Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks

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

Robert Frank (bob.frank@yale.edu)
Department of Linguistics, Yale University

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

Abstract

Syntactic rules in natural language typically need to make reference to hierarchical sentence structure. However, the simple examples that language learners receive are often equally compatible with linear rules. Children consistently ignore these linear explanations and settle instead on the correct hierarchical one. This fact has motivated the proposal that the learner's hypothesis space is constrained to include only hierarchical rules. We examine this proposal using recurrent neural networks (RNNs), which are not constrained in such a way. We simulate the acquisition of question formation, a hierarchical transformation, in a fragment of English. We find that some RNN architectures tend to learn the hierarchical rule, suggesting that hierarchical cues within the language, combined with the implicit architectural biases inherent in certain RNNs, may be sufficient to induce hierarchical generalizations. The likelihood of acquiring the hierarchical generalization increased when the language included an additional cue to hierarchy in the form of subject-verb agreement, underscoring the role of cues to hierarchy in the learner's input.

Keywords: learning bias; poverty of the stimulus; recurrent neural networks

Introduction

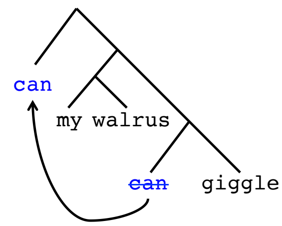
Speakers of a language can generalize from finite linguistic experience to sentences they have never heard or produced before. Although there are many possible ways to generalize from a set of sentences, language learners consistently choose certain generalizations over others. In the syntactic domain, learners typically learn generalizations that appeal to hierarchical structures rather than linear order. An influential explanation for this fact is that learners never entertain hypotheses based on linear order: they are innately constrained to assume that syntactic rules are structure-sensitive (Chomsky, 1980).

To test whether a structure-sensitivity constraint is necessary to account for the generalizations that human language learners make, we use recurrent neural networks (RNNs), which are not equipped with such an explicit pre-existing hierarchical constraint.[1] We simulate the acquisition of English **subject-auxiliary inversion**, the transformation that turns a declarative statement such as (1a) into a question such as (1b):

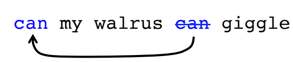
(1) a. My walrus **can** giggle.
b. **Can** my walrus giggle?

[1] In fact, RNNs are not just capable of using non-hierarchical structures but in fact appear to be biased in favor of linear structures over hierarchical ones (Christiansen & Chater, 1999).

At least two rules could generate (1b) from (1a):



Hierarchical rule: Move the main verb's auxiliary to the front of the sentence.



Linear rule: Move the linearly first auxiliary to the front of the sentence.

While both rules account for common cases such as (1), they make different predictions for complex sentences such as (2):

(2) My walrus that **will** eat **can** giggle.

Specifically, the hierarchical rule predicts the correct question (3a), while the linear rule predicts the incorrect question (3b):

(3) a. **Can** my walrus that **will** eat ___ giggle?
b. ***Will** my walrus that ___ eat **can** giggle?

Although such examples disambiguate the two hypotheses, Chomsky (1971) argues that they are highly infrequent, and thus children may never encounter them. Without these critical examples, according to Chomsky, children can only acquire the hierarchical rule by drawing on an innate constraint stipulating that syntactic rules must appeal to hierarchy.

This argument, known as the **argument from the poverty of the stimulus** (Chomsky, 1980), has been challenged in a number of ways. Some have disputed the assumption that children never encounter critical cases such as (3a) (Pullum & Scholz, 2002). Others have questioned the assumption that an explicit hierarchical constraint is necessary for hierarchical generalization. One such approach has been to argue that the hierarchical rule can fall out of weaker or non-syntactic structural biases. For example, Perfors, Tenenbaum, and Regier (2011) showed that a learner whose task is to choose between an innately available hierarchical representation and an innately available linear representation will choose the hierarchical one; and Fitz and Chang (2017) argued that the hierarchical structure of questions is rooted in innately available structured semantic representations.

	Training set, test set	Generalization set
	IDENT	QUEST
No RC	*Input:* the newt can confuse my yak by the zebra . *Output:* the newt can confuse my yak by the zebra .	*Input:* the newt can confuse my yak by the zebra . *Output:* can the newt confuse my yak by the zebra ?
RC on object	*Input:* the newt can confuse my yak who will sleep . *Output:* the newt can confuse my yak who will sleep .	*Input:* the newt can confuse my yak who will sleep . *Output:* can the newt confuse my yak who will sleep ?
RC on subject	*Input:* the newt who will sleep can confuse my yak . *Output:* the newt who will sleep can confuse my yak .	*Input:* the newt who will sleep can confuse my yak . *Output:* can the newt who will sleep confuse my yak ?

Table 1: Examples for each combination of a sentence type and a task. RC stands for "relative clause."

A second approach has dispensed with pre-existing structural representations altogether. Lewis and Elman (2001) argued that an RNN trained to predict the next word can learn which questions are well formed, but this conclusion was convincingly called into question by Kam, Stoyneshka, Tornyova, Fodor, and Sakas (2008). The most immediate precursor to our work is Frank and Mathis (2007). Like Lewis and Elman, they used RNNs, but instead of modeling the well-formedness of the question alone, they followed the traditional framework of transformational grammar in modeling the generation of a question from a declarative sentence.[2] Their results were difficult to interpret because the network's generalization behavior depended heavily on the identity of the auxiliaries in the input sentence, and neither the linear hypothesis nor the hierarchical hypothesis predict such lexically dependent behavior. We significantly expand on their experiments, taking advantage of recent technological and architectural advances in RNNs that have shown promise in the acquisition of syntax (Linzen, Dupoux, & Goldberg, 2016).

To anticipate our results, of the six RNN architectures we explored, one of the architectures consistently learned a hierarchical generalization for question formation. This suggests that a learner's preference for hierarchy may arise from the hierarchical properties of the input, coupled with biases implicit in the network's computational architecture and learning procedure, without the need for pre-existing hierarchical constraints in the learner. We provide further evidence for the role of the hierarchical properties of the input by showing that adding syntactic agreement to the input increased the probability that a network would make hierarchical generalizations.

Experimental setup

Languages

The networks were trained on two fragments of English, each consisting of a subset of all possible declarative sentences and questions.[3] We refer to the first fragment as the **no-agreement language**. Examples of declarative sentences in this language are given in (4):

(4) a. the walrus can giggle .
b. the yak could amuse your quails by my raven .
c. the walruses that the newt will confuse can high_five your peacocks .

Each noun phrase in the language had at most one modifier, either a relative clause or a prepositional phrase. Relative clauses were never embedded inside other relative clauses. Every verb was associated with one of the auxiliary verbs *can*, *could*, *will*, and *would*. Since such modals do not show agreement, any noun, whether singular or plural, was allowed to appear with any auxiliary.

The second fragment, the **agreement language**, was identical to the no-agreement language, except that the auxiliaries in this language were *do*, *don't*, *does*, and *doesn't*. Subjects in this language agreed with the auxiliaries of their verbs: singular subjects appeared with *does* or *doesn't*, while plural subjects appeared with *do* or *don't*. Examples of declarative sentences in the agreement language are given in (5):

(5) a. the walrus does giggle .
b. the yak doesn't amuse your quails by my raven .
c. the walruses that the newt does confuse do high_five your peacocks .

Both languages reused structural units; for example, the same prepositional phrases could modify both subject and object nouns. Such shared structure served as a possible cue to hierarchy because it is more efficiently represented in a hierarchical grammar than a linear one. Subject-verb agreement in the agreement language provided an additional cue to hierarchy; in (5c), for example, *do* agrees with its hierarchically-determined plural subject of *walruses* even though the singular noun *newt* is linearly closer to it. We therefore predict that hierarchical generalizations will be more likely with the agreement language than the no-agreement language.

Tasks

The networks were trained to perform two tasks: identity (returning the input sentence unchanged) and question formation. The task to be performed was indicated by a token at the

[2] This is a simplification—a more psychologically plausible assumption would be that questions are generated from a semantic representation shared with the declarative sentence (Fitz & Chang, 2017).

[3] The vocabulary of the fragments consisted of 66 words. The full context-free grammar characterizing the fragments, along with statistics about the generated sentences, can be found in the supplementary materials at https://arxiv.org/abs/1802.09091.

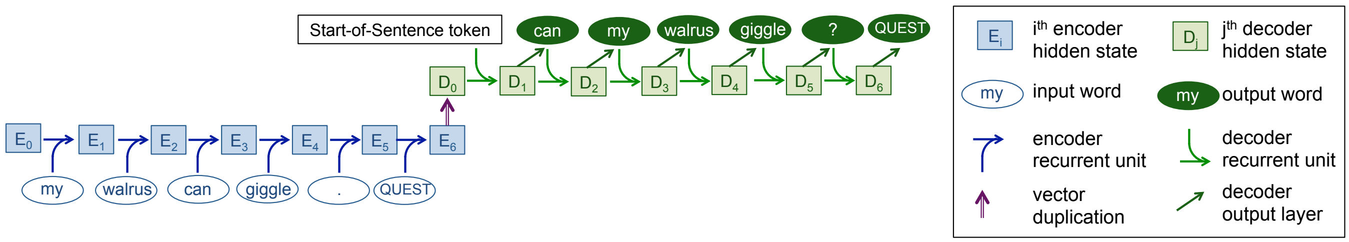


Figure 1: Basic sequence-to-sequence neural network without attention.

end of the sentence—either *IDENT* for identity or *QUEST* for question formation. *IDENT* and *QUEST* served as end-of-sequence tokens in both the input and output.

Table 1 provides examples of these tasks on each of the three types of sentences in the languages: sentences without relative clauses, sentences with a relative clause on the object, and sentences with a relative clause on the subject. During training we withheld the question formation task for sentences with a relative clause on the subject (the shaded cell in Table 1); these are the only cases that directly disambiguate the linear and hierarchical hypotheses. The identity task was included in the training setup to familiarize the networks with the critical sentence type withheld from the question task; without such exposure, the networks could be justified in concluding that subjects cannot be modified by relative clauses, making it difficult to test such sentences.

Evaluation

We used two sets of sentences for evaluation, a test set and a generalization set. The test set consisted of novel sentences from the five non-withheld cases in Table 1. It was used to assess how well a network had learned the patterns in its training set. The generalization set consisted of sentences from the withheld case (the question formation task for sentences with relative clauses on their subjects). This set was used to assess how the networks generalized to sentence types from which they had not formed questions during training. The test and generalization set both contained 10,000 unique sentences and the training set contained 120,000 unique sentences.

Architectures

Here we give a very brief bird's-eye view of our architectures. For a more precise description, including our hyperparameter values, see the supplementary materials.

For all experiments we used the sequence-to-sequence model (Botvinick & Plaut, 2006; Sutskever, Vinyals, & Le, 2014) illustrated in Figure 1. This network has two subcomponents called the **encoder** and the **decoder**, both of which are RNNs. The encoder processes the input sentence one word at a time to create a single vector representing the entire input sentence. The decoder then receives this vector (called the **encoding**) and, based on it, outputs one word at a time until it generates a special end-of-sequence token.

The encoder and decoder each possess a component called a **recurrent unit** which governs how information flows from one time step to the next. We tested three types of recurrent units: a simple recurrent network (SRN) (Elman, 1990), a gated recurrent unit (GRU) (Cho et al., 2014), and long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997). For each type of recurrent unit, we experimented with adding **attention** to the decoder (Bahdanau, Cho, & Bengio, 2015); attention is a mechanism which gives the decoder access to intermediate steps of the encoding process. For each pair of an architecture and a language, we trained 100 networks with different random initializations, for a total of 1200 networks.

Results

Test set

For the test set, all six architectures except the vanilla SRN (i.e., the SRN without attention) produced over 94% of the output sentences exactly correctly (accuracy was averaged across 100 trained networks for each architecture). The highest accuracy was 99.9% for the LSTM without attention. Using a more lenient evaluation criterion whereby the network was not penalized for replacing a word with another word of the same part of speech, the accuracy of the SRN without attention increased from 0.1% to 81%, suggesting that its main source of error was a tendency to replace words with other words of the same lexical category. This tendency is a known deficiency of SRNs (Frank & Mathis, 2007) and does not bear on our main concern of the networks' syntactic representations. Setting aside these lexical concerns, then, we conclude that all architectures were able to learn the language.

Generalization set

On the generalization set, the networks were rarely able to correctly produce the full question – only about 13% of the questions were exactly correct in the best-performing architecture (LSTM with attention). However, getting the output exactly correct is a demanding metric; the full-question accuracy can be affected by a number of errors that are not directly related to the research question of whether the network preferred a linear or hierarchical rule. Such errors include repeating or omitting words or confusing similar words. To abstract away from such extraneous errors, for the generalization set we focus on accuracy at the first word of the output. Because all examples in the generalization set involve question formation, this word is always the auxiliary that is moved to form the question, and the identity of this auxiliary

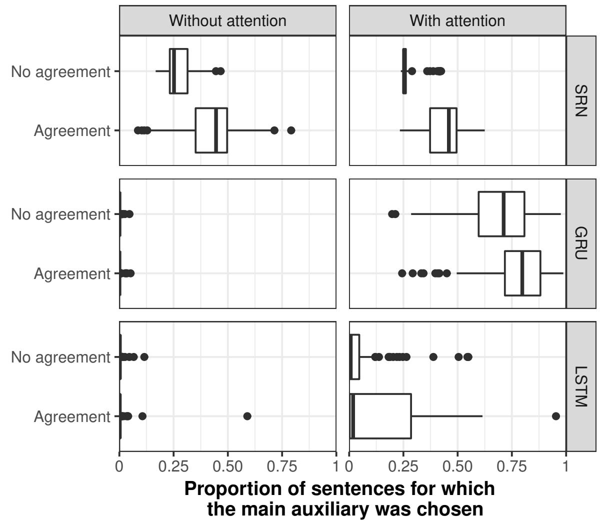


Figure 2: Accuracy of auxiliary prediction for questions of the withheld type (with a relative clause on the subject).

is enough to differentiate the hypotheses. For example, if the input is *my yak who the seal can amuse will giggle . QUEST*, a hierarchically-generalizing network would choose *will* as the first word of the output, while a linearly-generalizing network would choose *can*. This analysis only disambiguates the hypotheses if the two possible auxiliaries are different, so we only considered sentences where that was the case. For the agreement language, we made the further stipulation that both auxiliaries must agree with the subject so that the correct auxiliary could not be determined based on agreement alone.

Figure 2 gives the accuracies on this metric across the six architectures for the two different languages (individual points represent different initializations). We draw three conclusions from this figure:

1. Agreement leads to more robust hierarchical generalization: All six architectures were significantly more likely ($p < 0.01$) to choose the main auxiliary when trained on the agreement language than the no-agreement language. In other words, adding hierarchical cues to the input increased the chance of learning the hierarchical generalization.

2. Initialization matters: For each architecture, accuracy often varied considerably across random initializations. This fact suggests that the architectural bias is not strong enough to reliably lead the networks to settle on the hierarchical generalization, even in GRUs with attention. From a methodological perspective, this observation highlights the importance of examining many initializations of the network before drawing qualitative conclusions about an architecture (in a particularly striking example, though the accuracy of most LSTMs with attention was low, there was one with near-perfect accuracy).

3. Different architectures perform qualitatively differently: Of the six architectures, only the GRU with attention showed a strong preference for choosing the main auxiliary instead of the linearly first auxiliary. By contrast, the vanilla GRU chose the first auxiliary nearly 100% of the time. In this case, then, attention made a qualitative difference for the generalization that was acquired. By contrast, for both LSTM architectures, most random initializations led to networks that chose the first auxiliary nearly 100% of the time. Both SRN architectures showed little preference for either the main auxiliary or the linearly first auxiliary; in fact the SRNs often chose an auxiliary that was not even in the input sentence, whereas the GRUs and LSTMs almost always chose one of the auxiliaries in the input. In the next section, we take some preliminary steps toward exploring why the architectures behaved in qualitatively different ways.

Analysis of sentence encodings

A plausible hypothesis about the differences between networks is that linearly-generalizing networks used representations that contained linearly-relevant information whereas hierarchically-generalizing networks used representations that contained hierarchically-relevant information. To test this hypothesis, we analyzed the final hidden state of the encoder (E_6 in Figure 1), which we will refer to as the encoding of the sentence. In architectures without attention, this is the only information that the decoder has about the sentence; architectures with attention can use the intermediate encodings of sentence prefixes as well. We analyze the amount of information that these encodings contain about three properties of the input sentence: its main auxiliary, its fourth word, and the head noun of the subject (which, in the simple languages we used, was always the sentence's second word). Examples are shown in Table 2.

Main auxiliary: The main auxiliary of a sentence can appear in many different linear positions but has a consistent hierarchical position. Therefore, a network whose encodings can be used to identify sentences' main auxiliaries must contain some hierarchical information.

Fourth word: The fourth word of a sentence has a consistent role in a linear representation but not in a hierarchical one: the fourth word could be the main verb, the determiner on a prepositional object, or the auxiliary verb inside a subject relative clause. Therefore, a network whose encodings can be used to identify each sentence's fourth word must contain some information about linear order.

Subject noun/second word: The head noun of the subject is always the second word of the sentence in our languages. Thus, this word can be reliably identified either from a linear representation (as the second word) or from a hierarchical representation (as the subject noun).

Analysis: For each trained network, we trained three linear classifiers, one for each of these three properties of the sentence. Each classifier was trained to predict the word that filled the relevant role—main auxiliary, fourth word or subject noun/second word—from the final hidden state of the encoder. Each classifier's output layer had a dimensionality equal to the number of possible classes for that classifier's task: 4 for the main auxiliary, 28 for the fourth word, or 26 for the subject noun. The classifiers were trained on a training set

Main auxiliary	Fourth word	Subject noun
my unicorns **would** laugh .	my unicorns would **laugh** .	my **unicorns** would laugh .
my quail with her yak **will** read .	my quail with **her** yak will read .	my **quail** with her yak will read .
his newt who can giggle **could** swim .	his newt who **can** giggle could swim .	his **newt** who can giggle could swim .

Table 2: Examples of the entities identified by the linear classifiers.

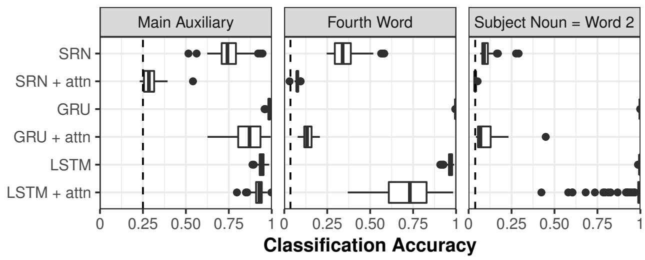


Figure 3: Linear classifier results. Dotted lines indicate chance performance.

and tested on a withheld test set (see the supplementary materials for details). Figure 3 shows the classification results on the test set.

Classifiers trained to predict the main auxiliary from the encodings produced by the SRNs with attention performed only slightly better than chance; this might explain why the SRNs with attention generalized poorly to the withheld sentence type in the question formation task. Similar classifiers trained on encodings from the other architectures did well at this task. Since the identity of the main auxiliary is the only information required to perform well on our evaluation of the networks' performance on the generalization set based on the first word produced, these results suggest that the differences in performance stem not from inability to identify the main auxiliary but rather from a misinterpretation of the task as requiring fronting of the linearly first auxiliary.

We now consider the fourth word and subject noun classifiers. The classifiers trained on the encodings from both types of LSTMs as well as the GRUs without attention performed well at both tasks. Crucially, the classifiers trained on the encodings from the GRU with attention did poorly on these tasks. Recall that the main auxiliary could be successfully decoded from the encodings of this architecture. The GRU with attention therefore appears to use its encoding only for information that could not be straightforwardly obtained from linear order, such as the main auxiliary, rather than information that could be obtained from linear order even if, like the subject head noun, that information was hierarchically relevant. On the other hand, the fact that the GRU without attention and both LSTM architectures performed very well at all three tasks suggests that they used their encodings for both linear and hierarchical information. Thus, perhaps the better generalization ability of the GRU with attention arises not from a better ability to encode relevant hierarchical information—all four LSTM and GRU architectures have that ability—but rather from an ability to ignore linear information (Frank & Mathis, 2007).

	Prepose 1st	Prepose 2nd	Prepose other
Delete 1st	7%	24%	4%
Delete 2nd	0%	3%	0%
Delete none	4%	21%	2%

Table 3: Analysis of output question types based on which auxiliary has been deleted (if any) and which auxiliary has been placed at the start of the sentence. Each number is the percent of GRU + attention outputs across all 100 random initializations that fit that category (the total sums to 65% because only 65% of the questions produced by the networks could be analyzed in that way). *1st* and *2nd* refer to the first and second auxiliaries in the input.

Comparing RNN Mistakes with Human Mistakes

We now return to the full questions produced by our networks and compare the networks' errors to the types of errors that humans make when acquiring English (Crain & Nakayama, 1987). We restrict ourselves to the GRU with attention networks as those were the networks that generally produced the correct auxiliary (see Figure 2).

Subject-auxiliary inversion can be decomposed into two subtasks: placing an auxiliary at the start of the sentence and deleting an auxiliary within the sentence. Only 65% of the outputs that the 100 networks collectively produced could be interpreted as having been formed by inserting an auxiliary before the sentence and deleting zero or one of the auxiliaries in the sentence. Table 3 breaks down those results based on which auxiliary was preposed and which (if any) was deleted.[4]

Two error types are by far the most common. In the first type, the network preposed the second auxiliary but did not delete either of the auxiliaries (*could his newt who can giggle could swim* from *his newt who can giggle could swim*). This error type is common among English-learning children (Crain & Nakayama, 1987) and is compatible with hierarchical generalization. In the other frequent error type, the network deleted the first auxiliary and preposed the second; for example, it might generate *could his newt who giggle could swim* from *his newt who can giggle could swim.* Such errors were never observed by Crain and Nakayama (1987) and are incompatible with a hierarchical generalization. In other

[4]See the supplementary materials for examples of the remaining 35% of outputs.

words, though the networks' common error types overlapped with the common error types for humans, the networks also frequently made some mistakes that humans never would.

Conclusions and Future Work

Learners of English acquire the correct hierarchical rule for forming questions even though there are few to no examples in their input that explicitly distinguish this rule from the linear one. This fact has been taken to suggest that learners must be innately constrained to consider only hierarchical syntactic rules. We have investigated whether a learner without such a constraint can learn the hierarchical generalization without the critical disambiguating examples. Based on the behavior of one of the architectures we examined (GRU with attention), the answer to this question appears to be yes. The hierarchical behavior of this non-hierarchically-constrained architecture plausibly arose from the influence of hierarchical cues in the input, a conclusion supported by the fact that the additional hierarchical cue of agreement increased the likelihood that a network would induce hierarchical generalizations.

Our argument has focused on a strong version of the poverty of the stimulus argument which claims that language learners require a hierarchical *constraint*. However, there remains a milder version which only claims that a hierarchical *bias* is necessary. This version of the argument is difficult to assess using RNNs because, while RNNs must possess some biases (Mitchell, 1980; Marcus, 2018), the nature of these biases—which likely arise both from the network architecture and from the learning algorithm—is currently poorly understood. However, given the linear way in which they process inputs, it is plausible that all six architectures we used had a bias toward linear order but that the GRU with attention was the only one that overcame this linear bias sufficiently to generalize hierarchically. It is not clear why it was the only architecture to do so; we intend to examine the differences in behavior between the recurrent units in future work.

Two caveats are in order. First, our results only cover restricted fragments of English and may not generalize to the linguistic input that human language learners encounter. In future work, we will replace our artificial languages with a corpus of child-directed speech. Second, even if our findings do generalize to realistic language, we would only be able to conclude that it is *possible* to solve the task without a hierarchical constraint; humans certainly could have such an innate constraint despite it being unnecessary for this particular task.

Acknowledgments

Our experiments were conducted using the resources of the Maryland Advanced Research Computing Center (MARCC). We thank Joe Pater, Paul Smolensky, and the JHU Computational Psycholinguistics group for helpful comments.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR.*

Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, *113*(2), 201–233.

Cho, K., Van Merrinboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of EMNLP.*

Chomsky, N. (1971). *Problems of knowledge and freedom.* New York: Pantheon.

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

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*(2), 157–205.

Crain, S., & Nakayama, M. (1987). Structure dependence in grammar formation. *Language*, 522–543.

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

Fitz, H., & Chang, F. (2017). Meaningful questions: The acquisition of auxiliary inversion in a connectionist model of sentence production. *Cognition*, *166*, 225–250.

Frank, R., & Mathis, D. (2007). Transformational networks. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Kam, X.-N. C., Stoyneshka, I., Tornyova, L., Fodor, J. D., & Sakas, W. G. (2008). Bigrams and the richness of the stimulus. *Cognitive Science*, *32*(4), 771–787.

Lewis, J. D., & Elman, J. L. (2001). Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In *Proceedings of BUCLD.*

Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, *4*, 521–535.

Marcus, G. (2018). Innateness, AlphaZero, and artificial intelligence. *arXiv preprint arXiv:1801.05667*.

Mitchell, T. M. (1980). *The need for biases in learning generalizations* (Tech. Rep.). Rutgers University.

Perfors, A., Tenenbaum, J. B., & Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, *118*(3), 306–338.

Pullum, G. K., & Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, *18*(1-2), 9–50.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of NIPS.*