ANNUAL REVIEWS

Annual Review of Linguistics

Syntactic Structure from Deep Learning

Tal Linzen[1] and Marco Baroni[2,3,4]

[1] Department of Linguistics and Center for Data Science, New York University, New York, NY 10003, USA; email: linzen@nyu.edu

[2] Facebook AI Research, Paris 75002, France; email: mbaroni@fb.com

[3] Catalan Institute for Research and Advanced Studies, Barcelona 08010, Spain

[4] Departament de Traducció i Ciències del Llenguatge, Universitat Pompeu Fabra, Barcelona 08018, Spain

Annu. Rev. Linguist. 2021. 7:2.1–2.18

The *Annual Review of Linguistics* is online at linguistics.annualreviews.org

https://doi.org/10.1146/annurev-linguistics-032020-051035

Keywords

deep learning, syntax, nature versus nurture, probing linguistic knowledge

Abstract

Modern deep neural networks achieve impressive performance in engineering applications that require extensive linguistic skills, such as machine translation. This success has sparked interest in probing whether these models are inducing human-like grammatical knowledge from the raw data they are exposed to and, consequently, whether they can shed new light on longstanding debates concerning the innate structure necessary for language acquisition. In this article, we survey representative studies of the syntactic abilities of deep networks and discuss the broader implications that this work has for theoretical linguistics.

Review in Advance first posted on September 21, 2020. (Changes may still occur before final publication.)

1. INTRODUCTION

In the last decade, artificial neural networks, which have been rebranded as deep learning (LeCun et al. 2015), have made an astounding comeback in a range of technological applications. Among those applications are natural language processing (NLP) tasks ranging from machine translation (Edunov et al. 2018) to reading comprehension (Cui et al. 2017). From a linguist's perspective, the applied success of deep neural networks (DNNs) is striking because, unlike the systems that were popular in NLP a decade ago (Jurafsky & Martin 2008), DNNs' input data and architectures are not based on the symbolic representations familiar from linguistics, such as parse trees or logical formulas. Instead, DNNs learn to encode words and sentences as vectors (sequences of real numbers); these vectors, which do not bear a transparent relationship to classic linguistic structures, are then transformed through a series of simple arithmetic operations to produce the network's output. Thus, any grammatical competence acquired by standard DNNs derives from exposure to large amounts of raw text as well as generic architectural features that have little to do with those that many linguists have deemed necessary for language acquisition, such as a preference for rules that are based on hierarchical tree structure (Chomsky 1986).

Ostensibly, the success of deep learning invites a reassessment of classic arguments that language acquisition necessitates rich innate structure. But measures of practical success do not directly engage with the evidence that has motivated structural assumptions in linguistics: Whereas success in applications rests primarily on the system's ability to handle common constructions, it is often the rare constructions that are the most informative ones from a theoretical standpoint. In this review, we focus on work that directly evaluates DNNs' syntactic knowledge and use paradigms from linguistics and psycholinguistics that highlight such theoretically significant cases. After a brief introduction to deep learning for language processing (Section 2), we review work that has applied (psycho)linguistic analysis methods to English subject–verb number agreement, filler–gap dependencies, and other syntactic phenomena (Sections 3 and 4); this body of work suggests that contemporary DNNs can learn a surprising amount about syntax but that this ability falls short of human competence. We then briefly survey work that aims to illuminate the internal processes by which DNNs accomplish their grammatical behavior (Section 5). Finally, we discuss our view regarding the implications of this body of work for linguistic theory (Section 6).

2. DEEP LEARNING FOR LANGUAGE PROCESSING

This section provides a very short overview of artificial neural networks as applied to language processing at the word and sentence level. For a contemporary book-length introduction to neural networks for language processing, readers are referred to Goldberg (2017); for a less technical introduction to artificial neural networks for cognitive scientists, which does not include a treatment of language processing, we refer readers to Elman et al. (1998).

Artificial neural networks are mathematical objects that compute functions from one sequence of real numbers to another sequence. They do so using large collections of simple computation units ("neurons"). Each of these units calculates a weighted average of its inputs; this weighted average is then passed as an input to a simple nonlinear function, such as the sigmoid: $\sigma(a) = 1/(1+e^{-a})$. In other words, the function computed by each unit is $\sigma(w_1x_1 + \cdots + w_nx_n)$, where $w_1, \ldots, w_n$ are the weights, and $x_1, \ldots, x_n$ are the inputs. Although the computation performed by each unit is very simple, by arraying a large number of units in layers—such that the output of the units in one layer serves as the input to the units in the following layer—much more complex functions can be computed (theoretically, all functions can be approximated to arbitrary precision; see Leshno et al. 1993). Incidentally, the presence of multiple layers is what makes the networks "deep."

The network's weights are not set by the designer of the system but, instead, are learned from examples. Each such training example consists of an input $\mathbf{x}_i$ and an expected output y_i. The training procedure starts from a random set of weights. The network then iterates through each input example and computes the output $\hat{y}_i$ based on the current weights. This output is then compared with the expected output y_i, and the weights are adjusted by a small amount such that the next time the DNN receives $\mathbf{x}_i$ as its input, the discrepancy between $\hat{y}_i$ and y_i will be smaller (this process is referred to as gradient descent).

Neural networks compute numerical functions. For them to process language, each input and output word needs to be encoded as a vector (a sequence of real numbers). In principle, a word could be encoded by a vector whose size is equal to the size of the vocabulary and that has zeros everywhere except for one component that indicates the identity of the word (a so-called one-hot vector); in practice, however, words are typically encoded by vectors that are much smaller and denser (with nonzero values in most components). Such distributed representations (known as word embeddings) make it possible to assign similar vectors to words that occur in similar contexts or have similar meanings [e.g., (1, 2.5, 3) for *table* and (1.2, 2.5, 2.8) for *desk*, but (2.1, −3, 4) for *dog*]. Rather than being set by the designer of the system, these word embeddings are learned using gradient descent, just like the network's weights.

Word embeddings provide a mechanism for encoding individual words, but additional machinery is needed to process sequences of words. One such approach is to use recurrent neural networks (RNNs). An RNN processes the sentence from left to right, maintaining a single vector $\mathbf{h}_t$, the so-called hidden state, which represents the first t words of the sentence. The next hidden state, $\mathbf{h}_{t+1}$, is computed from $\mathbf{h}_t$ and the embedding of the $t + 1$-th word by using standard neural network arithmetic. The hidden state thus acts as a bottleneck: The network does not have access to its earlier hidden states $\mathbf{h}_1, \ldots, \mathbf{h}_{t-1}$ when computing $\mathbf{h}_{t+1}$.

The performance of RNNs can be improved by the addition of gates. Gated RNNs, such as long short-term memory networks (LSTMs) (Hochreiter & Schmidhuber 1997) and gated recurrent units (GRUs) (Cho et al. 2014), possess a mechanism that allows them to better control the extent to which information in their hidden state is updated after each word. At least in principle, gating enables the network to better track dependencies that span a large number of words. LSTMs have become the de facto standard RNN variant in NLP, and most of the studies we review here use this architecture.

Attention, another important innovation, relaxes the single-hidden-state bottleneck and allows the network to take all of its previous hidden states into account when computing the next state (Bahdanau et al. 2015). The dynamics of both gating and attention are not hard-coded but, rather, controlled by weights learned during the training phase.

When the network has access to a large window of previous states through attention, the recurrence mechanism, which computes the new hidden state $\mathbf{h}_{t+1}$ based only on the most recent hidden state $\mathbf{h}_t$, becomes largely redundant. Consequently, some state-of-the-art sequence processing architectures, such as the transformer (Vaswani et al. 2017), dispense with recurrence altogether and rely on attention only to carry information across time. Using specialized hardware, transformers can be trained effectively on very large corpora, and they have become very common in NLP systems such as BERT (Devlin et al. 2019); however, a clear picture of the differences between their linguistic abilities and those of RNNs, especially when both are exposed to the same amount of training data, has yet to emerge (Y. Goldberg 2019, Rogers et al. 2020, Tran et al. 2018).

DNNs for language processing are used in three common settings. When used as a classifier, the network outputs a discrete label for the sequence; for example, a binary acceptability judgment system might output either "acceptable" or "unacceptable." This is a supervised setup, in which the network is trained on a corpus of example sentences annotated for acceptability.

When used in the setting referred to in computational linguistics as a language model, the network receives the first *n* words of a sentence and assigns a probability to each of the words that could come up next. The network is trained to maximize the probability of the word that actually occurs next in the sentence as it appears in the corpus. For example, given the first five words of the sentence *The children went outside to play*, the training objective would be to assign the largest possible probability to the final word, *play*. This setup is unsupervised, in the sense that all we need to train the network is a text corpus, without any annotation.

Finally, in the sequence-to-sequence (seq2seq) setting, the network is expected to generate an output sequence in response to an input sequence—for example, when translating between languages (Sutskever et al. 2014). This setting is configured by chaining two RNNs such that the last state of the encoder RNN serves as input to the decoder RNN. Attention can be used to allow the decoder to base its decisions on all of the encoders' hidden states rather than the last one only. The seq2seq setting requires pairs of input and output sequences as training materials (e.g., sentences in a source language and the corresponding translations in a target language), but it does not require any further annotation.

3. LONG-DISTANCE AGREEMENT

Much of the initial analytical work on the syntactic abilities of DNNs centered on long-distance agreement between subject and verb or between other elements in a syntactic dependency. Subject–verb agreement is a paradigmatic example of the observation that words in sentences are organized according to "structures, not strings" (Everaert et al. 2015): The notion of the subject makes crucial reference to the abstract structure of the sentence rather than to the linear sequence of the words it comprises. A DNN's ability to capture subject–verb agreement can be evaluated in a straightforward manner using the number prediction task (Linzen et al. 2016). In this setting, the DNN is exposed to a sentence prefix such as example 1, in which the word following the prefix is expected to be a verb; the DNN is then tasked with predicting whether the verb that is expected to come up next should be in plural or singular form:

(1) The **length** of the *forewings* (is/*are) . . .

The correct form depends on the number of the verb's subject, which, in example 1, is determined by its head *length* rather than by *forewings*. Nouns such as *forewings*, which intervene between the head of the subject and the verb, are referred to as attractors (Bock & Miller 1991). To correctly predict the number of the verb, the DNN must derive an implicit analysis of the structure of the sentence and resist the lure of the proximal but irrelevant attractor.

Linzen and colleagues (2016) exposed an LSTM (see Section 2) to a set of corpus-extracted English sentence prefixes, such as the one in example 1, and trained it to perform the number prediction task (i.e., it was trained in the supervised classification setting, in the terms established in Section 2). When tested on new prefixes that had not been presented to it during training, the network correctly predicted the number of the upcoming verb more than 99% of the time. However, since in the overwhelming majority of English sentences the head of the subject happens to be the most recent noun preceding the verb [as it is, for example, in *the well-known lawyers (are)*], overall accuracy is not directly informative about whether a DNN is able to identify the head of the subject. Crucially, even when it was tested on sentences with attractors, the DNN showed considerable robustness: In sentences with as many as four attractors, number prediction accuracy was still 82%. While the probability of an error increased substantially in the presence of attractors, it was considerably lower than what would be expected if the network was typically misled by attractors.

Extending these results, Bernardy & Lappin (2017) showed that other DNN architectures (GRUs and convolutional networks) can also tackle the number prediction task with considerable success. This finding indicates that Linzen and colleagues' (2016) original result did not crucially depend on specific features of the LSTM architecture. In particular, the convolutional network trained by Bernardy & Lappin completely dispensed with the recurrent mechanism central to LSTMs (Kalchbrenner et al. 2014). Its success therefore points to the generality of the outcome across current DNN models.

In these early studies, the DNNs were trained specifically to predict the number of an upcoming verb and were given explicit feedback about verb number in a large set of sentences. Gulordava et al. (2018) showed that LSTMs can learn a significant amount about long-distance agreement even when trained simply to predict the next word in a corpus without any specific focus on subject–verb agreement or number features (the language modeling setting). To test a trained DNN's ability to compute agreement, the authors exposed it to a sentence prefix and compared the probabilities it assigned, given the prefix, to the singular and plural forms of the upcoming verb. The DNN was considered to have made a correct number prediction if it assigned a higher probability to the contextually appropriate form; for example, after *the length of the forewings*, the probability of *is* was expected to be higher than that of *are*.

Using this methodology, Gulordava and colleagues (2018) showed that LSTMs trained only to predict the next word show high agreement prediction accuracy when tested on sentence prefixes extracted from Wikipedia, across four languages (English, Hebrew, Italian, and Russian), and in dependencies beyond subject–verb agreement, in languages that have them (e.g., adjective–noun agreement). When compared with human subjects, the LSTM's agreement prediction accuracy in Italian (the only language for which this comparison was carried out) was only moderately lower. Finally, Gulordava and colleagues tested the DNN's predictions in "colorless green ideas" (Chomsky 1957) prefixes: grammatically well-formed but highly semantically implausible prefixes, which are constructed by replacing the content words in prefixes from the corpus with other words from the same syntactic category (e.g., *the colorless green ideas near the duck (are/*is)*...). The DNN showed only a mild degradation in performance on these sentences, which suggests that it is capable of computing agreement in the absence of lexical or semantic cues. Overall, this study suggests that training on word prediction alone, without additional syntactic supervision or semantic grounding, can teach networks a substantial amount about long-distance agreement dependencies and the syntactic categories that underlie them (such as the notion of a syntactic subject).

This is not to say that LSTM language models acquire perfect syntactic competence; rather, there is evidence that they rely on simple heuristics. For example, because English does not indicate the end of a relative clause with an explicit marker, LSTMs tend to expect embedded clauses to be relatively short (Linzen & Leonard 2018); they also pay undue attention to the number of the first noun of the sentence, even when it is not the head of the subject (Kuncoro et al. 2018a). In a study using controlled experimental materials instead of evaluation sentences sampled from a corpus, Marvin & Linzen (2018) found that LSTMs performed poorly on some sentence types that are infrequent in corpora, such as nested agreement dependencies across an object relative clause [*The farmer that the parents love (swims/*swim)*]. At the same time, the fact that DNNs perform consistently well across a range of agreement dependencies suggests that they are able to extract certain abstract syntactic generalizations. In Section 5, we discuss work that brings us close to a mechanistic understanding of how LSTMs perform long-distance agreement.

4. OTHER SYNTACTIC PHENOMENA

Research on the syntactic abilities of neural networks quickly expanded beyond agreement to include a variety of other syntactic phenomena. Here, we review a few representative lines of work. For additional examples, we refer the reader to the proceedings of the BlackboxNLP workshops (Linzen et al. 2018, 2019) and of the Society for Computation in Linguistics (**https://scholarworks.umass.edu/scil/**).

Wilcox et al. (2018) tested the sensitivity of LSTM language models to English filler–gap dependencies. As in the study by Gulordava et al. (2018), the DNNs were trained only to predict the next word, without any specific supervision on this construction. In a filler–gap dependency, a *wh*-licensor sets up a prediction for a gap—one of the noun phrases in the embedded clause must be omitted:

(2a) I know that you insulted your aunt yesterday. (no *wh*-licensor, no gap)
(2b) *I know who you insulted your aunt yesterday. (*wh*-licensor, no gap)

According to Wilcox and colleagues' (2018) logic, if the network is sensitive to this constraint, we expect it to be more surprised by *yesterday* in the ungrammatical example 2b than in the grammatical example 2a (we say that the DNN is more surprised by a word if it assigns a lower probability to that word). The DNN's surprise is measured at *yesterday*—instead of at the filled gap *your aunt*, which is arguably the locus of ungrammaticality—because there are contexts in which the sentence can be continued in a grammatical way after this noun phrase (*I know who you insulted your aunt with*___).

It is not only the case that gaps are required after a *wh*-licensor; they are only allowed in the presence of such a licensor. If the DNN is fully sensitive to filler–gap dependencies, then we expect it to be more surprised by *yesterday* in example 3a, where the gap is ungrammatical, than in example 3b:

(3a) *I know that you insulted ____ yesterday. (no *wh*-licensor, gap)
(3b) I know who you insulted ____ yesterday. (*wh*-licensor, gap)

Wilcox et al. (2018) reported that the networks showed the expected pattern. This was the case not only for direct object extraction, as in examples 2 and 3, but also for subject extraction (example 4a) and indirect object extraction (example 4b):

(4a) I know who ____ showed the presentation to the visitors yesterday.
(4b) I know who the businessman showed the presentation to ____ yesterday.

The acceptability of grammatical filler–gap constructions was only marginally affected by the distance between the *wh*-licensor and the gap, in line with studies of human processing. Further, the networks correctly learned that a *wh*-phrase cannot license more than one gap, distinguishing the well-formed examples 5a and 5b from the less natural 5c:

(5a) I know what the lion devoured ___ at sunrise.
(5b) I know what ___ devoured a mouse at sunrise.
(5c) *I know what ___ devoured ___ at sunrise.

In some syntactic configurations that are referred to as islands (Ross 1967), extracting a noun phrase (replacing it with a gap) is ungrammatical. If the networks are sensitive to this constraint, their expectation of a gap should be attenuated in these contexts. Wilcox and colleagues (2018)

Review in Advance first posted on
September 21, 2020. (Changes may
still occur before final publication.)

found this to be the case for some contexts, such as *wh*-islands and adjunct islands. One exception was the Subject Island Constraint, under which a prepositional phrase modifying a noun phrase can only contain a gap if that noun phrase is not the subject:

(6a) I know who the family bought the painting by ___ last year.
(6b) *I know who the painting by ___ fetched a high price at auction.

Neither of the two LSTMs tested by Wilcox et al. (2018) captured this asymmetry. Although the two LSTMs were quite similar in terms of architecture and training corpus, they erred in opposite ways: One found both examples 6a and 6b grammatical, and the other found both unacceptable.

Overall, Wilcox and colleagues' (2018) conclusions about LSTM language models' sensitivity to filler–gap dependencies are quite upbeat. Other authors have reached more mixed conclusions. Chowdhury & Zamparelli (2018) argued that what appears to be an effect of syntactic islands on language model probabilities can be explained using other, nongrammatical factors. Chaves (2020) showed that DNNs do not capture the full complexity of island constraints—for example, they do not capture negative islands (e.g., **How fast didn't John drive___?*), where semantics and pragmatics play a central role. Warstadt et al. (2019) reported that DNNs that displayed significant sensitivity to a range of syntactic phenomena showed limited sensitivity to the island constraints they tested. In future research, it would be fruitful to establish how these mixed results arise from differences in the particular constructions and evaluation measures used in these studies.

Futrell et al. (2019) have provided converging evidence that LSTMs can keep track of syntactic state. In a paradigm similar to that of Wilcox et al. (2018), the LSTM language model they tested showed high surprise when a subordinate clause was not followed by a matrix clause, as in example 7:

(7) *As the doctor studied the textbook.

In so-called NP/Z garden path sentences such as example 8, in which a noun phrase (*the vet*...) that is preferentially attached as the direct object of the subordinate clause verb (*scratched*) later turns out to be the subject of the matrix clause, the DNN was surprised at *took* (the same reaction has been reported in human subjects; see also van Schijndel & Linzen 2018):

(8) When the dog scratched the vet with his new assistant took off the muzzle.

Futrell and colleagues (2019) compared the LSTM of Gulordava and colleagues (2018) with two models trained on orders-of-magnitude less data: a standard LSTM and a recurrent neural network grammar (RNNG) (Dyer et al. 2016), which processes words according to the correct parse of the sentences, thus incorporating a strong bias toward syntactic structure. In this minimal-data regime, only the syntactically biased RNNG was able to track syntactic state appropriately; however, the RNNG did not perform better than the Gulordava LSTM (which was trained on considerably more data).

Other studies have further investigated the ways in which the syntactic performance of a DNN is affected by its architecture and the amount of syntactic supervision it receives. McCoy et al. (2020) explored this question using the test case of auxiliary fronting in English question formation—a hierarchic phenomenon that has become the parade case of the poverty-of-the-stimulus argument (Chomsky 1986). Using the seq2seq framework (see Section 2), McCoy and colleagues trained a range of DNNs to produce questions from simple declarative sentences in a small fragment of English, as in the following pair:

(9a) The zebra **does** chuckle.
(9b) **Does** the zebra chuckle?

After training, the DNNs were asked to generate questions from statements in which the subject was modified by a relative clause, as in example 10a. This syntactic configuration, which had been withheld from the DNNs' training data (following Chomsky's assumption that such configurations are very rare in child-directed speech), differentiates two generalizations that a learner could plausibly acquire. If the DNN has learned, correctly, to front the main clause auxiliary (the MOVE-MAIN rule), it will produce question 10b. But the examples seen during training are also compatible with the nonhierarchical rule MOVE-FIRST, whereby the first auxiliary in the sentence is fronted irrespective of its syntactic role, as in example 10c:

(10a) Your zebras that *don't* dance **do** chuckle
(10b) **Do** your zebras that *don't* dance chuckle?
(10c) *Don't* your zebras that dance **do** chuckle?

Because the DNNs were trained only on examples that are ambiguous between the two rules, any preference for MOVE-MAIN would arise from the DNN's a priori bias and, possibly, from a specific bias favoring hierarchical rules. Indeed, at least some of the DNN architectures that McCoy and colleagues (2020) tested were biased in favor of MOVE-MAIN. However, this bias was not very robust: Small differences in network parameters, or even in random weight initializations of the exact same architecture, had a large effect on the outcome. Some differences are amenable to explanation; for example, gating mechanisms that disfavor counting lead to a preference for MOVE-MAIN, possibly because it is difficult to implement MOVE-FIRST without some counting device. Other factors are more difficult to interpret, especially because they interact in surprising ways; for example, different kinds of attention lead to more or less pronounced hierarchical behavior depending on the underlying gating mechanism. Finally, even the architectures that did acquire MOVE-MAIN for auxiliary fronting preferred a linear rule in a similarly ambiguous subject–verb agreement task—a result that suggests that these DNNs' bias is not reliably hierarchical.

McCoy and colleagues (2020) contrasted the mixed results for standard sequential RNNs with the robust results of experiments with tree-based RNNs, which, like the RNNG architecture we have discussed briefly above, combine the words of the sentence in an order determined by an explicit syntactic parse rather than from left to right as in standard RNNs (Pollack 1990, Socher et al. 2011). Such tree-based RNNs showed the clearest across-the-board preference for MOVE-MAIN (and for the hierarchical generalization in the agreement test). However, this robust preference emerged only when the bias was implemented in its strongest form: Both the encoder and the decoder were based on trees, and correct trees were provided for both the input and the output. An important direction for future work is to determine whether reliably hierarchical generalization can arise from weaker architectural features.

Most studies of agreement and related phenomena have focused on languages with syntactic properties similar to those of English. An interesting exception has been provided by Ravfogel et al. (2018), who studied case assignment in Basque. Basque has a number of characteristics that make it very different from English, such as relatively free word order, an ergative case system, and explicit marking of all arguments through morphemes suffixed to the verb form. Ravfogel and colleagues explored the task of reconstructing the case suffix (or lack thereof) of each word in sentences with all suffixes stripped off—for instance, reconstructing example 11a from 11b:

(11a)	Kutxazain-ek	bezeroa-ri	liburu-ak	eman	dizkiote.
	cashier-PL.ERG	customer-SG.DAT	book-PL.ABS	gave	they-them-to-her/him

‘The cashiers gave the books to the customer.’

(11b) Kutxazaina bezeroa liburua eman dizkiote.

Ravfogel et al. (2018) formulated this task as a supervised classification problem, which they tackled with a bidirectional LSTM—that is, an LSTM trained, in parallel, in both the left-to-right and right-to-left directions, to handle the fact that arguments can both precede and follow the verb. They achieved relatively high overall accuracy; the difficulties they observed were concentrated in particular around dative suffix prediction. A post hoc analysis revealed that the model was using a mix of shallow heuristics (e.g., relying on the closest verb to each argument to determine its case) and genuine generalizations (e.g., correctly encoding the distinction between the ergative and absolutive cases).

5. WHAT DO THE NETWORK’S INTERNAL REPRESENTATIONS ENCODE?

The work reviewed so far evaluates a neural network’s syntactic abilities by examining its output in response to inputs with particular syntactic properties. Much as linguists study human syntactic knowledge by providing or eliciting acceptability judgments without access to internal brain states, this so-called black box approach does not require access to the network’s inner workings. However, compared with the neuroscientific techniques that are currently available for studying human subjects, direct access to the state of an artificial network is much cheaper, and it can be more granular: We know precisely what the activation of each unit is after the DNN processes each word. To the extent that the network’s behavior indicates that it captures a particular syntactic phenomenon, there is clear interest in understanding how this behavior arises from the network’s internal states.

This sort of analysis faces major challenges. In traditional symbolic systems, internal representations are in a format that is readily interpretable by a researcher (e.g., a parse tree, a logical formula), and so are the processes that operate over those representations (e.g., syntactic transformations, deduction rules). By contrast, the internal states of a DNN consist of vectors of hundreds or thousands of real numbers, and processing in the network involves applying to those vectors arithmetic operations that are parameterized by millions of weights. To understand how the DNN’s behavior arises from its internal states, we need methods that allow us to translate these vectors into a format interpretable by a human. Such translation is difficult and may not be possible in all cases. In this section, we review a handful of methods that attempt to interpret DNN internal states and link them to the network’s behavior; for additional pointers, we refer readers to Belinkov & Glass (2019).

One of the most popular interpretation methods is based on diagnostic classifiers (also known as probing tasks; see Adi et al. 2017, Shi et al. 2016). This approach takes vectors produced by an existing network *N*—trained, for example, to represent sentences—and measures to what extent a new, separate classifier *C* can recover a particular linguistic distinction from *N*’s vector representations. This measurement is done by presenting *C* with *N*’s representations of sentences of different types—for example, sentences with singular subjects and sentences with plural subjects, labeled as such—and training it to distinguish the two classes. If *C* generalizes this distinction with high accuracy to the vector representations of new sentences, then a component of *N* that received this vector representation as input could, in principle, have access to this information (this process is analogous to multivoxel pattern analysis in neuroscience; see Haxby et al. 2001).

In a study of the mechanisms tracking subject–verb agreement in a DNN language model, for example, Giulianelli et al. (2018) showed that the plurality of the subject could be decoded with high accuracy from the hidden state of the DNN. In another study, Conneau et al. (2018) showed that a classifier could be trained to decode from a DNN's vector encoding of a sentence such syntactic information as the maximal depth of the parse tree of the sentence.

A related but distinct method is the structural probe of Hewitt & Manning (2019). This approach seeks to find a simple similarity metric between a DNN's internal representations of words in context, such that the similarity between the vector representations of two words corresponds to the distance between the two words in a syntactic parse of the sentence. Hewitt & Manning showed that such a similarity metric can be found for the representations generated by modern DNNs based on LSTMs or transformers but not for simpler (baseline) representations.

An important caveat of the methods described so far is that successful recovery of information from a network's representation does not establish that the network in fact uses that information: The information may not affect the network's behavior in any way. For example, in their study of auxiliary fronting, McCoy et al. (2018) analyzed the sentence representations produced by various networks, by training two diagnostic classifiers: one that decoded the main verb's auxiliary, which was relevant to correct generalization on the auxiliary fronting task, and one that decoded the irrelevant first auxiliary (see Section 4). Disconcertingly for the naive interpretation of diagnostic classifier accuracy, both types of information were decodable with high accuracy from the networks regardless of whether the DNNs' behavior indicated that they relied on one or the other. Giulianelli et al. (2018) addressed this concern in their diagnostic classifier study of subject–verb agreement in two ways. First, they showed that the classifier's accuracy was much lower in sentences in which the language model made incorrect agreement predictions, and second, they were able to use the classifier to intervene in the state of the network and cause it to change its agreement predictions. Very few of the studies that used the diagnostic classifier method have provided such compelling evidence linking the information decoded by the classifier to the analyzed network's behavior.

The studies discussed so far have examined the numerical activation of a large set of the units of a DNN and have treated these units jointly as a vector. Lakretz et al. (2019) explored whether the activation of an individual unit had a causal effect on the behavior of the DNN they studied. They ablated each unit—that is, set its activation to zero—and measured how this setting affected the long-distance agreement performance of Gulordava and colleagues' (2018) LSTM language model (Section 3). Using this method, Lakretz and colleagues uncovered a sparse mechanism whereby two units keep track of singular and plural number for agreement purposes. These units are in turn linked to a distributed circuit (i.e., a circuit consisting of a number of units) that records the syntactic structure of a sentence and signals when number information needs to be stored and released. This sophisticated, grammar-aware subnetwork is complemented by a bank of syntax-insensitive cells that apply agreement heuristics based on linear distance rather than syntactic structure.

In sum, Lakretz and colleagues (2019) showed that the neurons of an LSTM trained to predict the next word implement a genuine syntax-based rule to track agreement. However, the sparse mechanism the DNN develops, complemented by the heuristic linear-distance system, cannot handle multiply embedded levels of long-distance relations. What becomes increasingly difficult for the network is not the outermost agreement dependency, which is handled by the sparse circuit, but rather the embedded one: With the sparse circuit being occupied by outermost agreement tracking, agreement in the embedded clause can only rely on syntactically naive linear-distance units, which are fooled by intervening attractors. In other words, in a sentence such as example 12, the network has greater difficulty predicting the correct number for *like* than for *is*:

(12) The **kid**$_1$ that the **parents**$_2$ of our *neighbor* **like**$_2$ **is**$_1$ tall.

Such a precise understanding of how agreement checking is (imperfectly) implemented in a DNN may help us formulate predictions about the processing of syntactic dependencies by humans (Lakretz et al. 2020).

6. DISCUSSION

In the past decade, deep learning has underpinned significant advances in NLP applications. The quality of deep learning–based machine translation systems such as Google Translate and DeepL is sufficiently high that they have become useful in everyday life. Perhaps even more strikingly, DNNs that are trained only on large amounts of natural text—without "innate" linguistic constraints and with no support from explicit linguistic annotation—have shown an ability to generate long stretches of text that is grammatically valid, that is semantically consistent, and that displays coherent discourse structure (Radford et al. 2019). Here, we discuss to what extent this success should inform classic debates in linguistics about the innate mechanisms necessary for language acquisition and about human linguistic competence more generally (for a high-level perspective on the place of DNNs in cognitive science, see Cichy & Kaiser 2019).

6.1. Nature Versus Nurture

In early debates about the neural network approach to cognitive science (connectionism), neural networks were often portrayed as pure empiricist machines—that is, as learners that were devoid of innate biases and induced all their cognitive abilities from data (e.g., Christiansen & Chater 1999, Churchland 1989, Clark 1989, Fodor & Pylyshyn 1988, Pinker & Prince 1988). If that is the case, DNNs' success on syntactic tasks may be taken to indicate that human-like syntactic competence can be acquired through mere statistical learning, thus refuting the classic poverty-of-the-stimulus argument (Lasnik & Lidz 2017). However, learning theory considerations show that the notion of a tabula rasa is incoherent in practice. Finite training data are always consistent with an infinite number of possible generalizations; the stimulus is always poor. Consequently, any useful learner must have innate biases that lead it to prefer some possible generalizations over others (Mitchell 1980). DNNs are not an exception: They have biases that arise from their initial weights and from the structure of their architectures, which incorporate assumptions of temporal invariance, gating mechanisms, attention, encoding and decoding modules, and other architectural elements (see Section 2).

While DNNs are clearly not tabulae rasae, their biases are quite different from those traditionally proposed by linguists as underlying language acquisition. DNNs used for NLP are not constrained to perform only syntactically defined, recursive operations, as dictated by the structure-sensitivity-of-rules principle (Chomsky 1965, 1980) or Merge (Chomsky 1995, Hauser et al. 2002). If anything, the central architectural features of standard DNNs emphasize sequential left-to-right processing (RNNs) and content-addressable memory storage and retrieval (gating, attention). If a DNN performs syntactic tasks in a way that is consistent with human syntactic competence, we can conclude, for example, that an innate principle constraining the system to use Merge is not needed to acquire the relevant abilities. What we cannot do is ignore the biases contributed by a DNN's architecture and conclude that statistical learning from data alone suffices to acquire the relevant abilities.

At the moment, we do not know which (if any) DNN architectural features are fundamental for learning syntax. Future work should tease apart the crucial factors that enable particular DNN architectures to generalize in a human-like way from finite training data, along the lines of studies such as those of McCoy et al. (2020), which explicitly link success in diagnostic tasks to the specific priors of different DNN architectures, and those of Lakretz et al. (2019), which characterize at a mechanistic level how architectural features such as gates underlie the ability of a DNN to perform a particular linguistic task.

6.2. Incorporating Linguistic Principles into Deep Neural Networks

Instead of trying to understand how the somewhat opaque prior biases of DNNs affect their linguistic abilities, we can attempt to directly inject a particular bias of theoretical interest into their architecture and assess the impact of that bias on the generalizations acquired by the DNN. The studies of Futrell et al. (2019) and McCoy et al. (2020) reviewed in Section 4 illustrate this approach. These studies have shown that when DNNs are explicitly constrained to process words in an order dictated by a parse tree rather than from left to right, the DNNs' syntactic behavior more closely matches that of humans: They require fewer data than do standard DNNs to acquire certain generalizations, and they generalize in a human-like way to new syntactic structures (see also Hale et al. 2018, Kuncoro et al. 2018b). At this point in time, the technological tools for injecting linguistic constraints into standard DNN architectures are still in their infancy; we do not yet have reliable methods to implement proposals for innate constraints that are more specific than a general sensitivity to the parse of a sentence. Developing such tools would significantly benefit linguistics and cognitive science and, we argue, is an important area for future research. In the meantime, negative results must be taken with more than one grain of salt because they might reflect the technological difficulty of combining neural networks and symbolic knowledge rather than the inefficacy of the linguistic priors in question per se.

6.3. Amount and Nature of Training Data

Neural networks extract linguistic generalizations from raw, unannotated language data, or so the standard spiel goes. But to understand the implications of the successes and failures of a particular DNN experiment, we need to consider the nature of the training data used in the experiment. A first fundamental division is between supervised/focused and unsupervised/generic training. Consider, for example, the difference between the studies of Linzen et al. (2016) and Gulordava et al. (2018), both of which suggest that DNNs can learn (long-distance) agreement (Section 3). In Linzen and colleagues' supervised setup, the network was fed many sentences exemplifying subject–verb agreement and was explicitly trained on the objective of predicting verb number. This study thus asked the following question: Is the architecture of a DNN able, in principle, to learn agreement, even if explicit instruction is required? By contrast, Gulordava and colleagues took a network that had been trained to predict each word in a corpus from its context and investigated, with no further instruction, whether it displayed sensitivity to agreement dependencies. The question here changes to whether there is enough signal in a raw text corpus for the DNN architecture in question to correctly pick up the agreement rule.

Another central distinction is between synthetic and corpus-extracted test (or training) data. The agreement benchmarks of Linzen et al. (2016) and Gulordava et al. (2018) were derived from corpora; other studies have trained the DNN on corpus data but tested it on constructed examples (e.g., Futrell et al. 2019, Marvin & Linzen 2018) or have trained and tested the DNN on synthetic data (e.g., McCoy et al. 2020). All of these setups may lead to useful insights, but the interpretation of the results should change accordingly, and different confounds have to be taken into account. For example, on the one hand, unfiltered corpus data may contain spurious correlations that the network could rely upon (Kuncoro et al. 2018a). On the other hand, synthetic sentences might be so different from the corpus data the DNN has been trained on that its failure to handle them might have more to do with irrelevant differences in factors such as lexical frequency distributions than with the grammatical phenomenon under investigation.

Even when the training data consist of a raw corpus of naturally occurring language, as they do in popular NLP word prediction models such as BERT (Devlin et al. 2019) and GPT-2 (Radford et al. 2019), it is important to remember that these data differ significantly in both size and nature from those that a child is exposed to. In NLP, the old adage that more data is better data (Banko & Brill 2001) has held up remarkably. Over the course of just one year, NLP practitioners have increased the size of the corpora used to train word prediction models from 4 billion words (BERT) to 8 billion words (GPT-2) to well over 100 billion words (T5) (Raffel et al. 2019). Clearly, the steadily increasing volume of data made available to these systems is orders-of-magnitude larger than that available to children [at most 10 million a year, according to Hart & Risley (1995)]. This divergence limits the cognitive conclusions that can be drawn from testing off-the-shelf NLP systems and implies that linguists need to train their own DNNs to assess how the volume of training data affects DNNs' syntactic knowledge (van Schijndel et al. 2019).

DNNs' training data differ from those of children not only in size but also in nature. The books and articles that DNNs are trained on in NLP differ in their syntactic properties from child-directed speech. More generally, word prediction networks in NLP are, in essence, asked to learn language by reading entire libraries of books while sitting alone in a dark room. This setting is profoundly different from the context in which children learn language, which is grounded in perception and social interaction. Efforts are underway to introduce such grounding into DNNs (see, e.g., Chrupała et al. 2015 on perception and Weston 2016 on interaction), but these efforts are severely limited by the paucity of appropriate training data. When making claims about the syntactic abilities that can be acquired by the DNNs discussed in this survey, we must keep in mind that the real question we are asking is how much can be learned from huge amounts of written linguistic data alone. While purely distributional cues are one of the sources of information children use when acquiring syntax (Gómez & Gerken 2000), they are certainly not the only type of evidence children rely upon; this fact greatly complicates quantitative comparisons between the volume of data available to children and that available to DNNs.

6.4. Implications for the Study of Human Linguistic Abilities

The body of work we have reviewed in this survey establishes that DNNs are capable of high accuracy in challenging syntactic tasks such as implicitly distinguishing the head noun of the subject from other nouns in a sentence. At the same time, nearly all studies have reported that DNNs' behavior deviated from the idealized syntactic competence that a linguist might postulate; in the case of agreement, for instance, the DNNs' behavior suggests that they rely on a complex array of heuristics rather than on a fully fledged, context-free grammar that would allow them to correctly process center-embedded clauses, such as in example 12 above. In what ways, then, can findings on the syntactic abilities of DNNs inform the cognitive science of language in humans?

At a minimum, DNNs can be useful to thoroughly vet the stimuli of human language processing experiments. If a DNN that is known to fall short of human competence succeeds on a task, such a result suggests that the task in question may not probe the full-fledged grammatical abilities we think humans possess. Consider, for example, the experiment of Gulordava et al. (2018), who showed that DNNs were almost as good as Italian speakers in the long-distance agreement task. This finding suggests that the stimuli that were used did not truly probe the human ability to resort to a full-fledged, context-free grammar to parse sentences, or else the difference between

DNN and human subjects would have been much more dramatic given the limitations uncovered by Lakretz et al. (2019) and Marvin & Linzen (2018).

At the same time, humans themselves often deviate from linguists' idealized description of syntactic competence. We regularly make agreement errors (Bock & Miller 1991) and have difficulty parsing multiply center-embedded sentences (Chomsky & Miller 1963, Gibson & Thomas 1999). DNN error patterns, and the heuristics that give rise to those errors, may therefore serve as a source of hypotheses for experiments designed to study human syntactic performance (Linzen & Leonard 2018).

Furthermore, not all modern linguistic theorists recognize a sharp distinction between competence and performance. Under some views of syntax, grammar is more akin to a toolbox of tricks we picked up along our evolutionary way than to the maximally elegant and powerful formal grammars of computer science (e.g., Culicover & Jackendoff 2005, A. Goldberg 2019, Pinker & Jackendoff 2005). Under such views, the difference between the syntactic knowledge of DNNs and that of humans might be more one of quantity than one of quality: Humans possess a larger and more sophisticated set of heuristics to parse sentences than DNNs do, but they do not rely on any radically different, more powerful narrow language faculty abilities. If that is the case, the behavior of DNNs might give us insights not only into online processing (performance) but also into some of the core syntactic tools that constitute human grammatical competence. We look forward to theoretical work that links modern DNNs to construction grammar and similarly "shallow" syntactic formalisms.

7. CONCLUSION

In our view, the time is ripe for more linguists to get engaged in the lines of work we have sketched in this review. On the one hand, linguists' know-how in probing grammatical knowledge can help develop the next generation of language processing DNNs, and the success of events such as the BlackboxNLP series confirms that the deep learning community is warmly welcoming linguistic analyses of DNNs. On the other hand, studying what the best DNNs learn about grammar, and how they do so, can offer new insights into the nature of language and, ultimately, into what is genuinely unique about the human species. For this line of work to be effective, linguists will need to be closely involved in developing relevant network architectures, training them on appropriate data, and conducting experiments that address linguists' theoretical concerns.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

T.L. was supported by a Google Faculty Research Award, National Science Foundation grant BCS-1920924, and the United States–Israel Binational Science Foundation (award 2018284). We thank Robert Frank, Ethan Wilcox, and the JHU Computation and Psycholinguistics Lab for discussion of the issues reviewed in the article.

LITERATURE CITED

Adi Y, Kermany E, Belinkov Y, Lavi O, Goldberg Y. 2017. *Fine-grained analysis of sentence embeddings using auxiliary prediction tasks*. Paper presented at the 5th International Conference on Learning Representations (ICLR), Toulon, Fr., Apr. 24–26. **https://openreview.net/pdf?id=BJh6Ztuxl**

Bahdanau D, Cho K, Bengio Y. 2015. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 [cs.CL]

Banko M, Brill E. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 26–33. Stroudsburg, PA: Assoc. Comput. Linguist.

Belinkov Y, Glass J. 2019. Analysis methods in neural language processing: a survey. *Trans. Assoc. Comput. Linguist.* 7:49–72

Bernardy J, Lappin S. 2017. Using deep neural networks to learn syntactic agreement. *Linguist. Issues Lang. Technol.* 15:1–15

Bock K, Miller C. 1991. Broken agreement. *Cogn. Psychol.* 23:45–93

Chaves RP. 2020. What don't RNN language models learn about filler-gap dependencies? *Proc. Soc. Comput. Linguist.* 3:20–30

Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, et al. 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 (EMNLP)*, pp. 1724–34. Stroudsburg, PA: Assoc. Comput. Linguist.

Chomsky N. 1957. *Syntactic Structures*. The Hague, Neth.: Mouton

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

Chomsky N. 1980. Rules and representations. *Behav. Brain Sci.* 3:1–15

Chomsky N. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Westport, CT: Praeger

Chomsky N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press

Chomsky N, Miller GE. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, Vol. 2, ed. R Luce, R Bush, E Galanter, pp. 269–321. New York: Wiley

Chowdhury S, Zamparelli R. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 133–44. Stroudsburg, PA: Assoc. Comput. Linguist.

Christiansen M, Chater N. 1999. Connectionist natural language processing: the state of the art. *Cogn. Sci.* 23:417–37

Chrupała G, Kádár A, Alishahi A. 2015. Learning language through pictures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Vol. 2, pp. 112–18. Stroudsburg, PA: Assoc. Comput. Linguist.

Churchland P. 1989. *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science*. Cambridge, MA: MIT Press

Cichy RM, Kaiser D. 2019. Deep neural networks as scientific models. *Trends Cogn. Sci.* 23:305–17

Clark A. 1989. *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing*. Cambridge, MA: MIT Press

Conneau A, Kruszewski G, Lample G, Barrault L, Baroni M. 2018. What you can cram into a single $&!#* vector: probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 2126–36. Stroudsburg, PA: Assoc. Comput. Linguist.

Cui Y, Chen Z, Wei S, Wang S, Liu T, Hu G. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 593–602. Stroudsburg, PA: Assoc. Comput. Linguist.

Culicover P, Jackendoff R. 2005. *Simpler Syntax*. Oxford, UK: Oxford Univ. Press

Devlin J, Chang MW, Lee K, Toutanova K. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–86. Stroudsburg, PA: Assoc. Comput. Linguist.

Dyer C, Kuncoro A, Ballesteros M, Smith N. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 199–209. Stroudsburg, PA: Assoc. Comput. Linguist.

Edunov S, Ott M, Auli M, Grangier D. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500. Stroudsburg, PA: Assoc. Comput. Linguist.

Elman JL, Bates EA, Johnson MH, Karmiloff-Smith A, Parisi D, Plunkett K. 1998. *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press

Everaert M, Huybregts M, Chomsky N, Berwick R, Bolhuis J. 2015. Structures, not strings: linguistics as part of the cognitive sciences. *Trends Cogn. Sci.* 19:729–43

Fodor J, Pylyshyn Z. 1988. Connectionism and cognitive architecture: a critical analysis. *Cognition* 28:3–71

Futrell R, Wilcox E, Morita T, Qian P, Ballesteros M, Levy R. 2019. Neural language models as psycholinguistic subjects: representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 32–42. Stroudsburg, PA: Assoc. Comput. Linguist.

Gibson E, Thomas J. 1999. Memory limitations and structural forgetting: the perception of complex ungrammatical sentences as grammatical. *Lang. Cogn. Process.* 14:225–48

Giulianelli M, Harding J, Mohnert F, Hupkes D, Zuidema W. 2018. Under the hood: using diagnostic classifiers to investigate and improve how language models track agreement information. See Linzen et al. 2018, pp. 240–48

Goldberg A. 2019. *Explain Me This: Creativity, Competition, and the Partial Productivity of Constructions*. Princeton, NJ: Princeton Univ. Press

Goldberg Y. 2017. *Neural Network Methods for Natural Language Processing*. San Francisco: Morgan & Claypool

Goldberg Y. 2019. Assessing BERT's syntactic abilities. arXiv:1901.05287 [cs.CL]

Gómez R, Gerken L. 2000. Infant artificial language learning and language acquisition. *Trends Cogn. Sci.* 4:178–86

Gulordava K, Bojanowski P, Grave E, Linzen T, Baroni M. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 1195–1205. Stroudsburg, PA: Assoc. Comput. Linguist.

Hale J, Dyer C, Kuncoro A, Brennan J. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2727–36. Stroudsburg, PA: Assoc. Comput. Linguist.

Hart B, Risley TR. 1995. *Meaningful Differences in the Everyday Experience of Young American Children*. Baltimore, MD: Brookes

Hauser M, Chomsky N, Fitch T. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* 298:1569–79

Haxby J, Gobbini I, Furey M, Ishai A, Schouten J, Pietrini P. 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science* 293:2425–30

Hewitt J, Manning C. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 4129–38. Stroudsburg, PA: Assoc. Comput. Linguist.

Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput*. 9:1735–80

Jurafsky D, Martin J. 2008. *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall. 2nd ed.

Kalchbrenner N, Grefenstette E, Blunsom P. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 655–65. Stroudsburg, PA: Assoc. Comput. Linguist.

Kuncoro A, Dyer C, Hale J, Blunsom P. 2018a. The perils of natural behaviour tests for unnatural models: the case of number agreement. Poster presented at Learning Language in Humans and in Machines, Paris, Fr., July 5–6. **https://osf.io/9usyt/**

Kuncoro A, Dyer C, Hale J, Yogatama D, Clark S, Blunsom P. 2018b. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 1426–36. Stroudsburg, PA: Assoc. Comput. Linguist.

Lakretz Y, Dehaene S, King JR. 2020. What limits our capacity to process nested long-range dependencies in sentence comprehension? *Entropy* 22:446

Lakretz Y, Kruszewski G, Desbordes T, Hupkes D, Dehaene S, Baroni M. 2019. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 11–20. Stroudsburg, PA: Assoc. Comput. Linguist.

Lasnik H, Lidz J. 2017. The argument from the poverty of the stimulus. In *Oxford Handbook of Universal Grammar*, ed. I Roberts, pp. 221–48. Oxford, UK: Oxford Univ. Press

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* 521:436–44

Leshno M, Lin V, Pinkus A, Schocken S. 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw*. 6:861–67

Linzen T, Chrupała G, Alishahi A, eds. 2018. *The 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP: Proceedings of the First Workshop*. Stroudsburg, PA: Assoc. Comput. Linguist.

Linzen T, Chrupała G, Belinkov Y, Hupkes D, eds. 2019. *The BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP at ACL 2019: Proceedings of the Second Workshop*. Stroudsburg, PA: Assoc. Comput. Linguist.

Linzen T, Dupoux E, Goldberg Y. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguist.* 4:521–35

Linzen T, Leonard B. 2018. Distinct patterns of syntactic agreement errors in recurrent networks and humans. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pp. 692–97. Austin, TX: Cogn. Sci. Soc.

Marvin R, Linzen T. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202. Stroudsburg, PA: Assoc. Comput. Linguist.

McCoy T, Frank R, Linzen T. 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*, pp. 2093–98. Austin, TX: Cogn. Sci. Soc.

McCoy T, Frank R, Linzen T. 2020. Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Trans. Assoc. Comput. Linguist.* 8:125–40

Mitchell TM. 1980. *The need for biases in learning generalizations*. Tech. Rep., Rutgers Univ., New Brunswick, NJ

Pinker S, Jackendoff R. 2005. The faculty of language: What's special about it? *Cognition* 95:201–36

Pinker S, Prince A. 1988. On language and connectionism: analysis of a parallel distributed processing model of language acquisition. *Cognition* 28:73–193

Pollack JB. 1990. Recursive distributed representations. *Artif. Intel.* 46:77–105

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. 2019. *Language models are unsupervised multitask learners*. Work. Pap., OpenAI, San Francisco. **https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf**

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, et al. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683 [cs.LG]

Ravfogel S, Goldberg Y, Tyers F. 2018. Can LSTM learn to capture agreement? The case of Basque. See Linzen et al. 2018, pp. 98–107

Rogers A, Kovaleva O, Rumshisky A. 2020. A primer in BERTology: what we know about how BERT works. arXiv:2002.12327 [cs.CL]

Ross J. 1967. *Constraints on variables in syntax*. PhD Diss., Mass. Inst. Technol., Cambridge, MA

Shi X, Padhi I, Knight K. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1526–34. Stroudsburg, PA: Assoc. Comput. Linguist.

Socher R, Lin CC, Ng AY, Manning CD. 2011. Parsing natural scenes and natural language with recursive neural networks. In *ICML'11: Proceedings of the 28th International Conference on Machine Learning*, pp. 129–36. Madison, WI: Omnipress

Sutskever I, Vinyals O, Le QV. 2014. Sequence to sequence learning with neural networks. In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2, ed. Z Ghahramani, M Welling, C Cortes, ND Lawrence, KQ Weinberger, pp. 3104–12. Cambridge, MA: MIT Press

Tran K, Bisazza A, Monz C. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4731–36. Stroudsburg, PA: Assoc. Comput. Linguist.

van Schijndel M, Linzen T. 2018. Modeling garden path effects without explicit hierarchical syntax. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, ed. T Rogers, M Rau, J Zhu, C Kalish, pp. 2603–8. Austin, TX: Cogn. Sci. Soc.

van Schijndel M, Mueller A, Linzen T. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5831–37. Stroudsburg, PA: Assoc. Comput. Linguist.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, et al. 2017. Attention is all you need. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, ed. U von Luxburg, pp. 6000–10. Red Hook, NY: Curran

Warstadt A, Parrish A, Liu H, Mohananey A, Peng W, et al. 2019. BLiMP: the benchmark of linguistic minimal pairs for English. arXiv:1912.00582 [cs.CL]

Weston J. 2016. Dialog-based language learning. In *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, ed. DD Lee, pp. 829–37. Red Hook, NY: Curran

Wilcox E, Levy R, Morita T, Futrell R. 2018. What do RNN language models learn about filler–gap dependencies? See Linzen et al. 2018, pp. 211–21