvinick and David C. Plaut. 2006. Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, 113(2):201.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945. Association for Computational Linguistics.

Xinyun Chen, Chang Liu, and Dawn Song. 2018. Tree-to-tree neural networks for program translation. In *Advances in Neural Information Processing Systems*, pages 2547–2557.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, MIT Press, Cambridge, MA.

Noam Chomsky. 1980. Rules and representations. *Behavioral and Brain Sciences*, 3(1):1–15.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 577–585. MIT Press.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Alexander Clark and Rémi Eyraud. 2007. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8(Aug):1725–1745.

Alexander Clark and Shalom Lappin. 2010. *Linguistic Nativism and the Poverty of the Stimulus*, John Wiley & Sons.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM.

Stephen Crain and Mineharu Nakayama. 1987. Structure dependence in grammar formation. *Language*, pages 522–543.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

Chris Dyer, Gábor Melis, and Phil Blunsom. 2019. A critical analysis of biased parsers in unsupervised parsing. *arXiv:1909.09428v1 preprint arXiv:1909.09428*.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Jeffrey L. Elman, Elizabeth A. Bates, Mark H. Johnson, Annette Karmiloff-Smith, Domenico Parisi, and Kim Plunkett. 1998. *Rethinking innateness: A connectionist perspective on development*. MIT press.

Émile Enguehard, Yoav Goldberg, and Tal Linzen. 2017. Exploring the syntactic abilities of RNNs with multi-task learning. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 3–14, Vancouver, Canada. Association for Computational Linguistics.

Hartmut Fitz and Franklin Chang. 2017. Meaningful questions: The acquisition of auxiliary inversion in a connectionist model of sentence production. *Cognition*, 166:225–250.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71.

Robert Frank and Donald Mathis. 2007. Transformational networks. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*. Cognitive Science Society.

Lila R. Gleitman and Henry Gleitman. 1992. A picture is worth a thousand words, but that's the problem: The role of syntax in vocabulary acquisition. *Current Directions in Psychological Science*, 1(1):31–35.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv:1410.5401v2 preprint arXiv:1410.5401*.

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

Xuân-Nga Cao Kam, Iglika Stoyneshka, Lidiya Tornyova, Janet D. Fodor, and William G. Sakas. 2008. Bigrams and the richness of the stimulus. *Cognitive Science*, 32(4):771–787.

Christo Kirov and Robert Frank. 2012. Processing of nested and cross-serial dependencies: An automaton perspective on SRN behaviour. *Connection Science*, 24(1):1–24.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888.

Barbara Landau, Linda B. Smith, and Susan S. Jones. 1988. The importance of shape in early

lexical learning. *Cognitive Development*, 3(3): 299–321.

Stephen Laurence and Eric Margolis. 2001. The poverty of the stimulus argument. *The British Journal for the Philosophy of Science*, 52(2): 217–276.

Julie Anne Legate and Charles D. Yang. 2002. Empirical re-assessment of stimulus poverty arguments. *The Linguistic Review*, 18(1–2):151–162.

John D. Lewis and Jeffrey L. Elman. 2001. Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In *Proceedings of the 26th Annual Conference on Language Development*.

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.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Lisbon, Portugal. 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 Conference of the Cognitive Science Society*, pages 2093–2098. Madison, WI.

R. Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2019. RNNs implicitly implement tensor-product representations. In *Proceedings of the 2019 International Conference on Learning Representations*.

James L. Morgan and Katherine Demuth. 1996. *Signal to syntax: Bootstrapping from speech to grammar in early acquisition*. Psychology Press.

Amy Perfors, Joshua B. Tenenbaum, and Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338.

Steven Pinker. 1996. *Language learnability and language development, with new commentary by the author*, volume 7, Harvard University Press.

Geoffrey K. Pullum and Barbara C. Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 18(1–2): 9–50.

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.

Florencia Reali and Morten H. Christiansen. 2005. Uncovering the richness of the stimulus: Structure dependence and indirect statistical evidence. *Cognitive Science*, 29(6): 1007–1028.

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

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019. Can you tell me how to get past Sesame Street? Sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,

pages 4465–4476. Florence, Italy. Association for Computational Linguistics.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745. Association for Computational Linguistics.