Emergent linguistic structure in artificial neural networks trained by self-supervision

Christopher D. Manning[a,1], Kevin Clark[a], John Hewitt[a], Urvashi Khandelwal[a], and Omer Levy[b]

[a]Computer Science Department, Stanford University, Stanford, CA 94305; and [b]Facebook Artificial Intelligence Research, Facebook Inc., Seattle, WA 98109

Edited by Matan Gavish, Hebrew University of Jerusalem, Jerusalem, Israel, and accepted by Editorial Board Member David L. Donoho April 13, 2020 (received for review June 3, 2019)

This paper explores the knowledge of linguistic structure learned by large artificial neural networks, trained via self-supervision, whereby the model simply tries to predict a masked word in a given context. Human language communication is via sequences of words, but language understanding requires constructing rich hierarchical structures that are never observed explicitly. The mechanisms for this have been a prime mystery of human language acquisition, while engineering work has mainly proceeded by supervised learning on treebanks of sentences hand labeled for this latent structure. However, we demonstrate that modern deep contextual language models learn major aspects of this structure, without any explicit supervision. We develop methods for identifying linguistic hierarchical structure emergent in artificial neural networks and demonstrate that components in these models focus on syntactic grammatical relationships and anaphoric coreference. Indeed, we show that a linear transformation of learned embeddings in these models captures parse tree distances to a surprising degree, allowing approximate reconstruction of the sentence tree structures normally assumed by linguists. These results help explain why these models have brought such large improvements across many language-understanding tasks.

artificial neural netwok | self-supervision | syntax | learning

Human language communication is via sequences of words, canonically produced as a mainly continuous speech stream (1). Behind this linear organization is a rich hierarchical language structure with additional links (such as coreference between mentions) that needs to be understood by a hearer (or reader). In Fig. 1, for instance, a hearer has to understand a sentence structure roughly like the one shown to realize that the chef was out of food rather than the store.* Language understanding, like vision, can be seen as an inverse problem (3), where the hearer has to reconstruct structures and causes from the observed surface form.

In computational linguistics, the long dominant way of addressing this structure induction problem has been to hand design linguistic representations, broadly following proposals from linguistics proper. Under one set of conventions, the sentence in Fig. 1 would be annotated with the structure shown. Humans then label many natural language sentences with their underlying structure. Such datasets of annotated human language structure, known as treebanks (4, 5), have fueled much of the research in the field in the last 25 y. Researchers train progressively better supervised machine-learning models on the treebank, which attempt to recover this structure for any sentence (6–8). This approach has been very effective as an engineering solution, but beyond the high practical cost of human labeling, it gives no insight into how children might approach structure induction from observed data alone.

Recently, enormous progress has been made in natural language representation learning by adopting a self-supervised learning approach. In self-supervised learning, a system is given no explicit labeling of raw data, but it is able to construct its own supervised learning problems by choosing to interpret some of the data as a "label" to be predicted.† The canonical case for human language is the language-modeling task of trying to predict the next word in an utterance based on the temporally preceding words (Fig. 2). Variant tasks include the masked language-modeling task of predicting a masked word in a text [a.k.a. the cloze task (11)] and predicting the words likely to occur around a given word (12, 13). Autoencoders (14) can also be thought of as self-supervised learning systems. Since no explicit labeling of the data is required, self-supervised learning is a type of unsupervised learning, but the approach of self-generating supervised learning objectives differentiates it from other unsupervised learning techniques such as clustering.

One might expect that a machine-learning model trained to predict the next word in a text will just be a giant associational learning machine, with lots of statistics on how often the word restaurant is followed by kitchen and perhaps some basic abstracted sequence knowledge such as knowing that adjectives are commonly followed by nouns in English. It is not at all clear that such a system can develop interesting knowledge of the linguistic structure of whatever human language the system is trained on. Indeed, this has been the dominant perspective in linguistics, where language models have long been seen as inadequate and having no scientific interest, even when their usefulness in practical engineering applications is grudgingly accepted (15, 16).

Starting in 2018, researchers in natural language processing (NLP) built a new generation of much larger artificial

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, "The Science of Deep Learning," held March 13–14, 2019, at the National Academy of Sciences in Washington, DC. NAS colloquia began in 1991 and have been published in PNAS since 1995. From February 2001 through May 2019 colloquia were supported by a generous gift from The Dame Jillian and Dr. Arthur M. Sackler Foundation for the Arts, Sciences, & Humanities, in memory of Dame Sackler's husband, Arthur M. Sackler. The complete program and video recordings of most presentations are available on the NAS website at http://www.nasonline.org/science-of-deep-learning.

Author contributions: C.D.M., K.C., J.H., U.K., and O.L. designed research; K.C., J.H., and U.K. performed research; and C.D.M., K.C., J.H., U.K., and O.L. wrote the paper.

Competing interest statement: K.C. and U.K. have been/are employed part time at Google Inc., and K.C. has a Google PhD Fellowship. Researchers at Google Inc. developed the BERT model analyzed in this paper.

This article is a PNAS Direct Submission. M.G. is a guest editor invited by the Editorial Board.

Data deposition: Code and most of the data to reproduce the analyses in this paper are freely available at https://github.com/clarkkev/attention-analysis and https://github.com/john-hewitt/structural-probes.

[1]To whom correspondence may be addressed. Email: manning@cs.stanford.edu.

First published June 3, 2020.

*There are two main approaches to depicting a sentence's syntactic structure: phrase structure (or constituency) and dependency structure (or grammatical relations). The former is dominant in modern linguistics, but in this paper we use the latter, which is dominant in computational linguistics. Both representations capture similar, although generally not identical, information (2).

†The approach of self-supervised learning has existed for decades, used particularly in robotics, e.g., refs. 9 and 10, but it has recently been revived as a focus of interest, used also for vision and language.

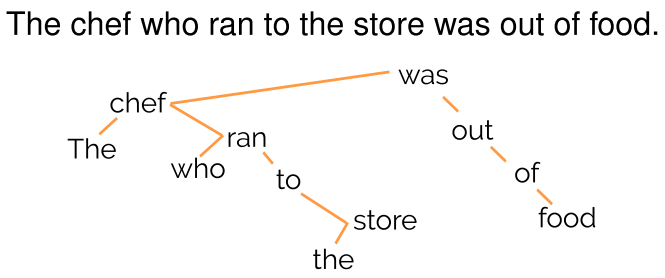


Fig. 1. A hearer must reconstruct that the store is in a relative clause modifying the chef to know that it is the chef who is out of food rather than the linearly closer store.

neural network models, which construct rich, word-token specific deep contextual representations of human language as numeric vectors (17, 18). In this paper, we examine how, at this larger scale, there is a dramatic increase in what is modeled by such networks. The simple task of word prediction is a highly effective self-supervision signal: Neural networks can and do improve on this task by inducing their own representations of sentence structure which capture many of the notions of linguistics, including word classes (parts of speech), syntactic structure (grammatical relations or dependencies), and coreference (which mentions of an entity refer to the same entity, such as, e.g., when "she" refers back to "Rachel"). We examine learned attention structure in models and develop simple probes to show that these models know about each of these types of linguistic information. Indeed, the learned encoding of a sentence to a large extent includes the information found in the parse tree structures of sentences that have been proposed by linguists.

This is a startling and intriguing result. Traditionally much of the emphasis in NLP has been on using labels for part of speech, syntax, etc., as an aid in other downstream tasks. This result suggests that large-scale hand construction of syntactically labeled training data may no longer be necessary for many tasks. Despite its simple nature, the generality of word prediction, as a task that benefits from syntactic, semantic, and discourse information, leads to it being a very powerful multidimensional supervision signal.

While the work presented here is interesting food for thought about the starting point and process of human language acquisition, we make no attempt to model human learning. These models are operating in a quite different environment from that of children, with exposure to much more linguistic input but no real-world environment to aid learning. Nevertheless, this work bears on the logical problem of language acquisition (19). Importantly, it shows successful language structure learning from positive evidence alone.

Bidirectional Encoder Representations from Transformers: A Self-Supervised Artificial Neural Network

Current state-of-the-art NLP systems typically involve a deep artificial neural network that was trained on a large corpus of text using self-supervision. As an example, we describe Bidirectional Encoder Representations from Transformers (BERT), a recently proposed Transformer model and training procedure that has gained prominence by dominating multiple key NLP benchmarks (18, 20).

(a) A typical restaurant kitchen has several different _____
(b) A typical restaurant _____ has several different stations

Fig. 2. The next word prediction (language-modeling) task (a) and the cloze task (b).

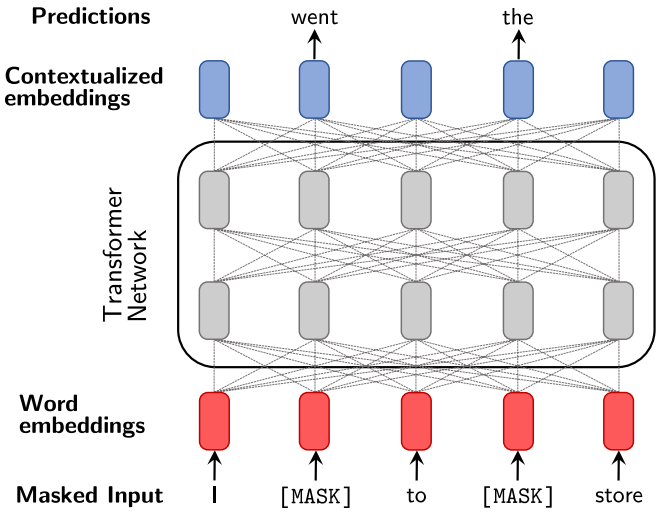


Fig. 3. A high-level illustration of BERT. Words in the input sequence are randomly masked out and then all words are embedded as vectors in $\mathbb{R}^d$. A Transformer network applies multiple layers of multiheaded attention to the representations. The final representations are used to predict the identities of the masked-out input words.

The self-supervision task used to train BERT is the masked language-modeling or cloze task, where one is given a text in which some of the original words have been replaced with a special mask symbol. The goal is to predict, for each masked position, the original word that appeared in the text (Fig. 3). To perform well on this task, the model needs to leverage the surrounding context to infer what that word could be.

BERT is a Transformer model (21), a neural network architecture, without any recurrent connections (22), which takes a sequence of words (or other symbols) as input and produces a contextualized vector representation of each word as its output (Fig. 3). It contains many millions of trainable parameters in a number of layers, typically requiring massive amounts of data and computation to train. This makes Transformers difficult to train, but also highly expressive models that can outperform other contemporary neural networks when properly optimized.

The key mechanism by which Transformers contextualize representations is multiheaded attention (see Fig. 5). Attention (23) dynamically assigns a weight to every pair of words in the sequence, indicating how much the model should "pay attention to" the first word when computing the representation of the second one. Transformers use multiple attention heads in parallel, where each head can potentially capture a completely different word–word relation. Transformers aggregate the information from each head to produce a single output vector representation for each word in the sequence. We provide more mathematical detail below.

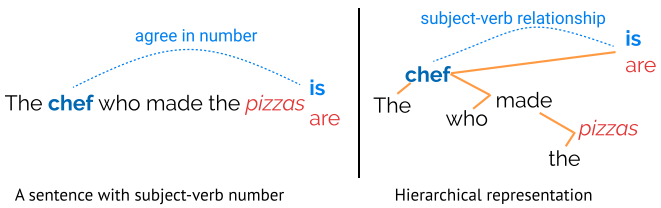


Fig. 4. An example where implicitly modeling syntactic structure may assist in predicting the missing word and improve language-modeling performance.

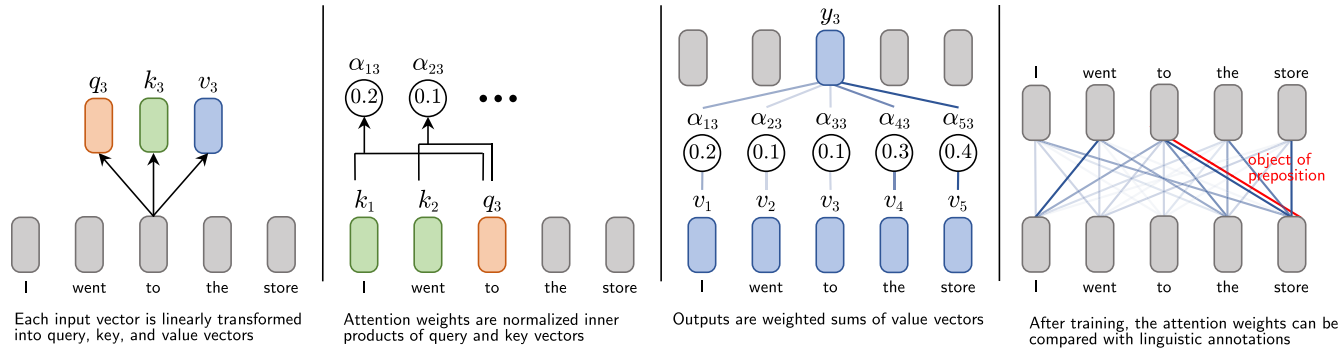


Fig. 5. An overview of the neural attention mechanism. NLP models often contain many attention mechanisms, each producing a different set of attention weights.

The Syntax Sensitivity of Language Models

Models trained to predict words, such as BERT, receive just a list of words as their input and do not explicitly represent syntax. Are they nevertheless able to learn the hierarchical structure of language? One approach to investigating this question is through examining the model's predictions in syntax-dependent scenarios.

For example, performing English subject–verb agreement requires an understanding of hierarchical structure. In the sentences "The chef is here" and "The chefs are here" the form of the verb depends on whether the subject is singular or plural: "is" agrees with "chef" but "are" does not. It is quite unsurprising that neural language models can learn statistics about sequential co-occurrence, e.g., that a singular noun is often followed by a singular verb. However, subject–verb agreement is based not on the linear ordering of the words, but on the words' syntactic relationship. For example, in "The chef who made the pizzas is here," the intervening phrase does not affect the correct agreement of "is" despite the phrase containing an "attractor" noun "pizzas," which has the opposite number to the verb's subject (Fig. 4).

Linzen et al. (24) first evaluated neural language models on their ability to perform such agreement tasks. Models were asked to predict the next word for inputs with zero or more intervening attractor nouns. Accuracy was measured as how often the model assigns higher probability to the correct verb form. To correctly perform this word form prediction task, the model must ignore the attractor(s) (here, pizzas) and assign the verb's form based on its syntactic subject (chef), in accord with the sentence's implied hierarchical structure rather than linear proximity.

Subsequent work has evaluated models on more challenging sentences where potential confounds are removed (25), evaluated models on other grammatical phenomena such as reflexive anaphora (26), evaluated models that explicitly model hierarchy (27), and evaluated BERT on agreement across attractors (28). The studies have found models like BERT to perform well at these tasks; for example, ref. 28 finds that BERT makes as many or fewer mistakes than humans for some types of agreement.[‡]

Agreement relations highlight the richness of language modeling as a pretraining task, perhaps explaining some of its recent success in NLP. However, while this black box approach teaches us that the model learns some form of syntax, it does not tell us how. We therefore shift our discussion to analyze the internals of BERT and demonstrate two separate mechanisms (attention and structural probes) by which we can observe the structure the model constructs from the input.

Attention Probes

Neural attention is a neural network component prevalent in contemporary state-of-the-art NLP models such as BERT. Here, we provide a precise description of attention and explain how it can be interpreted linguistically. We then apply this interpretation to BERT's multiple-attention heads and show that many heads are highly correlated with well-known concepts of linguistic structure.

The Neural Attention Mechanism. Given a query q and a set of n items $\{x_1, \ldots, x_n\}$, the attention mechanism (23) induces a probability distribution α over the item set and then produces an expectation (weighted sum) of the items as the output. Intuitively, attention makes a soft selection of which item x_i is the best fit for the query q. More formally, each item x_i is represented by two vectors: key k_i and value v_i. The key k_i is used to determine the distribution α via an inner product with the query vector q, normalized by a softmax:

$$\alpha_i = \frac{\exp q^\top k_i}{\sum_{\ell=1}^{n} \exp q^\top k_\ell}. \quad [1]$$

The output y is then the expectation over the value vectors v_i:

$$y = \sum_{i=1}^{n} \alpha_i v_i. \quad [2]$$

Table 1. Well-performing BERT attention heads on WSJ SD dependency parsing by dependency type

Relation	Attention precision	Baseline precision
Microaverage across dependency types		
Best single head	34.5	26.3
Best head per dependency type	69.3	50.8
Single heads for individual dependency types		
Nominal subject	58.4	45.4
Direct object	86.8	40.0
Clausal complement	48.8	12.4
Object of preposition	76.3	34.6
Predeterminer	94.3	51.7
Marker	50.7	14.5
Passive auxiliary	82.5	40.5
Phrasal verb particle	99.1	91.4

[‡]Psycholinguists have shown that attractors also cause humans to make agreement errors (29, 30).

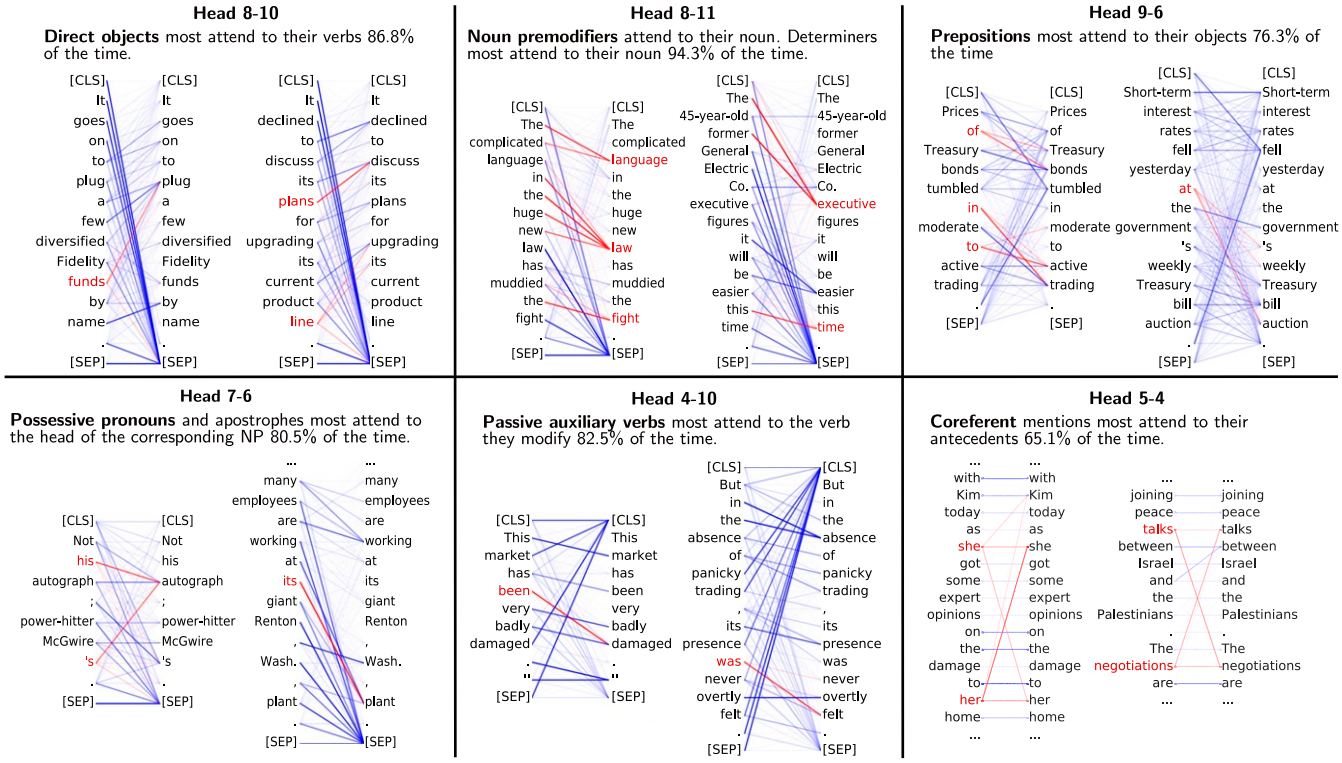


Fig. 6. Some BERT attention heads that appear sensitive to linguistic phenomena, despite not being explicitly trained on linguistic annotations. In the example attention maps, the darkness of a line indicates the size of the attention weight. All attention to/from red words is colored red; these words are chosen to highlight certain of the attention heads' behaviors. [CLS] (classification) and [SEP] (separator) are special tokens BERT adds to the input during preprocessing. Attention heads are numbered by their layer and index in BERT. Reprinted with permission from ref. 59, which is licensed under CC BY 4.0.

The Transformer network (21) uses a flavor of this mechanism called self-attention, where each input word plays a dual role as both a query and a selectable item. This is implemented by passing the vector representation of every word x_i through three different linear transformations, resulting in query q_i, key k_i, and value v_i vectors. Each query q_j can then attend over all of the key-value pairs (k_i, v_i) in the sequence, producing a different attention distribution α^j (i.e., α^j_i denotes the attention weight toward position i from position j) and output y_j for each word, as shown in Fig. 5.

Attention has been a highly successful neural network component for processing text (31), video (32), image (33), and speech (34) data. A Transformer network consists of multiple layers with each layer containing multiple attention heads. Each head computes its own independent attention weights and output vectors; the output vectors across heads are concatenated together when producing a layer's output.

Method: Attention Heads as Simple Classifiers. We quantitatively study the correspondence between attention heads and linguistic phenomena by observing the behavior of attention heads on corpora of annotated data. We focus on data where the annotations assign each pair of words (w_i, w_j) a label $l(w_i, w_j)$ that is 1 if a particular linguistic relationship between words holds (e.g., w_i is w_j's syntactic head) and is 0 if otherwise. To interpret what an attention head in BERT is computing, we examine the most-attended-to word at each position. More formally, if $\alpha(w, h)$ denotes the attention distribution of head h when BERT is run over the sequence of words $w = [w_1, \ldots, w_n]$, we find the most-attended-to word $w_{\text{argmax}_i \alpha(w,h)^j_i}$ for each position $1 \le j \le n$. We then evaluate whether the attention head is expressing a particular linguistic relationship by computing how often the most-attended-to word is in that relationship with the input word (i.e., how often the head "pays attention to" linguistically relevant words). If $S_l(w) = \{j : \sum_{i=1}^{n} l(w_i, w_j) > 0\}$ is the subset of the input expressing the annotated relationship, the precision score for the head is computed as

$$\text{precision}(h) = \frac{1}{N} \sum_{w \in \text{corpus}} \sum_{j \in S_l(w)} l(w_{\text{argmax}_i \alpha(w,h)^j_i}, w_j), \quad [3]$$

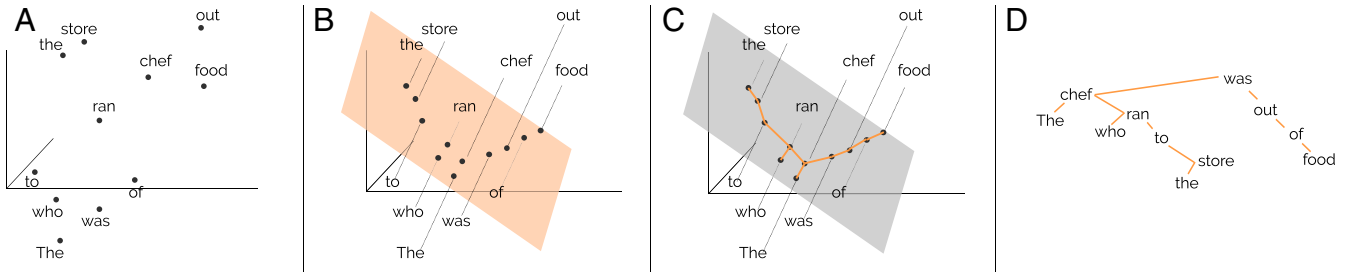
where N is the total number of words in the corpus expressing the relationship. This score can be viewed as evaluating the attention head as a simple classifier that predicts the presence of the linguistic relationship of interest.

Experiments. We use the original base-sized uncased BERT model from Google, which consists of 12 layers each containing 12 attention heads and 768-dimensional hidden vectors. We use "head ⟨layer⟩-⟨index⟩" to denote a particular attention head.

Our first evaluation is on syntactic dependencies, using the Wall Street Journal (WSJ) portion of the Penn Treebank (4, 35) annotated with Stanford Dependencies (SD) (36) as the corpus. In dependency syntax, typed directed edges connect words, forming a tree structure describing the sentence's syntax. In particular, the tree structure results from each word having exactly

Table 2. Precisions (%) of systems selecting a correct antecedent for a coreferent mention in the CoNLL-2012 data by mention type

Model	All	Pronoun	Proper	Nominal
Nearest	15	23	9	11
Rule based	66	72	73	48
Head 5-4	70	68	76	64



Each of the words of the sentence *The chef who ran to the store was out of food* is internally represented in context as a vector.

A structural probe finds a linear transform of that space under which squared L_2 distance between vectors best reconstructs tree path distance between words.

Once in this latent space, the structure of the tree is globally represented by the geometry of the vector space, meaning words that are close in the space are close in the tree.

In fact, the tree can be approximately recovered by taking a minimum spanning tree in the latent syntax space.

Fig. 7. (*A–D*) An overview of the structural probe method.

one incoming edge, from either another word (the "syntactic head") or a distinguished sentence root symbol, with a type indicating the grammatical relation (e.g., prepositional object). The tree can be expressed as word-pair labels where $l(w_i, w_j)$ is 1 if w_i is w_j's head and 0 if otherwise. We also perform more fine-grained analysis over specific grammatical relations by (for example) restricting the label to 1 if w_i is w_j's direct object and 0 if otherwise.[§] Some dependency relations are simpler to predict than others; e.g., a noun's determiner is often the immediately preceding word. Therefore, as a point of comparison, we show predictions from a simple fixed-offset baseline. A fixed offset of -2 means the word two positions to the left of the dependent is always considered to be the head; we report scores for the best-performing offset in $[-5, 5]$.

We also evaluate BERT attention heads at the semantic phenomenon of coreference. Coreference occurs when two or more mentions (text expressions referring to real-word entities) in a document refer to the same entity (e.g., "London," "the city," and "it" could be coreferent mentions). We evaluate the attention heads on coreference resolution using the Conference on Natural Language Learning shared task (CoNLL-2012) dataset (37). We report antecedent selection precision: how often the head word of a coreferent mention most attends to the head word of one of that mention's antecedents, so $l(w_i, w_j) = 1$ when w_i and w_j are head words of coreferent mentions and 0 if otherwise. We compare against two baselines for selecting an antecedent: first, picking the nearest other mention as the antecedent, and second, using a rule-based coreference system. It proceeds through four sieves: 1) full string match, 2) head word match, 3) number/gender/person match, and 4) all other mentions. The nearest mention satisfying the earliest sieve is returned. Although simple, these baselines can perform surprisingly well at coreference (38).

Results. Results for dependency syntax are shown in Table 1. No single attention head corresponds well to dependency syntax overall; the best head gets 34.5% accuracy, which is not much better than the right-branching baseline (26.3% accuracy). However, we find that certain attention heads specialize to specific dependency relations, sometimes achieving high accuracy and substantially outperforming the fixed-offset baseline. Qualitative examples of these attention heads are shown in Fig. 6. Beyond looking at individual attention heads, we also report the combined score when taking the best attention head for each dependency type and find it achieves around 70% precision, substantially higher than the baseline. Explicitly incorporating syntactic information to improve attention has been an active area of NLP research (39–41). Our results suggest that self-supervised training can cause learned syntax-aware attention to arise in NLP models.

We note that heads can disagree with annotation conventions while still performing syntactic behavior. For example, head 7-6 marks the 's as the dependent (e.g., Fig. 6, *Bottom Left*) for the noun possessive relation, while gold-standard labels mark the complement of an 's as the dependent. There is significant arbitrariness and variation in the annotation conventions of different treebanks and linguistic theories, and such behavior is not clearly wrong. The disagreement highlights how these syntactic behaviors in BERT emerge as a by-product of self-supervised training, not by copying a human design.

Results for coreference are shown in Table 2. One of BERT's attention heads achieves quite strong performance, outscoring the rule-based system. It is particularly effective on nominal mentions, perhaps because neural representations are well suited to fuzzy matching between synonyms (e.g., Fig. 6, *Bottom Right*).

Finding Syntax Trees in Word Representations

Since many of BERT's attention heads encode individual syntactic relations, it is natural to wonder whether the representation, that is, the vectors that represent the words in each layer of BERT, embed syntax trees. More generally, a fundamental question when investigating linguistic structure in neural networks is whether the internal representations of networks are reconcilable with the tree structures of sentences. At face value, this seems unlikely; trees are discrete structures, not immediately compatible with the high-dimensional real-valued $\mathbb{R}^d$ spaces of neural representations.

In this section, we explain the structural probe method (42) for finding trees embedded in the internal representations of a network and demonstrate the surprising extent to which syntax trees are linearly extractable from internal representations. At a high level, this method finds a single distance metric on $\mathbb{R}^d$ which, when applied to any two word representations constructed by the model for the same sentence, approximates the result of the distance metric defined by the syntax tree of that sentence (Fig. 7).

The Structural Probe Method. To test whether trees are embedded in an $\mathbb{R}^d$ representation, we require common ground between trees and vector spaces. We observe that the set of nodes in an undirected tree T is a metric space, where the metric $d_T(i, j)$ between nodes i and j is defined as the number of edges in the path between i and j. Given only a path metric, one can reconstruct the tree itself by indicating that all elements i, j with $d_T(i, j)$ equal to 1 are connected by an edge. So, instead of attempting to find trees in $\mathbb{R}^d$, we can look for a distance metric,

[§]Intuitively, the precision score in this case measures what percent of the time the most-attended-to word of a verb is that verb's direct object.

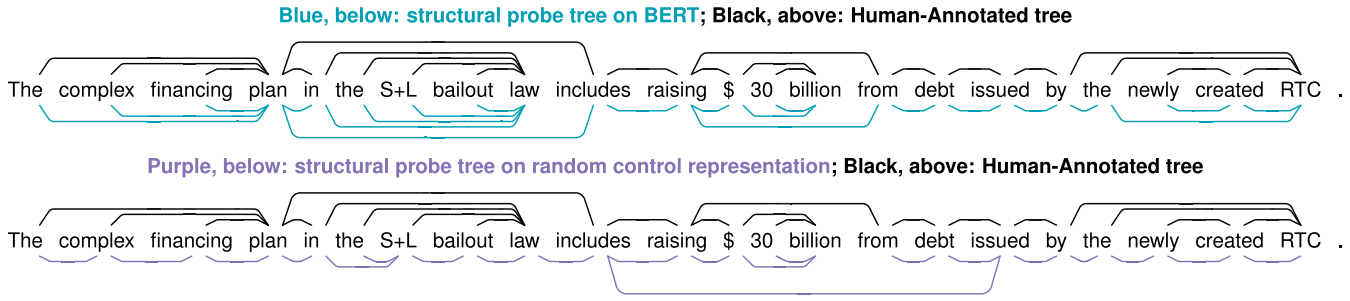


Fig. 8. Minimum-spanning trees resultant from structural probes on BERT and a random control representation Proj0 compared to the human-annotated parse tree. In the text sentence, "S+L" refers to American savings and loans banks and "RTC" refers to the Resolution Trust Corporation. Reprinted with permission from ref. 42, which is licensed under CC BY 4.0.

a more natural notion for a vector space. "Looking for a distance metric" means defining a distance metric with tunable parameters and using supervised data to find the parameters that best fit the data.

Intuitively, we want to construct a distance metric that focuses on certain aspects of the space; we expect only some aspects of the representation to encode syntax, since most will encode sentence meaning. We note that L_2 distance on $\mathbb{R}^d$ can be parameterized with a positive semidefinite[¶] matrix $A \in \mathbb{S}_+^{d\times d}$. All such matrices can in turn be represented as $A = B^\top B$ for some matrix $B \in \mathbb{R}^{k\times d}$, leading to a distance of the following form:

$$\begin{aligned} d_A(\mathbf{h}_i, \mathbf{h}_j)^2 &= (\mathbf{h}_i - \mathbf{h}_j)^\top A\,(\mathbf{h}_i - \mathbf{h}_j) \\ &= (B\,(\mathbf{h}_i - \mathbf{h}_j))^\top (B(\mathbf{h}_i - \mathbf{h}_j)) \\ &= \|B(\mathbf{h}_i - \mathbf{h}_j)\|_2^2. \end{aligned} \quad [4]$$

The above distance focuses on the aspects of $\mathbb{R}^d$ that have high dot product with the rows of B.

We want to find a linear transformation B such that syntax trees' path metrics are best approximated by the squared[#] distance $\|\cdot\|_B^2$, on internal representations. We approximate this by finding B such that it best recreates all pairs of distances for sentences in a dataset of sentences annotated with syntax trees by humans. We consider each sentence s^ℓ, which has tree structure T^ℓ. This tree defines a distance $d_{T^\ell}(w_i^\ell, w_j^\ell)$ for each pair of words (w_i^ℓ, w_j^ℓ) in the sentence. Recalling that the internal representations of w_i^ℓ and w_j^ℓ are denoted $\mathbf{h}_i^\ell$ and $\mathbf{h}_j^\ell$, we want the difference between $d_{T^\ell}(w_i^\ell, w_j^\ell)$ and $\|B(h_i^\ell, h_j^\ell)\|_2^2$ to be as small as possible in expectation. We implement this by averaging over all sentences s^ℓ in the human-parsed corpus and normalize each sentence's influence by the number of pairs of words in the sentence, $|s^\ell|^2$. This leads to the following optimization problem for finding B:

$$\arg\min_B \sum_\ell \frac{1}{|s^\ell|^2} \sum_{i,j} \left| d_{T^\ell}(w_i^\ell, w_j^\ell) - \|B(\mathbf{h}_i^\ell, \mathbf{h}_j^\ell)\|_2^2 \right|. \quad [5]$$

We approximate this objective through gradient descent to find B.

Once we have a distance metric that approximates tree distance, we can predict trees on new sentences and compare them to human-annotated trees. Recall that given the true tree distances d_T, we can reconstruct the tree by constructing an edge between each pair of words with distance 1. This is equivalent to taking the minimum spanning tree of words in the distance space. To construct trees for a new sentence, we similarly predict distances for all pairs of words and construct the minimum-spanning tree according to the predicted distances (Fig. 7*C*).

So far in this section, we have discussed syntax trees as being undirected, an assumption we make so that we can view them as distance metrics. However, dependency syntax trees are rooted, and all edges are directed such that every edge moves from the word closer to the root to the word farther away. The directions of these edges reflect which word is governed by the other.

Since distances are unordered functions, we cannot use them to model directed edges; to do this we use a slight variation on the distance probe. We first connect edge orderings and vector spaces through the concept of a norm—a notion of length in a space. Edge orderings in a rooted tree follow the depth norm $\|w_i\|_{\text{depth}}$, equal to the number of edges between a word and the root of the tree. Given that two words have an edge between them, then the word with the greater norm has the edge pointing to it.

Viewing edge orderings as resulting from a norm again helps us connect tree structures to the vector space that internal representations reside in. Like distance metrics, norms are a natural way to look at global properties of a vector space. Our goal is now to find a norm on $\mathbb{R}^d$ such that the norm on the vector space approximates the tree depth norm. From here the method is similar to finding the distance metric. We consider an L_2 norm parameterized by a matrix B, that is, $\|B\mathbf{h}_i\|$, and attempt to find B to make the norm best fit the tree norm in expectation over the corpus:

$$\arg\min_B \sum_\ell \frac{1}{|s^\ell|} \sum_i \left| \|w_i^\ell\|_{\text{depth}} - \|B\mathbf{h}_i^\ell\|_2^2 \right|. \quad [6]$$

Experiments. We analyze the internal representations of BERT, keeping the same dataset as our analysis of attention weights. Because of the supervised nature of the approach, we train our distance metric on the training split of the dataset (sections 2 to 21) and report all tree reconstruction accuracies on the testing split (section 23) of ref. 35.

We evaluate the syntactic distances predicted by the distance structural probe in two ways. The first metric is the undirected unlabeled attachment score (UUAS), which is the fraction of edges in the true syntax tree that are also in the predicted minimum-spanning tree. The second one is the distance Spearman correlation (DSpr.), which measures how close the predicted distances are to recreating the ordering of distances between all pairs of words. We evaluate the syntax depths

[¶]Strictly, since A is positive semidefinite instead of positive definite, the result is a semimetric.

[#]We found that squared L_2 distance performs substantially better than the true distance at approximating tree distance (42), an observation explored in more depth by ref. 43. If a valid distance metric is desired, using the square root of all predictions results in the same relative distances.

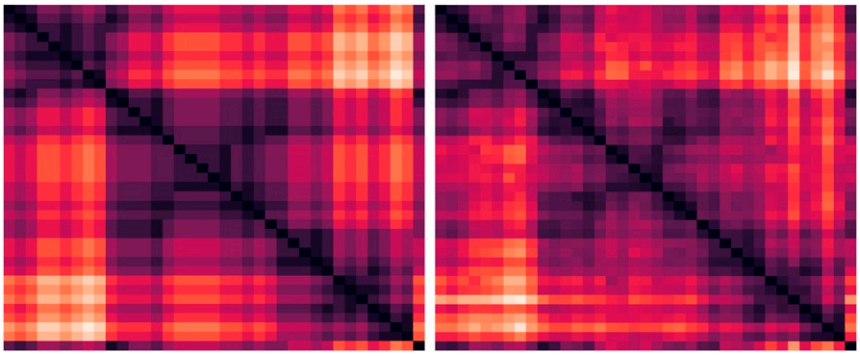


Fig. 9. (*Left*) Matrix showing gold tree distances between all pairs of words in a sentence, whose linear order runs top to bottom and left to right. Darker colors indicate closer words. (*Right*) The same distances as embedded by BERT (squared). Reprinted with permission from ref. 42, which is licensed under CC BY 4.0.

predicted by a depth structural probe by evaluating the root%, the percentage of tree roots predicted to be the least deep. We also evaluate the norm Spearman correlation (NSpr.), which measures how close the predicted depths are to recreating the ordering of depths defined by the tree.

Because structural probes are supervised models, we want to ensure that the probes themselves are not simply learning to parse on top of BERT, instead of revealing properties of BERT. To do this, we define Proj0 (projection of layer 0), a neural network with random weights that takes in a sequence of noncontextual word representations and outputs a sequence of contextual word representations like the ones we evaluate in BERT. We train and evaluate structural probes on this model's representations as a baseline against which to compare BERT.[‖]

Results. We find that dependency tree structures are embedded in BERT representations to a striking extent.[**]

These results are visualized in Figs. 8 and 9. In Table 3, we report our evaluation metrics for BERT, which significantly outperform in terms of UUAS (82.5) compared to our random representation control (59.8) and a baseline that attaches each adjacent word from left to right (48.9).

Likewise for parse depth, we find that the structural probe norm reconstructs the dependency tree edge directions to a substantial extent. Fig. 10 shows how predicted depths according to the probe recreate the depths in a true parse tree. Quantitatively (Table 3), the structural probe on BERT correctly identifies the root of the sentence 90.1% of the time, above the baseline of 64.4%.

Related Emergent Properties

Our work here studying syntax and coreference is part of a growing body of work exploring what linguistic properties emerge in BERT and related representations. Recent work on deep contextual word representations follows an earlier literature examining distributed representations of word types, with models including latent semantic analysis (44) and neural network models such as word2vec (12) and GloVe (Global Vectors) (13). These models and investigation of them focused much more on lexical semantics. In this section, we focus on probing methods, where a simple supervised model is trained to predict a linguistic property from a fixed representation being examined. We review insights gained not only about BERT, but also about similar neural networks such as ELMo (Embeddings from Language Models) (17).

Syntax. Predating our attention and structural probes, early work by Shi et al. (45) introduced the probing task of predicting the label of the smallest phrasal constituent above each word in the tree using its representation (similar to "Is this word in a noun phrase, a verb phrase, or other?"). This method has been extended (46, 47) to predicting the label of the dependency edge governing a word (e.g., nominal subject, direct object, etc.), the label of the edge governing the word's parent, and so on. Predicting these labels requires information about how a word takes part in the meaning of a sentence at multiple levels of hierarchy. Separately, it has been shown that the presence of individual dependency edges can be predicted from probes on pairs of word representations (47, 48).

Parts of Speech. Word tokens are categorized into classes like nouns, adjectives, and adverbs. The same word can have different parts of speech depending on context, as in "the guard sounds (verb) the alarm upon hearing sounds (noun)." Several studies use part-of-speech tagging as a probing task (48–51) and demonstrate that BERT and related models encode information necessary for predicting the part of speech of each word in context from its vector representation. Considerable part-of-speech information was also available from context-free, word type-level embeddings like word2vec and GloVe (12, 13), as shown by earlier probing work (52), but the more recent studies additionally demonstrate the ability of contextual word representations to encode information about the rest of the sequence useful for disambiguating the part of speech in context, as needed for the above example.

Morphology. Morphology concerns how the form and meaning of words are derived from subword components, like how "runs" is an inflected form of run, or "silversmith" can be decomposed into silver + smith. Morphological tagging requires the prediction of properties like singular and third person for runs. Such tagging is prevalent in languages with rich morphology like Turkish, where one would predict definite, locative, singular for *garajın* (of the garage). Conneau et al. (53) found that NLP systems encode information about the number (singular, plural)

Table 3. Results of structural probes on the WSJ SD test set (baselines in the top half, models hypothesized to encode syntax below)

	Distance		Depth	
Method	UUAS (%)	DSpr.	Root%	NSpr.
Linear	48.9	0.58	2.9	0.27
Proj0	59.8	0.73	64.4	0.75
BERT	81.7	0.87	90.1	0.89

[‖] See ref. 42 for more details on these evaluation metrics and Proj0.

[**] We determined and report the best results, taken from the 16th layer of the original Google BERT large cased model (24 layers, 1,024-dimensional hidden layers, 16 attention heads per layer).

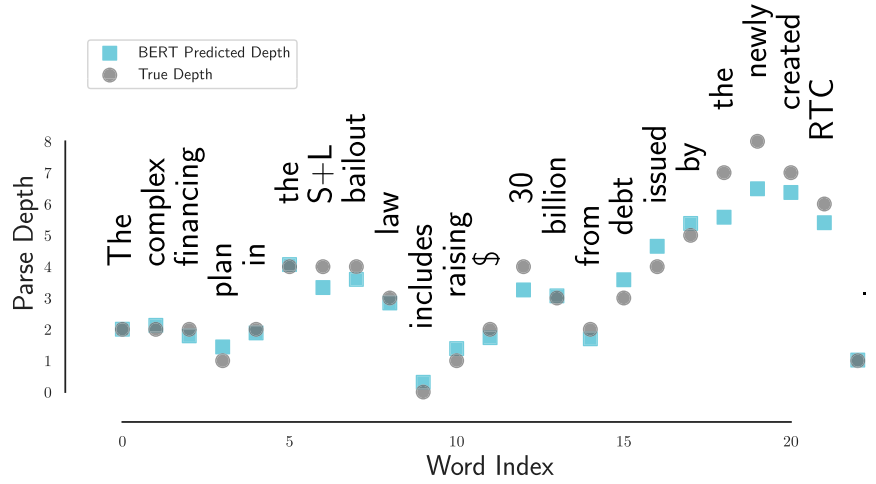


Fig. 10. A sentence with human-annotated parse depths (gray circles) and BERT-predicted parse depths (blue squares). Reprinted with permission from ref. 42, which is licensed under CC BY 4.0.

of the subjects and objects of a sentence even when one represents the whole sentence as a single vector (as opposed to using the vector representation of the subject and object to do so). Belinkov et al. (54) ran probing experiments on neural machine translation systems (like Google Translate), showing that they implicitly learn a considerable amount of information about the morphology of both the input language and the output language.

Semantics. Semantics refers to how language structures encode information about events and ideas. Probing work has shown that BERT and similar systems encode in each word representation information about the semantic role of each word in the sentence, like agent (the "doer" of an action) and patient (the thing the action is done to) (47, 48). There is also evidence that finer-grained attributes, like whether a doer was aware it did the action, also seem to be encoded (48).[‡‡]

Named entity recognition (NER) is a word-tagging task (like part-of-speech tagging) which labels words as being part of a type of named entity or not. For example, Apple would be labeled organization, while Tim and Cook would be labeled person. Multiple studies have shown that the word representations of BERT encode information about the category of words in NER (47, 48).

[‡‡]Results cover semantic dependencies (57), semantic role labeling, and semantic protoroles (58).

Conclusion

We have demonstrated the surprising extent to which BERT, an NLP representation learner trained via self-supervision on word prediction tasks, implicitly learns to recover the rich latent structure of human language. This result has been demonstrated in attention: how BERT looks at sentential context for encoding a word. Separately, we found a similar result through structural probes on internal vector representations, showing that the hierarchical tree structures of language emerge in BERT vector space. That such rich information emerges through self-supervision is surprising and exciting, with intriguing implications for both NLP research and the logical problem of language acquisition.

Data Availability. Code and most of the data to reproduce the analyses in this paper are freely available at https://github.com/clarkkev/attention-analysis and https://github.com/john-hewitt/structural-probes (55, 56); the BERT models are freely available at https://github.com/google-research/bert (20). The syntactic evaluations use the Penn Treebank-3 (PTB), which is available under license from the Linguistic Data consortium at https://catalog.ldc.upenn.edu/LDC99T42 (35).

ACKNOWLEDGMENTS. We thank Tatsunori B. Hashimoto, Abigail See, Siva Reddy, Drew A. Hudson, Roma Patel, Percy Liang, and the PNAS reviewers for helpful comments. K.C. was supported by a Google Fellowship. J.H. and C.D.M. were partly funded by a gift from Tencent Corp.

1. P. K. Kuhl, Early language acquisition: Cracking the speech code. *Nat. Rev. Neurosci.* **5**, 831–843 (2004).
2. O. Rambow, "The simple truth about dependency and phrase structure representations: An opinion piece" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, R. Kaplan, J. Burstein, M. Harper, G. Penn, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2010), pp. 337–340.
3. Z. Pizlo, Perception viewed as an inverse problem. *Vis. Res.* **41**, 3145–3161 (2001).
4. M. P. Marcus, B. Santorini, M. A. Marcinkiewicz, Building a large annotated corpus of English: The Penn treebank. *Comput. Ling.* **19**, 313–330 (1993).
5. J. Nivre *et al.*, "Universal dependencies V1: A multilingual treebank collection" in *LREC International Conference on Language Resources and Evaluation*, N. Calzolari *et al.*, Eds. (European Language Resources Association, Paris, France, 2016), pp. 1659–1666.
6. M. Collins, Head-driven statistical models for natural language parsing. *Comput. Ling.* **29**, 589–637 (2003).
7. D. Chen, C. D. Manning, "A fast and accurate dependency parser using neural networks" in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, A. Moschitti, B. Pang, W. Daelemans, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2014), pp. 740–750.
8. T. Dozat, C. D. Manning, "Deep biaffine attention for neural dependency parsing." https://openreview.net/pdf?id=Hk95PK9le. Accessed 21 May 2020.
9. J. Schmidhuber, "An on-line algorithm for dynamic reinforcement learning and planning in reactive environments" in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (Institute of Electrical and Electronic Engineers, Piscataway, NJ, 1990), pp. 253–258.
10. D. Lieb, A. Lookingbill, S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow" in *Proceedings of Robotics: Science and Systems (RSS)*, S. Thrun, G. S. Sukhatme, S. Schaal, Eds. (MIT Press, Cambridge, MA, 2005), pp. 273–280.
11. W. L. Taylor, Cloze procedure: A new tool for measuring readability. *Journal. Q.* **30**, 415–433 (1953).
12. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, "Distributed representations of words and phrases and their compositionality" in *Advances Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger, Eds. (Curran Associates, Red Hook, NY, 2013), pp. 3111–3119.
13. J. Pennington, R. Socher, C. Manning, "Glove: Global vectors for word representation" in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, A. Moschitti, B. Pang, W. Daelemans, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2014), pp. 1532–1543.
14. Y. Bengio, Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2**, 1–127 (2009).
15. R. C. Berwick, P. Pietroski, B. Yankama, N. Chomsky, Poverty of the stimulus revisited. *Cognit. Sci.* **35**, 1207–1242 (2011).

16. T. L. Griffiths, Rethinking language: How probabilities shape the words we use. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 3825–3826 (2011).
17. M. Peters *et al.*, "Deep contextualized word representations" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Walker, H. Ji, A. Stent, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 2227–2237.
18. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, J. Burstein, C. Doran, T. Solorio, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2019), pp. 4171–4186.
19. N. Chomsky, *Knowledge of Language: Its Nature, Origin, and Use* (Praeger, New York, NY, 1986).
20. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT. https://github.com/google-research/bert. Accessed 14 May 2020.
21. A. Vaswani *et al.*, "Attention is all you need" in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. (Curran Associates, Red Hook, NY, 2017), pp. 5998–6008.
22. J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 (11 Dececember 2014).
23. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 (16 January 2019) .
24. T. Linzen, E. Dupoux, Y. Goldberg, Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguist.* **4**, 521–535 (2016).
25. K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, M. Baroni, "Colorless green recurrent networks dream hierarchically" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Walker, H. Ji, A. Stent, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 1195–1205.
26. R. Marvin, T. Linzen, "Targeted syntactic evaluation of language models" in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 1192–1202.
27. A. Kuncoro *et al.*, "LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, I. Gurevych, Y. Miyao, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 1426–1436.
28. Y. Goldberg, Assessing BERT's syntactic abilities. arXiv:1901.05287 (16 January 2019).
29. K. Bock, C. A. Miller, Broken agreement. *Cognit. Psychol.* **23**, 45–93 (1991).
30. C. Phillips, M. W. Wagers, E. F. Lau, "Grammatical illusions and selective fallibility in real-time language comprehension" in *Experiments at the Interfaces, Syntax and Semantics*, J. Runner, Ed. (Emerald Group Publishing Limited, 2011), vol. 37, pp. 147–180.
31. T. Luong, H. Q. Pham, C. D. Manning, "Effective approaches to attention-based neural machine translation" in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, J. Su, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2015), pp. 1412–1421.
32. S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention. arxiv:1511.04119 (14 February 2016).
33. K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention" in *Proceedings of the International Conference on Machine Learning*, F. Bach, D. Blei, Eds. (Proceedings of Machine Learning Research, Brookline, MA, 2015), pp. 2048–2057.
34. J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, "Attention-based models for speech recognition" in *Advances Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett, Eds. (Curran Associates, Red Hook, NY, 2015), pp. 577–585.
35. M. P. Marcus, B. Santorini, M. A. Marcinkiewicz, A. Taylor, Treebank-3. Linguistic Data Consortium LDC99T42. https://catalog.ldc.upenn.edu/LDC99T42. Accessed 14 May 2020.
36. M. C. de Marneffe, B. MacCartney, C. D. Manning, "Generating typed dependency parses from phrase structure parses" in *LREC International Conference on Language Resources and Evaluation*, N. Calzolari *et al.*, Eds. (European Language Resources Association, Paris, France, 2006), pp. 449–454.
37. S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in Ontonotes" in *Joint Conference on EMNLP and CoNLL – Shared Task*, S. Pradhan, A. Moschitti, N. Xue, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2012), pp. 1–40.
38. H. Lee *et al.*, "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task" in *Proceedings of the Conference on Computational Natural Language Learning: Shared Task*, S. Pradhan, Ed. (Association for Computational Linguistics, Stroudsburg, PA, 2011), pp. 28–34.
39. A. Eriguchi, K. Hashimoto, Y. Tsuruoka, "Tree-to-sequence attentional neural machine translation" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Erk, N. A. Smith, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2016), pp. 823–833.
40. K. Chen, R. Wang, M. Utiyama, E. Sumita, T. Zhao, "Syntax-directed attention for neural machine translation" in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, Palo Alto, CA, 2018), pp. 4792–4799.
41. E. Strubell, P. Verga, D. Andor, D. I. Weiss, A. McCallum, "Linguistically-informed self-attention for semantic role labeling" in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 5027–5038.
42. J. Hewitt, C. D. Manning, "A structural probe for finding syntax in word representations" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, J. Burstein, C. Doran, T. Solorio, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2019), pp. 4129–4138.
43. E. Reif *et al.*, "Visualizing and measuring the geometry of BERT" in *Advances in Neural Information Processing Systems 32*, H. Wallach et al., Eds. (Curran Associates, Red Hook, NY, 2019), pp. 8594–8603.
44. T. K. Landauer, S. T. Dumais, A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **104**, 211–240 (1997).
45. X. Shi, I. Padhi, K. Knight, "Does string-based neural MT learn source syntax?" in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, X. Carreras, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2016), pp. 1526–1534.
46. T. Blevins, O. Levy, L. Zettlemoyer, "Deep RNNs encode soft hierarchical syntax" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, I. Gurevych, Y. Miyao, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 14–19.
47. N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, N. A. Smith, "Linguistic knowledge and transferability of contextual representations" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, J. Burstein, C. Doran, T. Solorio, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2019), pp. 1073–1094.
48. I. Tenney *et al.*, "What do you learn from context? Probing for sentence structure in contextualized word representations." https://openreview.net/pdf?id=SJzSgnRcKX. Accessed 21 May 2020.
49. M. Peters, M. Neumann, L. Zettlemoyer, W. T. Yih, "Dissecting contextual word embeddings: Architecture and representation" in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 1499–1509.
50. N. Saphra, A. Lopez, "Understanding learning dynamics of language models with SVCCA" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, J. Burstein, C. Doran, T. Solorio, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2019), pp. 3257–3267.
51. K. W. Zhang, S. R. Bowman, "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis" in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 359–361.
52. A. Kõhn, "What's in an embedding? Analyzing word embeddings through multilingual evaluation" in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, J. Su, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2015), pp. 2067–2073.
53. A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, "What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, I. Gurevych, Y. Miyao, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2018), pp. 2126–2136.
54. Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, J. Glass, "What do neural machine translation models learn about morphology?" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, R. Barzilay, M.-Y. Kan, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2017), pp. 861–872.
55. K. Clark, BERT attention analysis. https://github.com/clarkkev/attention-analysis. Deposited 27 June 2019.
56. J. Hewitt, Structural probes. https://github.com/john-hewitt/structural-probes. Deposited 27 May 2019.
57. S. Oepen *et al.*, "SemEval 2014 task 8: Broad-coverage semantic dependency parsing" in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, P. Nakov, T. Zesch, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2014), pp. 63–72.
58. D. Reisinger *et al.*, Semantic proto-roles. *Trans. Assoc. Comput. Linguist.* **3**, 475–488 (2015).
59. K. Clark, U. Khandelwal, O. Levy, C. D. Manning, "What does BERT look at? An analysis of BERT's attention" in *Proceedings of the Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, T. Linzen, G. Chrupała, Y. Belinkov, D. Hupkes, Eds. (Association for Computational Linguistics, Stroudsburg PA, 2019), pp. 276–286.