Interpretability in Neural Machine Translation

Brian Ondov
University of Maryland
ondovb@umd.edu

Gina Wong
University of Maryland
ginaw15@gmail.com

Jacqueline Nelligan
University of Maryland
jbnell@umd.edu

Yancy Liao
University of Maryland
yancy.liao@gmail.com

Abstract

The Transformer, a deep neural architecture based on self-attention, has quickly become state-of-the-art for many language tasks. However, it is not fully understood how Transformers form internal language models, which is a significant barrier to interpretability. While some syntactic exploration has been performed on pre-trained, self-supervised models (namely BERT), it is not known whether Transformers trained directly on language translation tasks form similar representations during the encoding phase. Here we apply recent methods for analyzing the syntactic properties of attention heads to a Transformer trained end-to-end on English-to-German translation. The results reveal that syntactic properties are concordant for Transformers that are pre-trained or task-specific. Additionally, we train the Transformer to prune heads while maintaining good performance (high BLEU score), and we find that the Transformer learned to retain the syntactically important heads, while pruning the rest.

1 Introduction

1.1 Transformers and attention

The Transformer is an encoder-decoder architecture based around a self-attention mechanism known as "scaled dot-product attention" Vaswani et al. [2017]. For matrices Q, V, K (called the Query, Key, and Value) which are learned projections of our input embeddings,

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{1}$$

In Transformers, there are multiple attention "heads" operating concurrently in each layer. We call this multi-headed self-attention. The redundancy allows different heads to learn different patterns of self-attention. For a single layer,

$$\text{MultiHead}(Q, K, V) = \text{Concat}_i(\text{head}_i)W^O, \tag{2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where index i ranges over the heads in that layer. Concatenation and multiplication by W^O allows us to produce an output of roughly the same dimensions as single-headed attention. The $W_i^Q, W_i^K, W_i^V, W_i^O$ are weight matrices specific to that head. They are parameters learned by the model.

Preprint. Under review.

Transformers were initially developed for machine translation by Vaswani et al. [2017]. They have since been adapted for a variety of language tasks. An important milestone was the 2018 release of BERT (Bidirectional Encoder Representations for Transformers) by Devlin et al. [2019], which showed superior performance in question answering, natural language inference, and paraphrase detection.

In this paper we focus on interpretability at the level of attention heads: whether some heads learn interpretable patterns of self-attention corresponding to syntactic features, and whether some heads can be pruned.

1.2 Previous work on interpretability for language models

There has been a big push in recent years to explore interpretability of machine learning algorithms in general and language models in particular. For language models, this exploration tends to come in two flavors: work that tests models against carefully curated, linguistically interesting test sets (Linzen et al. [2016]; Goldberg [2019]) and work that probes the model's returned encodings to analyze what linguistic information they capture (Raganato et al. [2018]).

Our work builds on prior work primarily from two papers that analyse interpretability of attention heads in Transformer models in particular.

Clark et al. [2019] analyze the role of attention heads in the BERT model for a variety of properties, by treating each head as a simple no-training-required classifier that, given a word as input, outputs the most-attended-to other word. They found that many heads often attend to certain positional offsets (words to the left or right), or to end-of-sentence tokens. More interestingly, they see that some heads seem to encode particular syntatic and semantic relations such as subjects and direct objects of verbs, determiners and adjectives of nouns, and coreference relations. In particular, no one head covers all syntactic dependencies (also found in Raganato et al. [2018]) but several heads perform well on specific dependencies, suggesting that the model as a whole learns this type of structure without explicitly being trained on it.

Voita et al. [2019] analyses the role of attention heads in a Transformer trained on a machine translation task. They tested three models, English-Russian, English-German, and English-French, and analyze heads for focus on positional cues, syntactic cues, or rare words. They find a few heads that attend to positions and a single head that attends strongly to rare words, and several heads that attend to syntactic dependencies. (Their syntactic analysis is over subject, direct object, and adverb and adjective dependencies, though only subject and object heads perform substantially better than the baseline.) They take this analysis one step further by pruning heads from the model in descending order of importance and find that they can prune almost 80% of the heads from the model without substantial drops in the model's BLEU score. Interestingly, the heads with interpretable functions are the last to be pruned from the model, meaning that they are heavily weighted by the model and suggesting that the things that humans attend to during translation align with what the model learns to attend to.

1.3 Our contributions

Transformers can be pre-trained under a self-supervision regime, as in BERT, or from scratch with a supervised, task-specific objective. Models of both paradigms have been investigated for interpretability of heads and effects of model ablation. However, it is not clear whether the results are directly comparable, and thus whether Transformers utilize self-attention in similar ways in different settings. We thus provide syntactic results similar to those obtained for BERT (as by Clark et al. [2019]), but applied to an end-to-end language translation task (as by Voita et al. [2019]).

We also contribute to the prior ablation studies of Voita et al. [2019] with additional studies elucidating the importance of heads deemed important by parametrically learned gates.

Table 1: Best performing attention heads

Relation	Head	Accuracy	Baseline (Offset)
All	2-4	42.1	28.9 (1)
case	2-4	**78.5**	36.4 (2)
nmod	2-7	32.3	27.5 (-3)
det	2-4	**90.5**	55.0 (1)
nsubj	5-2	**63.7**	43.3 (1)
compound	2-4	75.5	73.4 (1)
amod	2-4	82.8	71.6 (1)
advmod	2-4	52.9	45.1 (1)
dobj	4-2	**81.2**	32.3 (-2)
mark	2-4	57.7	47.7 (1)
aux	2-4	**89.8**	58.1 (1)
nmod:pos	2-4	**65.6**	45.5 (1)
ccomp	2-6	**40.3**	15.9 (-3)
advcl	3-2	**34.6**	7.3 (-2)
cop	2-4	**69.0**	33.6 (2)
auxpass	2-4	**94.9**	75.6 (1)

2 Syntactic Properties of Attention Heads

2.1 Data and Model

We utilize the same Transformer architecture as Voita et al. [2019], with three instances of attention (encoder-encoder, decoder-encode, and decoder-decoder), each with 6 layers of attention and 8 heads in each layer. We focus here on encoder-encoder attention. We trained our model using the EN-DE News shared task data from the ACL 2016 First Conference on Machine Translation (WMT 16, Bojar et al. [2016]).

2.2 Method

While the model we trained was similar to Voita et al. [2019], we wanted to expand their syntactic analysis such that we explored more detailed dependencies and also so we could make loose comparisons to the analysis of BERT from Clark et al. [2019].

For this analysis we used `newstest2016`, a subset of the English news text from the 2016 Workshop on Machine Translation task. To get the dependency parses for these sentences we used StanfordNLP (Qi et al. [2018]), a Python package that provides a wrapper for acessing the Java Stanford CoreNLP Server (Manning et al. [2014]). We used CoreNLP to annotate each sentence and extracted, for each word in the sentence, what its syntactic head is and what relation it is in to that syntactic head. Separately, we ran each sentence through our trained model and extracted the attention maps for it.

For each word in a test sentence we find a predicted head word from 6 different baselines and every attention head. Each baseline always predicts a head word with a particular offset; there are 6 baselines spanning the range of offsets [-3, 3]. The prediction of a given attention head for a word at index i of a sentence is defined as the word index to which it assigns maximum weight:

$$\hat{y}_i = \arg\max_j \alpha_{i,j}, \tag{4}$$

where α is the self-attention weight matrix for the sentence. These predictions are then compared to the parsed sentence to calculate accuracy, and statistics are kept for each relation. In other words, each attention head and each baseline maintain an accuracy for each type of syntactic dependency seen in the test set so far. The accuracies get updated as we run through the test set.

For each syntactic dependency, the analysis returns how many times it was seen in the test set, the highest performing attention head, and the highest performing baseline.

2.3 Syntactic Results

The results of our analysis can be seen in Table 1, which shows the syntactic dependency (relation),[1] which head attends to that dependency the most and with what accuracy, and the accuracy of the best baseline. Baselines are the best a head could do if it only attended to words in certain sequential positions to the word in question; the offset of the baseline is shown in parentheses. For instance, an offset of 1 denotes always attending to the next word in the sentence and an offset of -2 always attending to the word two before the word in question. The table displays results for the ten most common dependencies found in our test set plus results for six more dependencies with well-performing syntactic heads. These six dependencies are also fairly frequent in the test set. Bolded accuracies denote heads that performed particularly well with respect to their baselines, where "particularly well" is loosely defined as achieving an accuracy +20% from the baseline.[2]

Not surprisingly, the subject (`nsubj`) and direct object (`dobj`) heads both do well relative to their baselines, which suggests that the model is prioritizing attention to the key players in the sentence. This can be thought of paying attention to the "who did what to whom" of a given sentence. Examples can be found in Figure 1. However, the best `nsubj` head still only achieves 63.7% accuracy, which is relatively low. We suspect that this may be because of the test data used, which is a news corpus. In news writing, a strong subset of the sentences are written in the passive voice, hiding the subject of the sentence. For instance, in the sentence

> *the meeting was also planned to cover the conflict with the Palestinians and the disputed two state solution.*

The meeting is the object of *planned*, but the subject is nonexistent. The same thing happens in the headline

> *Relations between Obama and Netanyahu have been strained for years*.

Relations is the object but there is no clear subject; the "strain-er" of the relations is not explicitly named. It would be interesting to see if this head is still relatively inaccurate when tested on a different set. We note that Clark et al. [2019] and Voita et al. [2019] both also see low accuracies for the subject head on their news corpora; however, Voita et al. [2019] see substantially higher accuracies for their subject head on the OpenSubtitles test set, which does not have the same issue. Thus we reason that this low accuracy is an artifact of the test set.

[1] Definitions of all dependencies used in this analysis can be found at http://universaldependencies.org/docsv1/u/dep/index.html

[2] Voita, et al. (2019) say that a head is syntactic if its accuracy is 10% higher than the baseline; using this threshold would not substantially change our analysis.

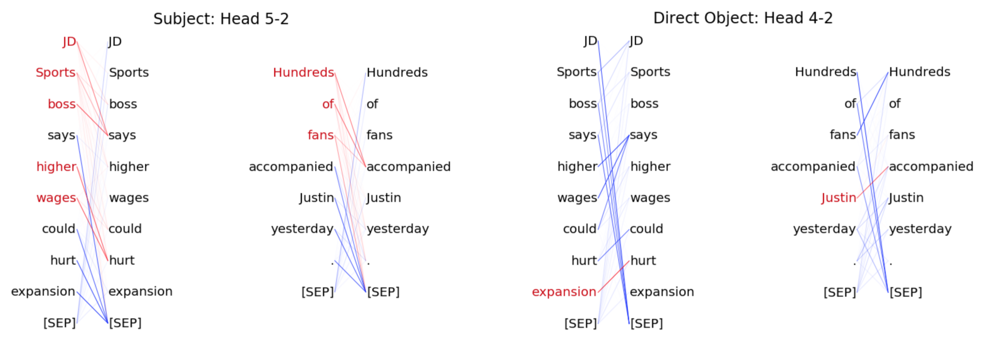

Figure 1: Model attention head that encapsulates subject and object dependencies. In the examples, the darkness of a line corresponds to the strength of the attention weight. All attention from red words is colored red; these colors are there to highlight certain parts of the attention heads' behaviors. Note that these heads are not capturing only the subject or object but also their modifiers. Also note that words not in any of these relations attend to `[SEP]`.

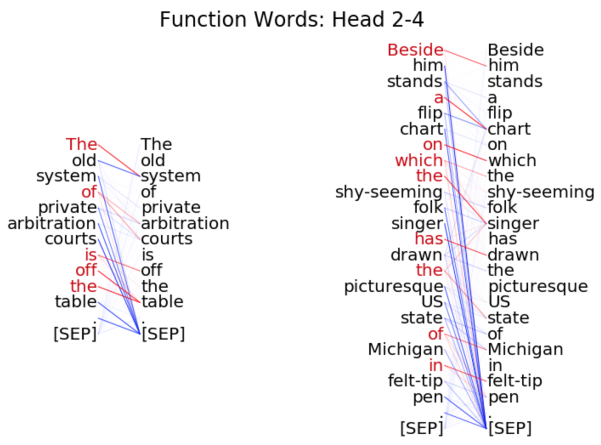

Figure 2: Model attention head that encapsulates most functional dependencies. In the examples, the darkness of a line corresponds to the strength of the attention weight from head 2-4. All attention from red words is colored red; these colors are there to highlight certain parts of the attention heads' behaviors. Here we can see that this single head captures several types of dependencies function and content words. Note that words not in any of these relations attend to [SEP].

There are also, however, several functional dependencies that the model seems to attend to. The `case` head does particularly well relative to its baseline, achieving a 78.5% accuracy above a 36.4% baseline. The `case` dependency ties together any element that is case-marking but a separate word and the word it marks case on; in English this generally refers to possessives (e.g. *Obama's plan*, where *'s* is marking *Obama* with the genitive case) or prepositions. The determiner (`det`) head also does very well, achieving 90.5% accuracy above a 55.0% baseline. Determiners are function words marking nouns, including words like *the* and *a* in English. The fact that the model does very well at picking out these function words suggests that it is attending to more subtle parts of language than the "who did what to whom" attention seen above from the subject and object heads. In particular, it suggests that this translation model encodes features of syntactic hierarchy in the input to use in generation of the translated sentence.

One distinct difference between our results and those of Clark et al. [2019] is that we see one head doing a lot of the work syntactically. Head 2-4 is responsible for increased accuracy of attention with respect to case, determiners, adjectives and adverbs, marks (words that introduce subclauses, such as *that* and *while*), auxiliaries, and copulas. A couple of examples of the attention maps for Head 2-4 can be found in Figure 2. It is not clear why our results end up like this when Clark et al. [2019] find multiple heads mapping these several different functional dependencies. It is possible that it is due to the difference in model sizes; BERT has 12 layers with 12 attention heads each for 144 heads in total, while our model has 6 layers with 8 heads each for only 48 heads in total. Thus it would make sense that our model shares dependencies across heads more readily than BERT.

In summary, many of the dependencies for which we found high-accuracy heads are the same dependencies that Clark et al. [2019] found high-accuracy heads for in BERT, though our analysis also reveals more sharing of heads by syntactic dependencies. For us, one head (2-4) does a lot of the work in attending to function words in the input. Our results also align with those found in Voita et al. [2019] in that `nsubj` and `dobj` heads perform very well relative to the baseline and `amod` and `advmod` perform moderately well relative to the baseline. We extend the analysis performed by Voita et al. [2019] to now show that this kind of model also attends to more nuanced syntactic dependencies such as case, which intuitively makes sense for a translation task.

3 Learning to Prune Attention Heads

The alignment of attention heads with interpretable syntactic functions tells a possible story of their function in the model. However, this alone does not prove that heads that appear more interpretable are enabling these functions, or are needed by the model at all. Thus, it is also important to investigate the value of these heads through ablation studies. Based on the previous work of Voita et al. [2019], we focus on pruning attention heads in encoder layers. Our Transformer model has 6 stacked encoder layers of 8 attention heads each. We can modify the multi-headed attention mechanism as follows. For a single layer,

$$\text{MultiHead}_P(Q, K, V) = \text{Concat}_i(g_i \cdot \text{head}_i)W^O \tag{5}$$

where each head_i is multiplied by a gate variable $g_i \in \{0, 1\}$ controlling whether that head is turned on or off.

Pruning via Ablation Pruning can be done by ablation, or manually setting gate values to zero. Michel et al. [2019] showed that if you iteratively ablate the attention heads of an already trained model (according to their ranking by a gradient-based importance score), the BLEU score is largely preserved—even when some layers have only a single attention head active.

Pruning while Training The pruning of Michel et al. [2019] was performed after training. In contrast, we follow the work of Voita et al. [2019], where pruning is incorporated as part of the training process. That is, the model is trained with a regularization term that penalizes the number of heads that are turned on, encouraging the model to learn to turn off the less important heads. This naturally suggests an L_0 regularization on the gate vector g. Although the L_1 norm also induces sparsity, it would minimize the sum of absolute values of the gates so that the non-zero gates would be pushed to have a low value. Since we want the gate values to be either 0 or 1, the L_0 norm is more suited for our task. However, the L_0 norm is non-differentiable, and would pose problems for optimization.

3.1 Stochastic relaxation of L_0 regularization

In their work, Louizos et al. [2018] propose a stochastic relaxation of L_0 regularization. Rather than using a fixed parameter $g_i \in \{0, 1\}$, they let g_i be a Bernoulli random variable $g_i \sim \text{Bernoulli}(\phi_i)$. Thus g_i still only takes the values 0 and 1, but does so with $P(g_i = 0) = \phi_i$. If we take the expectation of the L_0 regularized loss function with respect to random variable g_i, our goal is then to find parameters ϕ that minimize this expectation:

$$\mathbb{E}[L(\theta, \phi)] = \mathbb{E}[L_{xent}(\theta, \phi)] + \lambda \mathbb{E}[\|\phi\|_0] \tag{6}$$

where θ are the regular parameters, $\mathbb{E}[L_{xent}(\theta, \phi)]$ is the expectation of the regular cross-entropy loss with each head scaled by g_i, and $\mathbb{E}[\|\phi\|_0]$ is the expectation of the L_0 norm of g. It can be easily shown that $\mathbb{E}[\|\phi\|_0] = \sum(1 - \phi_i)$.

3.2 Continuous relaxation of discrete distribution

The presence of the discrete g_i in the body of the loss function still poses problems for gradient-based optimization. The work of Jang et al. [2017] and Maddison et al. [2017] introduces a general method for the continuous relaxation of discrete random variables. As it turns out, any discrete distribution can be approximated by a continuous distribution called the Concrete (or Gumbel-Softmax); this distribution uses the same parameters with an additional "temperature" parameter τ, where the approximation becomes perfect as $\tau \to 0$.

In our case, each g_i is approximated with a binary version of the Concrete with parameter ϕ_i and common temperature τ. With some massaging by Louizos et al., g_i can now take any value in the interval $[0, 1]$. As long as τ is small, the probability mass will be highly concentrated on 0 and 1, with the particular weighting between the two determined by the parameter ϕ_i.

The sum of the probabilities of heads being non-zero, L_C, is thus this stochastic relaxation of the L_0 norm. We first train a model to convergence with objective function L_{xent}. Then, we add gates to the converged model and train according to

$$L = L_{xent} + \lambda L_C. \tag{7}$$

3.3 Convergence of ϕ_i parameters

Recall that ϕ_i indicates the probability that g_i is 0. Intuitively we would want these parameters to be either 0 or 1 at convergence. This would mean that each gate is "almost always" on or off. This isn't

Table 2: Converged gate values

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
Layer 1	0	0	0	0	0	0	1	1
Layer 2	1	0	0	1	0	0	0	0
Layer 3	0	0	0	0	1	0	0	0
Layer 4	1	1	0	0	0	0	0	0
Layer 5	0	1	0	0	0	1	0	1
Layer 6	0	0	0	1	0	0	0	0

enforced by the L_0 regularization, but should be encouraged by the main body of the loss function, assuming that some heads are simply more important than others for the task at hand. Voita et al. [2019] indeed found that gate parameters ϕ_i do in fact converge to either 0 or 1, and hence the model does learn which heads are, broadly, more important than others.

3.4 Pruning results

With our WMT EN-DE data set, we set the regularization coefficient λ and temperature τ to 0.1 and 0.33 respectively. We trained a baseline model with seq2seq, and then trained from the baseline according to equation (7).

The gate values converged to approximately 0 or 1, as shown in Table 2. The gate values of 1 correspond to important heads that should be kept on, while gate values of 0 correspond to non-important heads that can be turned off. The layers are sparse: some have only one activated head. This result agrees with the findings of Michel et al. [2019] that Transformers can maintain performance despite aggressive pruning. Additionally, we see that the activated heads include those found to be important in the syntactic analysis (namely heads 2-4, 4-2, and 5-2). The final BLEU score after training while learning gate parameters was 27.6 (Table 3, row 2).

3.5 Fine-tuning with fixed heads

After training with the objective function (7) until the gate values converged, Voita et al. [2019] train a model from scratch with only $g_i = 1$ heads active. They then show the model pruned and trained from scratch performs significantly worse (lower BLEU score) than a model that has learned to prune concomitantly with its translation task. We reason that training a pruned model from scratch is essentially just training a smaller model. Thus, we extend this result by training a baseline model and pruning the baseline, where we only select the $g_i = 1$ heads. Training of the pruned baseline shows that although the BLEU score degrades slightly, it still remains within 5% of the original score (27.5, converging to ≈ 26.2, Table 3, rows 3 and 4) although only 11/48 of the original heads remain (Table 2).

3.6 Inverse pruning

Prior results show that a significant portion of the unpruned BLEU score can be retained if pruning is learned and applied during fine-tuning or testing. We investigate the converse of this result: if only the important heads are removed, and the non-important heads left intact, does the model's performance degrade as expected? The results show that after inverting the gates, the BLEU score actually degrades less than for fixing only important heads (28.3, Table 3). While this inverted model has many more heads, it is still surprising that it performs relatively well without the "important" heads. This suggests that, while functionally interpretable heads are indeed important, those functions may be more distributed than they initially appeared in the studies of Voita et al. [2019].

4 Future work

Syntactic: Moving forward, there are several interesting directions to take this work. Firstly, it is important to extend these experiments to models of languages very different from English and German to determine whether or not the dependencies that the model picks out are universal cross-linguistically. It would also be interesting to see this analysis for models trained on more kinds of tasks

Table 3: BLEU scores of pruning experiment models

Experiment	**BLEU**
Baseline (converged)	29.3
Pruned (converged)	27.6
Non-inverted (initial)	27.5
Non-inverted (16000 batches)	26.2
Inverted (initial)	28.3
Inverted (4000 batches)	28.09

and different data sets to see if the results are task or dataset specific. As mentioned previously, we would also like to check if getting a head like 2-4, which carries most of the functional dependencies, is a byproduct of the model parameters by running the same analysis on a model trained in the same way but with parameters matching BERT (12 layers with 12 attention heads each).

Pruning: Following the results of our inverse pruning experiment, it would be interesting to look more closely into why the non-important heads perform so well (with the important heads removed). To better characterize the effects of the regularization coefficient and temperature on our dataset, it would also be informative to vary those parameters and compare to results on other datasets—for example, the work of Voita et al. [2019]. For a given regularization coefficient, comparing the number of retained heads and their functions from two different languages may be an insightful comparison in the modeling of those languages. Another interesting avenue may be swapping heads with the same head functions between two different trained models, to see if their activations are still predictive of annotated syntactic features in their new setting. This may help illuminate whether functionality is modularized within attention heads or more distributed in the model.

5 Conclusion

We analyze the syntactic properties of Transformer attention heads and find that our baseline model has several heads that attend to syntactic dependencies, extending the analysis in Voita et al. [2019] to include results from heads attending to functional dependencies. We see that many of the syntactic heads align with the syntactic heads found in BERT by Clark et al. [2019], however, we find a single head to be attending to many of the different syntactic dependencies. We posit this is likely because our model is much smaller than BERT.

We also extended the ablation studies of Voita et al. [2019] by training a baseline model with all the heads on, learning which heads were important, and then continuing training with only the important heads. We find that the BLEU score dropped after turning off non-important heads, but only slightly. This verifies the idea that certain heads are less important to the translation task and can be turned off without affecting the BLEU score significantly. Moreover, the heads found to be syntactically important are retained by the pruned model, verifying that they are indeed strongly utilized by the model. In addition, we tried inverting the learned gate values to turn on the non-important heads and turn off the important heads. Interestingly, this increased the BLEU score slightly. This suggests that while the group of all non-important heads are less necessary to the model's performance than the important heads, they are still effective at the translation task.

Acknowledgments

We would like to thank Voita et al. and Clark et al. for their insightful works. Our project extensively uses the code from those works, and is largely based upon the ideas the authors originally developed. We especially thank Lena Voita for graciously responding to our inquiries regarding their methods and implementations. We also acknowledge Marine Carpuat for helpful suggestions and direction.

References

O. Bojar, Y. Graham, A. Kamran, and M. Stanojević. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, 2016.

K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What Does BERT Look At? An Analysis of BERT's Attention. *arXiv*, 2019. URL `https://arxiv.org/pdf/1906.04341.pdf`.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*, 2019.

Y. Goldberg. Assessing BERT's Syntactic Abilities. *arXiv preprint arXiv:1901.05287*, 2019.

E. Jang, S. Gu, and B. Poole. Categorical Reparameterization with Gumbel-Softmax. *ICLR*, 2017.

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

C. Louizos, M. Welling, and K. P. Diederik. Learning Sparse Neural Networks Through L_0 Representation. *ICLR*, 2018.

C. J. Maddison, A. Mnih, and Y. W. Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *ICLR*, 2017.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010`.

P. Michel, O. Levy, and G. Neubig. Are Sixteen Heads Really Better than One? *NeurIPS*, 2019.

P. Qi, T. Dozat, Y. Zhang, and C. D. Manning. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL `https://nlp.stanford.edu/pubs/qi2018universal.pdf`.

A. Raganato, J. Tiedemann, et al. 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*. Association for Computational Linguistics, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Å. Kaiser, and I. Polosukhin. Attention Is All You Need. *NIPS*, 2017.

E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Tito. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *ACL*, 2019.