What does BERT learn about the structure of language?

Ganesh Jawahar **Benoît Sagot** **Djamé Seddah**
Inria, France
{firstname.lastname}@inria.fr

Abstract

BERT is a recent language representation model that has surprisingly performed well in diverse language understanding benchmarks. This result indicates the possibility that BERT networks capture structural information about language. In this work, we provide novel support for this claim by performing a series of experiments to unpack the elements of English language structure learned by BERT. We first show that BERT's phrasal representation captures ① phrase-level information in the lower layers. We also show that BERT's intermediate layers encode a rich ② hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle and semantic features at the top. BERT turns out to require deeper layers when ③ long-distance dependency information is required, e.g. to track subject-verb agreement. Finally, ④ we show that BERT representations capture linguistic information in a compositional way that mimics classical, tree-like structures.

模仿

1 Introduction

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) is a bidirectional variant of Transformer networks (Vaswani et al., 2017) trained to jointly predict a masked word from its context and to classify whether two sentences are consecutive or not. The trained model can be fine-tuned for downstream NLP tasks such as question answering and language inference without substantial modification. BERT outperforms previous state-of-the-art models in the eleven NLP tasks in the GLUE benchmark (Wang et al., 2018) by a significant margin. This remarkable result suggests that BERT could "learn" structural information about language.

Can we unveil the representations learned by BERT to proto-linguistics structures? Answering this question could not only help us understand the reason behind the success of BERT but also its limitations, in turn guiding the design of improved architectures. This question falls under the topic of the interpretability of neural networks, a growing field in NLP (Belinkov and Glass, 2019). An important step forward in this direction is Goldberg (2019), which shows that BERT captures syntactic phenomena well when evaluated on its ability to track subject-verb agreement.

In this work, we perform a series of experiments to probe the nature of the representations learned by different layers of BERT.[1] We first show that the lower layers capture phrase-level information, which gets diluted in the upper layers. Second, we propose to use the probing tasks defined in Conneau et al. (2018) to show that BERT captures a rich hierarchy of linguistic information, with surface features in lower layers, syntactic features in middle layers and semantic features in higher layers. Third, we test the ability of BERT representations to track subject-verb agreement and find that BERT requires deeper layers for handling harder cases involving long-distance dependencies. Finally, we propose to use the recently introduced Tensor Product Decomposition Network (TPDN) (McCoy et al., 2019) to explore different hypotheses about the compositional nature of BERT's representation and find that BERT implicitly captures classical, tree-like structures.

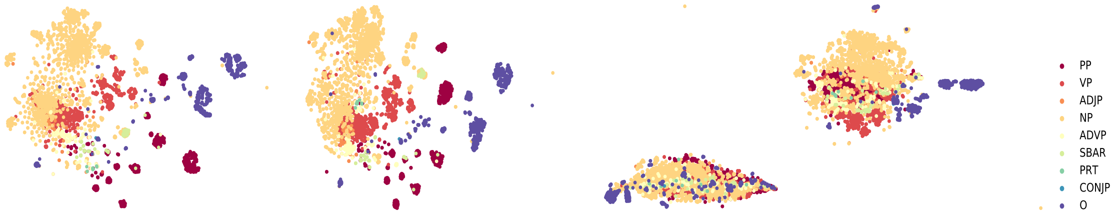
4个实验，4项结果

2 BERT

BERT (Devlin et al., 2018) builds on Transformer networks (Vaswani et al., 2017) to pre-train bidirectional representations by conditioning on both left and right contexts jointly in all layers. The representations are jointly optimized by predicting randomly masked words in the input and classify-

[1] The code to reproduce our experiments is publicly accessible at https://github.com/ganeshjawahar/interpret_bert

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3651–3657
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics



(a) Layer 1 (b) Layer 2 (c) Layer 11 (d) Layer 12

Figure 1: $2D$ t-SNE plot of span embeddings computed from the first and last two layers of BERT.

layer	1	2	3	4	5	6	7	8	9	10	11	12
NMI	**0.38**	0.37	0.35	0.3	0.24	0.2	0.19	0.16	0.17	0.18	0.16	0.19

Table 1: Clustering performance of span representations obtained from different layers of BERT.

ing whether the sentence follows a given sentence in the corpus or not. The authors of BERT claim that bidirectionality allows the model to swiftly adapt for a downstream task with little modification to the architecture. Indeed, BERT improved the state-of-the-art for a range of NLP benchmarks (Wang et al., 2018) by a significant margin.

In this work, we investigate the linguistic structure implicitly learned by BERT's representations. We use the PyTorch implementation of BERT, which hosts the models trained by (Devlin et al., 2018). All our experiments are based on the `bert-base-uncased` variant,[2] which consists of 12 layers, each having a hidden size of 768 and 12 attention heads ($110M$ parameters). In all our experiments, we seek the activation of the first input token ('`[CLS]`') (which summarizes the information from the actual tokens using a self-attention mechanism) at every layer to compute BERT representation, unless otherwise stated.

3 Phrasal Syntax

Peters et al. (2018) have shown that the representations underlying LSTM-based language models(Hochreiter and Schmidhuber, 1997) can capture phrase-level (or span-level) information.[3] It remains unclear if this holds true for models not trained with a traditional language modeling objective, such as BERT. Even if it does, would the information be present in multiple layers of the model? To investigate this question we extract span representations from each layer of BERT. Following Peters et al. (2018), for a token sequence $s_i, \ldots, s_j$, we compute the span representation $\mathbf{s}_{(s_i,s_j),l}$ at layer l by concatenating the first ($\mathbf{h}_{s_i,l}$) and last hidden vector ($\mathbf{h}_{s_j,l}$), along with their element-wise product and difference. We randomly pick 3000 labeled chunks and 500 spans not labeled as chunks from the CoNLL 2000 chunking dataset (Sang and Buchholz, 2000).

实验设置

As shown in Figure 1, we visualize the span representations obtained from multiple layers using t-SNE (Maaten and Hinton, 2008), a non-linear dimensionality reduction algorithm for visualizing high-dimensional data. We observe that BERT mostly captures phrase-level information in the lower layers and that this information gets gradually diluted in higher layers. The span representations from the lower layers map chunks (e.g. 'to demonstrate') that project their underlying category (e.g. VP) together. We further quantify this claim by performing a k-means clustering on span representations with $k = 10$, i.e. the number of distinct chunk types. Evaluating the resulting clusters using the Normalized Mutual Information (NMI) metric shows again that the lower layers encode phrasal information better than higher layers (cf. Table 1).

结论

4 Probing Tasks

Probing (or diagnostic) tasks (Adi et al., 2017; Hupkes et al., 2018; Conneau et al., 2018) help in unearthing 发现 the linguistic features possibly encoded in neural models. This is achieved by setting up an auxiliary classification task where the final output of a model is used as features to predict a linguistic phenomenon of interest. If the auxiliary classifier can predict a linguistic prop-

[2] We obtained similar results in preliminary experiments with the `bert-large-uncased` variant.

[3] Peters et al. (2018) experimented with ELMo-style CNN and Transformer but did not report this finding for these models.

Layer	SentLen (Surface)	WC (Surface)	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
1	93.9 (2.0)	24.9 (24.8)	35.9 (6.1)	63.6 (9.0)	50.3 (0.3)	82.2 (18.4)	77.6 (10.2)	76.7 (26.3)	49.9 (-0.1)	53.9 (3.9)
2	95.9 (3.4)	65.0 (64.8)	40.6 (11.3)	71.3 (16.1)	55.8 (5.8)	85.9 (23.5)	82.5 (15.3)	80.6 (17.1)	53.8 (4.4)	58.5 (8.5)
3	**96.2 (3.9)**	66.5 (66.0)	39.7 (10.4)	71.5 (18.5)	64.9 (14.9)	86.6 (23.8)	82.0 (14.6)	80.3 (16.6)	55.8 (5.9)	59.3 (9.3)
4	94.2 (2.3)	**69.8 (69.6)**	39.4 (10.8)	71.3 (18.3)	74.4 (24.5)	87.6 (25.2)	81.9 (15.0)	81.4 (19.1)	59.0 (8.5)	58.1 (8.1)
5	92.0 (0.5)	69.2 (69.0)	40.6 (11.8)	81.3 (30.8)	81.4 (31.4)	89.5 (26.7)	85.8 (19.4)	81.2 (18.6)	60.2 (10.3)	64.1 (14.1)
6	88.4 (-3.0)	63.5 (63.4)	**41.3 (13.0)**	83.3 (36.6)	82.9 (32.9)	89.8 (27.6)	**88.1 (21.9)**	82.0 (20.1)	60.7 (10.2)	71.1 (21.2)
7	83.7 (-7.7)	56.9 (56.7)	40.1 (12.0)	**84.1 (39.5)**	83.0 (32.9)	89.9 (27.5)	87.4 (22.2)	**82.2 (21.1)**	61.6 (11.7)	74.8 (24.9)
8	82.9 (-8.1)	51.1 (51.0)	39.2 (10.3)	84.0 (39.5)	83.9 (33.9)	89.9 (27.6)	87.5 (22.2)	81.2 (19.7)	62.1 (12.2)	76.4 (26.4)
9	80.1 (-11.1)	47.9 (47.8)	38.5 (10.8)	83.1 (39.8)	**87.0 (37.1)**	**90.0 (28.0)**	87.6 (22.9)	81.8 (20.5)	63.4 (13.4)	**78.7 (28.9)**
10	77.0 (-14.0)	43.4 (43.2)	38.1 (9.9)	81.7 (39.8)	86.7 (36.7)	89.7 (27.6)	87.1 (22.6)	80.5 (19.9)	63.3 (12.7)	78.4 (28.1)
11	73.9 (-17.0)	42.8 (42.7)	36.3 (7.9)	80.3 (39.1)	86.8 (36.8)	89.9 (27.8)	85.7 (21.9)	78.9 (18.6)	64.4 (14.5)	77.6 (27.9)
12	69.5 (-21.4)	49.1 (49.0)	34.7 (6.9)	76.5 (37.2)	86.4 (36.4)	89.5 (27.7)	84.0 (20.2)	78.7 (18.4)	**65.2 (15.3)**	74.9 (25.4)

Table 2: Probing task performance for each BERT layer. The value within the parentheses corresponds to the difference in performance of trained vs. untrained BERT.

Layer	**0** (1.5)	**1** (5.2)	**2** (7.7)	**3** (10.5)	**4** (13.3)
1	90.89	40.43	23.22	21.46	20
2	92.01	42.6	25.84	24.78	26.02
3	92.77	47.05	29.77	27.22	29.56
4	94.39	52.97	33.02	29.13	30.09
5	94.98	63.12	43.68	36.61	36.11
6	95.45	67.28	46.93	38.22	36.46
7	95.52	72.44	53.03	43.5	41.06
8	**95.68**	**75.66**	**58.74**	48.88	45.49
9	95.54	73.84	57.96	**50.34**	**48.85**
10	95.09	69.21	51.5	43.26	41.59
11	94.33	66.62	51.69	46.09	42.65
12	94.06	62.78	51.07	46.04	46.37

Table 3: Subject-verb agreement scores for each BERT layer. The last five columns correspond to the number of nouns intervening between the subject and the verb (attractors) in test instances. The average distance between the subject and the verb is enclosed in parentheses next to each attractor category.

erty well, then the original model likely encodes that property. In this work, we use probing tasks to assess individual model layers in their ability to encode different types of linguistic features. We evaluate each layer of BERT using ten probing sentence-level datasets/tasks created by Conneau et al. (2018), which are grouped into three categories. Surface tasks probe for sentence length (SentLen) and for the presence of words in the sentence (WC). Syntactic tasks test for sensitivity to word order (BShift), the depth of the syntactic tree (TreeDepth) and the sequence of top-level constituents in the syntax tree (TopConst). Semantic tasks check for the tense (Tense), the subject (resp. direct object) number in the main clause (SubjNum, resp. ObjNum), the sensitivity to random replacement of a noun/verb (SOMO) and the random swapping of coordinated clausal conjuncts (CoordInv). We use the SentEval toolkit (Conneau and Kiela, 2018) along with the recommended hyperparameter space to search for the best probing classifier. As random encoders can surprisingly encode a lot of lexical and structural information (Zhang and Bowman, 2018), we also evaluate the untrained version of BERT, obtained by setting all model weights to a random number.

连词

对比

Table 2 shows that BERT embeds a rich hierarchy of linguistic signals: surface information at the bottom, syntactic information in the middle, semantic information at the top. BERT has also surpassed the previously published results for two tasks: BShift and CoordInv. We find that the untrained version of BERT corresponding to the higher layers outperforms the trained version in the task of predicting sentence length (SentLen). This could indicate that untrained models contain sufficient information to predict a basic surface feature such as sentence length, whereas training the model results in the model storing more complex information, at the expense of its ability to predict such basic surface features.

结论1

结论2

5 Subject-Verb Agreement

Subject-verb agreement is a proxy task to probe whether a neural model encodes syntactic structure (Linzen et al., 2016). The task of predicting the verb number becomes harder when there are more nouns with opposite number (attractors) intervening between the subject and the verb. Goldberg (2019) has shown that BERT learns syntactic phenomenon surprisingly well using various stimuli for subject-verb agreement. We extend his work by performing the test on each layer of BERT and controlling for the number of attractors. In our study, we use the stimuli created by Linzen et al. (2016) and the SentEval toolkit (Conneau and Kiela, 2018) to build the binary classifier with the recommended hyperparameter space, using as features the activations from the (masked) verb at hand.

Role scheme \ Layer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Left-to-right	0.0005	**0.0007**	0.0008	0.0034	0.0058	0.0087	0.0201	0.0179	0.0284	0.0428	0.0362	0.0305
Right-to-left	**0.0004**	**0.0007**	**0.0007**	**0.0032**	0.0060	0.0099	0.0233	0.0203	0.0337	0.0486	0.0411	0.0339
Bag-of-words	0.0006	0.0009	0.0012	0.0039	0.0066	0.0108	0.0251	0.0221	0.0355	0.0507	0.0422	0.0348
Bidirectional	0.0025	0.0030	0.0034	0.0053	0.0079	0.0106	0.0226	0.0201	0.0311	0.0453	0.0391	0.0334
Tree	0.0005	0.0009	0.0011	0.0037	**0.0055**	**0.0081**	**0.0179**	**0.0155**	**0.0249**	**0.0363**	**0.0319**	**0.0278**
Tree (random)	0.0005	0.0009	0.0011	0.0038	0.0063	0.0099	0.0237	0.0214	0.0338	0.0486	0.0415	0.0340

Table 4: Mean squared error between TPDN and BERT representation for a given layer and role scheme on SNLI test instances. Each number corresponds to the average across five random initializations.

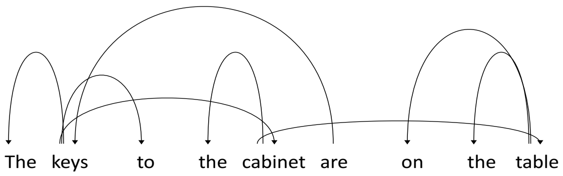


Figure 2: Dependency parse tree induced from attention head #11 in layer #2 using gold root ('are') as starting node for maximum spanning tree algorithm.

结论1

结论2

Results in Table 3 show that the middle layers perform well in most cases, which supports the result in Section 4 where the syntactic features were shown to be captured well in the middle layers. Interestingly, as the number of attractors increases, one of the higher BERT layers (#8) is able to handle the long-distance dependency problems caused by the longer sequence of words intervening between the subject and the verb, better than the lower layer (#7). This highlights the need for BERT to have deeper layers to perform competitively on NLP tasks.

6 Compositional Structure

Can we understand the compositional nature of representation learned by BERT, if any? To investigate this question, we use Tensor Product Decomposition Networks (TPDN) (McCoy et al., 2019), which explicitly compose the input token ("filler") representations based on the role scheme selected beforehand using tensor product sum. For instance, a role scheme for a word can be based on the path from the root node to itself in the syntax tree (e.g. 'LR' denotes the right child of left child of root). The authors assume that, for a given role scheme, if a TPDN can be trained well to approximate the representation learned by a neural model, then that role scheme likely specifies the compositionality implicitly learned by the model. For each BERT layer, we work with five different role schemes. Each word's role is computed based on its left-to-right index, its right-to-left index, an ordered pair containing its left-to-right and right-to-left indices, its position in a syntactic tree (formatted version of the Stanford PCFG Parser (Klein and Manning, 2003) with no unary nodes and no labels) and an index common to all the words in the sentence (bag-of-words), which ignores its position. Additionally, we also define a role scheme based on random binary trees.

结论1

Following McCoy et al. (2019), we train our TPDN model on the premise sentences in the SNLI corpus (Bowman et al., 2015). We initialize the filler embeddings of the TPDN with the pre-trained word embeddings from BERT's input layer, freeze it, learn a linear projection on top of it and use a Mean Squared Error (MSE) loss function. Other trainable parameters include the role embeddings and a linear projection on top of tensor product sum to match the embedding size of BERT. Table 4 displays the MSE between representation from pretrained BERT and representation from TPDN trained to approximate BERT. We discover that BERT implicitly implements a tree-based scheme, as a TPDN model following that scheme best approximates BERT's representation at most layers. This result is remarkable, as BERT encodes classical, tree-like structures despite relying purely on attention mechanisms.

Motivated by this study, we perform a case study on dependency trees induced from self attention weight following the work done by Raganato and Tiedemann (2018). Figure 2 displays the dependencies inferred from an example sentence by obtaining self attention weights for every word pairs from attention head #11 in layer #2, fixing the gold root as the starting node and invoking the Chu-Liu-Edmonds algorithm (Chu and Liu, 1967). We observe that determiner-noun dependencies ("the keys", "the cabinet" and "the table") and subject-verb dependency ("keys" and "are") are captured accurately. Surprisingly, the predicate-argument structure seems to be partly modeled as shown by the chain of dependencies between "key","cabinet" and "table".

7 Related Work

Peters et al. (2018) studies how the choice of neural architecture such as CNNs, Transformers and RNNs used for language model pretraining affects the downstream task accuracy and the qualitative properties of the contextualized word representations that are learned. They conclude that all architectures learn high quality representations that outperform standard word embeddings such as GloVe (Pennington et al., 2014) for challenging NLP tasks. They also show that these architectures hierarchically structure linguistic information, such that morphological, (local) syntactic and (longer range) semantic information tend to be represented in, respectively, the word embedding layer, lower contextual layers and upper layers. In our work, we observe that such hierarchy exists as well for BERT models that are not trained using the standard language modelling objective. Goldberg (2019) shows that the BERT model captures syntactic information well for subject-verb agreement. We build on this work by performing the test on each layer of BERT controlling for the number of attractors and then show that BERT requires deeper layers for handling harder cases involving long-distance dependency information.

Tenney et al. (2019) is a contemporaneous work that introduces a novel edge probing task to investigate how contextual word representations encode sentence structure across a range of syntactic, semantic, local and long-range phenomena. They conclude that contextual word representations trained on language modeling and machine translation encode syntactic phenomena strongly, but offer comparably small improvements on semantic tasks over a non-contextual baseline. Their result using BERT model on capturing linguistic hierarchy confirms our probing task results although using a set of relatively simple probing tasks. Liu et al. (2019) is another contemporaneous work that studies the features of language captured/missed by contextualized vectors, transferability across different layers of the model and the impact of pretraining on the linguistic knowledge and transferability. They find that (i) contextualized word embeddings do not capture fine-grained linguistic knowledge, (ii) higher layers of RNN to be task-specific (with no such pattern for a transformer) and (iii) pretraining on a closely related task yields better performance than language model pretraining. Hewitt and Manning (2019) is a very recent work which showed that we can recover parse trees from the linear transformation of contextual word representation consistently, better than with non-contextual baselines. They focused mainly on syntactic structure while our work additionally experimented with linear structures (left-to-right, right-to-left) to show that the compositionality modelling underlying BERT mimics traditional syntactic analysis.

The recent burst of papers around these questions illustrates the importance of interpreting contextualized word embedding models and our work complements the growing literature with additional evidences about the ability of BERT in learning syntactic structures.

8 Conclusion

With our experiments, which contribute to a currently bubbling line of work on neural network interpretability, we have shown that BERT does capture structural properties of the English language. Our results therefore confirm those of Goldberg (2019); Hewitt and Manning (2019); Liu et al. (2019); Tenney et al. (2019) on BERT who demonstrated that span representations constructed from those models can encode rich syntactic phenomena. We have shown that phrasal representations learned by BERT reflect phrase-level information and that BERT composes a hierarchy of linguistic signals ranging from surface to semantic features. We have also shown that BERT requires deeper layers to model long-range dependency information. Finally, we have shown that BERT's internal representations reflect a compositional modelling that shares parallels with traditional syntactic analysis. It would be interesting to see if our results transfer to other domains with higher variability in syntactic structures (such as noisy user generated content) and with higher word order flexibility as experienced in some morphologically-rich languages.

Acknowledgments

We thank Grzegorz Chrupała and our anonymous reviewers for providing insightful comments and suggestions. This work was funded by the ANR projects ParSiTi (ANR-16-CE33-0021), SoSweet (ANR15-CE38-0011-01) and the French-Israeli PHC Maimonide cooperation program.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *International Conference on Learning Representations*.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Yoeng-Jin Chu and Tseng-Hong Liu. 1967. On the shortest arborescence of a directed graph. In *Science Sinica*, pages 1396–1400.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 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 (Volume 1: Long Papers)*, pages 2126–2136. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Computing Research Repository*, arXiv:1810.04805. Version 1.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *Computing Research Repository*, arXiv:1901.05287. Version 1.

John Hewitt and Christopher D. Manning. 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*. Association for Computational Linguistics.

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

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure. *J. Artif. Int. Res.*, 61(1):907–926.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A. Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

R. Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2019. RNNs Implicitly Implement Tensor Product Representations. *International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task Chunking. In *Fourth Conference on Computational Natural Language Learning, CoNLL 2000, and the Second Learning Language in Logic Workshop, LLL 2000, Held in cooperation with ICGI-2000, Lisbon, Portugal, September 13-14, 2000*, pages 127–132.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

Kelly W. Zhang and Samuel R. Bowman. 2018. Language Modeling Teaches You More Syntax than Translation Does: Lessons Learned Through Auxiliary Task Analysis. *Computing Research Repository*, arXiv:1809.10040. Version 2.