

Context-Sensitive Visualization of Deep Learning Natural Language Processing Models

Andrew Dunn
Central Washington University
Ellensburg, WA, USA
Email: andrew.dunn@cwu.edu

Diana Inkpen
University of Ottawa
Ottawa, ON, Canada
Email: diana.inkpen@uOttawa.ca

Răzvan Andonie
Central Washington University
Ellensburg, WA, USA
and
Transilvania University, Braşov, Romania
Email: razvan.andonie@cwu.edu

Abstract—The introduction of Transformer neural networks has changed the landscape of Natural Language Processing (NLP) during the last years. So far, none of the visualization systems has yet managed to examine all the facets of the Transformers. This gave us the motivation of the current work. We propose a novel NLP Transformer context-sensitive visualization method that leverages existing NLP tools to find the most significant groups of tokens (words) that have the greatest effect on the output, thus preserving some context from the original text. The original contribution is a context-aware visualization method of the most influential word combinations with respect to a classifier. This context-sensitive approach leads to heatmaps that include more of the relevant information pertaining to the classification, as well as more accurately highlighting the most important words from the input text. The proposed method uses a dependency parser, a BERT model, and the leave- n -out technique. Experimental results suggest that improved visualizations increase the understanding of the model, and help design models that perform closer to the human level of understanding for these problems.

Index Terms—Natural Language Processing, Transformers, BERT, Visualization of Neural Networks

I. INTRODUCTION

The latest deep learning models are extremely complex, and can contain hundreds of millions of different trainable parameters. This high level of complexity makes it very difficult for humans to understand their inner workings. Visualization methods give us a window into these black box models, and can be beneficial for a number of reasons. For example, visualizations may provide a mechanism for debugging complex models by showing which parts of the input data the model is focusing on. This can be especially useful in classification models, because it may show that a model is focusing on parts of the data that are irrelevant to the predicted class.

Saliency maps are a popular tool for gaining insight into deep learning. In this case, saliency maps are typically depicted as heatmaps of neural layers, where "hotness" corresponds to regions that have a big impact on the model's final decision. Grad-CAM is a state-of-the art technique for saliency map visualization [1] using the gradient information obtained from backpropagating the error signal from the loss function with respect to a specific feature map, at any layer of the network.

While the idea of using visualization methods on image processing models makes intuitive sense, in recent years there

has been a growing interest in visualizing NLP models. However, with NLP models visualization is not as straight forward due to working with text. Current solutions to visualizing NLP models involve heatmaps similar to Grad-CAM, and visualizing connections between tokens in models that utilize attention mechanisms. These solutions typically look at the importance of individual tokens on the model output. Before being passed to an NLP model, the text must be broken up into a series of tokens that are then transformed into numerical vectors. Each token typically represents a single word or character in the input text. The goal of these NLP visualization methods is to highlight the most significant parts of the text that have the greatest impact on the model output. While effective, current methods look at each separate token independently, thus lacking context from the entire text. We believe the combination and order in which the words appear is significant and that they influence an NLP model's interpretation of the text.

Typically, for tasks such as machine translation, the output is in textual form, and an encoder-decoder mechanism is used. The weakness of this approach is that the encoder produces a finite-length vector, without regards to whether or not any of the inputs are more important (or less important). A solution to this problem is the *attention* mechanism. Several variants of this mechanism have been introduced: convolutional, intra-temporal, gated, and self-attention [2]. Self-attention has recently become very popular in the state-of-the art model called *Transformer*, and provides attention to words in the same sentence.

Visualization is considered a preliminary step for interpreting or explaining a trained machine learning model. The terms interpretability and explainability are often used interchangeably. However, it is generally accepted that there is a difference between them. Interpretability in a machine learning model means that the model behaves in a consistent and predictable manner, so that changes to model parameters result in expected changes in the model's output. Explainability refers to the process of providing *post hoc* explanations for the behavior of black box models [3]. In this paper, we are primarily concerned with the explainability of Deep Neural Network (DNN) models.

One common method for measuring the impact of individual

elements of an input data on a model’s decision making process is called leave-one-out, and can be used for visualization. The leave-one-out (LOO) process is a feature erasure technique that systematically identifies and removes a single feature from the input data to measure its affect. In an NLP model, a feature is the vector representing a single token in the input text, typically a word or character, although character n-grams and subwords may also be selected as tokens. The modified data is then passed through the model and the difference in output strength from the full input is recorded. This process is repeated for every feature. The token influence is calculated by subtracting the output strength of the modified inputs from the output strength of the original unmodified input. The output strength of the removed features are then ranked in order of their influence score, and all tokens that had a positive difference in output value when removed are considered as the set of most influential features on the model’s output.

Other visualization solutions, like LOO and saliency maps, look at each input token individually. Since new language models such as BERT are capable of understanding complex phrases and relationships between words, we believe new visualization methods that preserve some context from the original text will improve the explainability of these models.

Our core contribution is a novel NLP Transformer context-sensitive visualization method that leverages existing NLP tools to find the most significant groups of tokens (words) that have the greatest effect on the output, thus preserving some context from the original text.

First, we use a sentence-level dependency parser to highlight promising word groups. The dependency parser creates a tree of relationships between the words in the sentence. Next, we apply a leave- n -out (LNO) process: we systematically remove adjacent and non-adjacent tuples of n tokens from the input text, producing several new texts with those tokens missing. The resulting texts are then passed to a BERT-based model. The classification output is compared with that of the full text, and the difference in the activation strength is recorded. The modified texts that produce the largest difference in the target classification output neuron are selected, and the combination of removed words are then considered to be the most influential on the model’s output. In the visualization stage, these most influential word combinations are highlighted by a heatmap.

The rest of the paper is structured as follows. Section II introduces basic concepts and describes related work. Section III describes our method. Experimental results are detailed and discussed in Section IV. Finally, Section V presents conclusions and open problems.

II. RELATED WORK: VISUALIZATION OF TRANSFORMER-BASED DEEP NETWORKS

One of the first papers introducing visualizations of neural NLP models is [4]. Four strategies for visualizing neural models for NLP were described, all inspired from similar work in computer vision. However, NLP visualization is more complex

than visualization in image recognition models. Recently, more specialized techniques were developed for NLP visualization, the most popular one being Transformers-based deep networks [5].

The introduction of Transformer neural networks has changed the landscape of NLP during the last years. The Transformer combines an encoder, decoder, and a self-attention mechanism to create a very effective model that is able to infer connections between inputs in sequential data, similar to recurrent neural networks like the popular long short-term memory (LSTM) models [6]. This allows the Transformer model to extract complex features from the input data, and has proven to be very effective in solving NLP problems.

BERT [7] is a state-of-the-art NLP model developed by Google in 2018 that has seen many successes in common NLP tasks such as classification, sentence prediction, named entity recognition, and more. BERT stands for Bidirectional Encoder Representations from Transformers, and is built using a stack of Transformer encoders. BERT is a very large and deep network containing 110 million trainable parameters (BERT-base) or 345 million parameters (BERT-large), which makes training BERT from scratch very time consuming. Due to the large number of parameters, explaining the individual components or behavior of the model is very difficult.

The success of BERT has lead to researchers to create several similar language models based on Transformers, such as RoBERTa [8], DistilBERT [9], and XLNet [10] that are aimed at either improving the accuracy of the model, or reducing the training time and the complexity of the model.

GPT-2 is another very large language model developed by OpenAI in 2019 [11]. Like BERT, GPT-2 is based on Transformers and was designed to come in multiple sizes. GPT-2 Large contains 762 million trainable parameters, making it considerably larger than BERT-large. Where BERT is built using a stack of Transformer encoders, GPT-2 is made up from a stack of Transformer decoders. In addition, GPT-2 uses masked self-attention, which limits attention weights between the current and previous tokens in the sequence. In contrast, BERT uses unmasked self-attention, which allows attention weights to be generated between all tokens in the text, regardless of their position. In 2020, OpenAI released GPT-3 [12], a massive language model composed of 175 billion trainable parameters. At the time of completion, GPT-3 was the largest DNN language model ever created. In certain applications, GPT-3 has been able to generate text that indistinguishable from human written text. As of the writing of this paper, GPT-3 is not publicly available for download.

Explaining the information processing flow and results in a Transformer is difficult because of its complexity. A convenient and very actual approach is visualization. Visualization enables some degree of explainability. A first survey on visualization techniques for Transformers is [5].

Visualizing complex models like those built with Transformers, can sometimes be accomplished by using model-agnostic tools specifically built for benchmarking or hyperparameter

scoring, such as *Weights and Biases*¹ - which provides the largest sets of visualization and customization capabilities. Perhaps more interesting are the visualizations specifically built around Transformers, either for the purpose of explaining the models (like ExBERT [13]), or for explaining certain model-specific attributes (like embeddings or attention maps [14]). Details are provided in [5], including a list of recent articles focused on explaining Transformer topics through visualizations.

In addition to Transformer visualizations, there are several specific BERT visualization approaches [5] focused on the role of embeddings and relational dependencies within the Transformer learning phase [15], the role of attention during pre-training or training [16], [14], and the importance of various linguistic phenomena encoded in its language model [17].

A general goal would be to visualize the entire lifecycle of a Transformer models. However, none of the current visualization systems (considered in [5]) has yet managed to examine all the facets of the Transformers. This is due to the fact that this area is relatively new and there is no consensus on what needs to be visualized. The visualization of NLP neural models is still rather young, and we believe that our proposed method brings a new and interesting insight.

Recently, Han *et al.* [18] mapped saliency over NLP inputs to explain predictions via influential training examples using a BERT-based model. They found that in a lexicon-driven sentiment analysis task, saliency maps and influence functions are largely consistent with each other.

However, Han *et al.*'s [18] results were not consistent for the more complex task of Natural Language Inference, a classification problem that concerns the relationship between a premise sentence and a hypothesis sentence. An explanation could be that saliency maps focus on the input, and may neglect to explain how the model makes decisions. Some saliency methods are incapable of supporting tasks that require explanations that are faithful to the model or the data generating process [19].

Very recently, Yann LeCun's team from Facebook AI Research, UC Berkeley and New York University applied dictionary learning techniques to provide detailed visualizations of transformer representations and insights into the semantic structures [20]. They have created an interactive website² where users can gain additional insights into transformer models by visualizing their latent space. They used a pretrained BERT model for their evaluation experiments. Transformer factors were divided into three different levels: low level (word-level), mid level (sentence-level) and high level (long range patterns). To the extend of our knowledge, the closest existent result to our contribution is [20]. Our work was performed before the publication of this result.

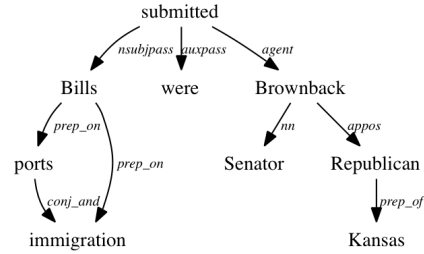


Fig. 1. Dependency parser example output [21], [22]

III. PROPOSED METHOD: CONTEXT-SENSITIVE VISUALIZATION

Based on LOO, the leave- n -out (LNO) process systematically removes adjacent and non-adjacent tuples of n tokens from the input text, producing several new texts with those tokens missing. Where LOO removes a single token at a time, LNO removes multiple tokens at each step. The resulting texts are then passed to the model and their classification output is compared with that of the full text, and the difference in the activation strength is recorded. The modified texts that produce the largest difference in the target classification output neuron are selected, and the combination of removed words are then considered to be the most influential on the model's output. In our experiments, we elected to use word pairs, and thus use tuples of size two. Tuples of more than two tokens could be also used, and may provide different results. However, the computational overhead may increase in this case exponentially and the benefits are questionable.

Our approach looks at combinations of words and phrases to discover their effect on the model's output, producing a visualization that is more context sensitive to the original text. Our method leverages existing NLP tools for dependency parsing combined with a model-agnostic LNO token pre-processing approach to identify and highlight contextual groups of tokens that have the largest perceived effect on the model's classification output. The model itself can be any binary or multi-label NLP neural network classification model as long as the output layer is a fully-connected layer with the same number of neurons as classes.

Since the LNO processing step is computationally expensive and the number of possible word combinations explodes exponentially, we use a sentence-level dependency parser from the spaCy NLP library³ to highlight promising word groups. The dependency parser creates a tree of relationships between the words in the sentence. By only selecting groups of words that are connected via a branch in the dependency tree, we are able to greatly reduce the search space for the algorithm. Figure 1 shows a visual example of a dependency tree generated by the Stanford Dependencies library, created at the Stanford

¹<https://www.wandb.com/>

²<https://transformervis.github.io/transformervis/>

³<https://spacy.io/>

University⁴. The original sentence that was given to the dependency parser as follows: *Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas.*

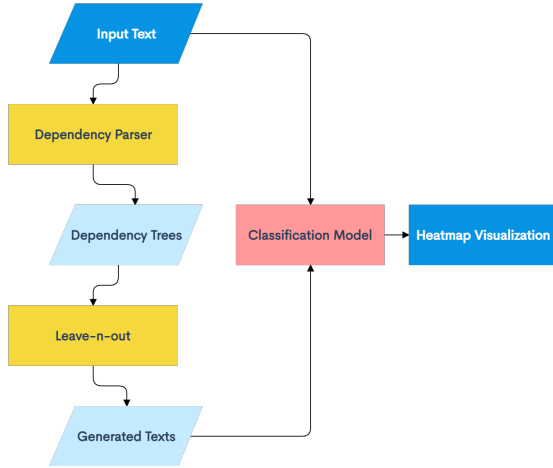


Fig. 2. LNO process flowchart

Using the previously described processes, the steps required to visualize a text using our method are as follows. First, an input text is passed through the classification model and the class prediction and output strength is recorded as the baseline. Next, each sentence in the text is passed through a dependency parser which produces a tree structure highlighting dependency relations between the various words in each sentence. Using these dependency trees, new texts are generated with word pairs removed. Word pairs are limited to those that are connected via a branch in the dependency tree. Each generated text has only one word pair removed at a time, and words are unique to their position in the sentence. After all new texts are generated, we proceed to the LNO process. All generated texts are passed through the model and their class prediction and output strengths are recorded. Once all texts have been processed, the output strength of the predicted class neuron for each generated text is then compared to output strength of the original baseline text. The heatmap is then generated by first sorting all word pairs by their output strength difference to that of the original text. Word pairs that did not contribute to the class prediction are culled. If an individual word appears in more than one pair, the maximum output strength for that word out of all pairs is selected. Finally, the output strengths for all remaining words is normalized using a linear scale. Figure 2 contains a simplified flowchart detailing the order of operations for our method.

IV. EXPERIMENTS

For our experiments, we used two datasets. First, the IMDB Large Movie Review sentiment analysis dataset⁵ from Stanford University, which contains 25,000 training and 25,000 testing reviews labelled with two categories: positive reviews

⁴<https://nlp.stanford.edu/software/nndep.html>

⁵<https://ai.stanford.edu/amaas/data/sentiment/>

and negative reviews. We used 20% of the training set as a separate validation dataset. As mentioned, the test set is separate from the training and validation sets.

Second, we used the Amazon Reviews sentiment analysis dataset from Xiang Zhang⁶, which was randomly sampled to contain 600,000 training and 260,000 testing reviews evenly split between two classes: One star and five star reviews, representing negative and positive reviews respectively. Again, 20% of the training set was used as a separate validation dataset.

All code is written in Python 3.8, and utilizes the Tensorflow version of the Transformers library⁷. A dropout layer and a fully connected layer are appended to the pretrained BERT-base model to create our sentiment analysis classification model. Text tokenization and dependency parsing is executed using the spaCy NLP library⁸. The visualization images are generated using the Matplotlib library⁹.

In the following, we will present some experimental results and heatmap visualizations for both the LOO and LNO methods for two models trained on the aforementioned datasets. Both models are based on the same architecture.

A. IMDB Large Movie Reviews Dataset Model

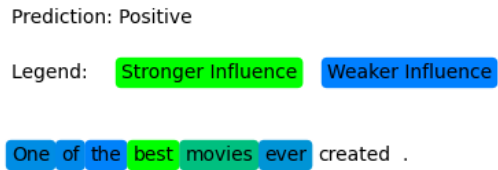


Fig. 3. Example LOO heatmap output - positive movie review

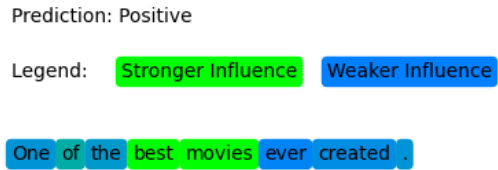


Fig. 4. Example LNO heatmap output - positive movie review

Figures 3 and 4 show a short example text that represents a positively-classified movie review. In Figure 3, the LOO process identified the word "best" as the most influential to the classification. We can see in Figure 4 that more of the sentence context is highlighted, with the word pair "best movies" most prominently highlighted, showing that LNO picked out more of the context.

⁶<https://www.kaggle.com/bittlingmayer/amazonreviews>

⁷<https://huggingface.co/transformers/>

⁸<https://spacy.io/>

⁹<https://matplotlib.org/>

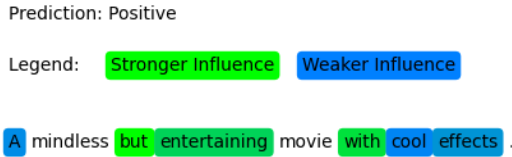


Fig. 5. Example LOO heatmap output - mixed positive movie review

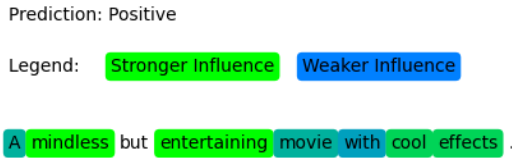


Fig. 6. Example LNO heatmap output - mixed positive movie review

Figures 5 and 6 show a mixed positively-classified review. Again, Figure 5 shows the resulting visualization from the LOO process. Note that the word "mindless" is not highlighted, but the word "but" is. Interestingly, Figure 6 shows that heatmap result from LNO has reversed for these two words, with "mindless" highlighted with a stronger influence on the classification and "but" being not highlighted. Also note that the LOO result gives the word "with" a stronger influence than the words "cool effects", which is contrary to what we might expect. In contrast, the LNO heatmap gives more importance to the "cool effects", which is more inline with our expectations.

B. Amazon Reviews Dataset Model

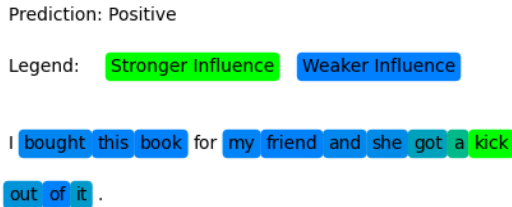


Fig. 7. Example LOO heatmap output - positive product review

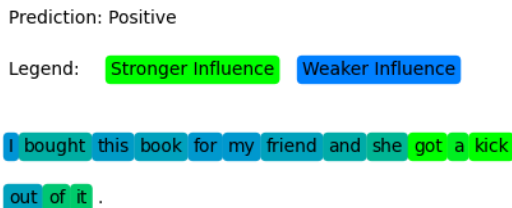


Fig. 8. Example LNO heatmap output - positive product review

Figure 7 contains a positive Amazon product review visualized using the LOO process, while figure 8 shows the same text visualized using the LNO process. Here we observe that the phrase "got a kick" shows a stronger influence with the LNO process over LOO, highlighting more of the context around the word "kick".

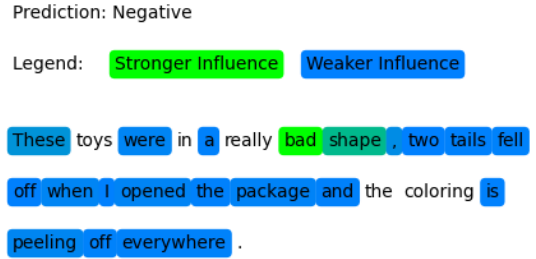


Fig. 9. Example LOO heatmap output - negative product review

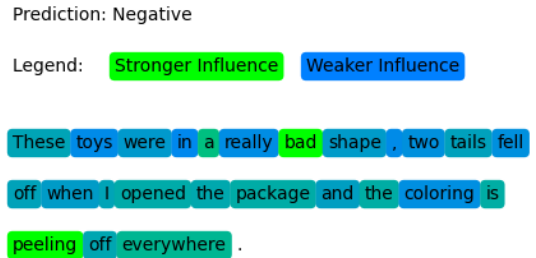


Fig. 10. Example LNO heatmap output - negative product review

Figures 9 and 10 display a negative product review. In Figure 9, the LOO process visualization highlighted the words "bad shape" most strongly, while the LNO process highlighted more of the text. Notably, the LNO process selected the word "peeling" as strongly influencing the classification, which was shown as having a weaker influence in the LOO process. Finally, we observe that in general the LNO process utilizes more of the heatmap color range than the LOO process.

V. CONCLUSION

Given that NLP tasks require some special considerations due to the nature of how languages are represented and understood, visualization tools that are tailored specifically for NLP models may be preferred over more traditional methods, such as saliency maps based on gradients. The knowledge contained within a text goes beyond the meaning of the its individual words. The relationships between words and their positions within the sentence are necessary for a full understanding of the text. Thus, we believe a visualization method that retains some of the contextual meaning between the different words in the input text will produce improved results that are more inline with our expectations. Improved visualizations increase our understanding of the model, and help us design better models that perform closer to the human level of understanding

for these problems. With the method presented in this paper, we have attempted to create a context-sensitive visualization method for deep NLP classification models. This more context-sensitive approach leads to heatmaps that include more of the relevant information pertaining to the classification, as well as more accurately highlighting the most important words from the text.

The terms attention and explanation are used frequently in the context of visualization of deep networks, for instance when using saliency (or heatmaps) as visualization tools. While attention modules may yield improved performance on NLP tasks, their ability to provide meaningful explanations for model predictions is questionable. However, according to [23], attention weights over non-recurrent encoders exhibit relatively strong correlations, on average, with LOO feature importance scores and gradient-based measures. Since we use LNO and gradient-based techniques, we may expect to have some correlation between attention weights and feature importance scores for the top features. We leave this aspect as a direction of investigation in future work.

We also leave for future work an experimental comparison with the recently proposed method in [20].

REFERENCES

- [1] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, p. 336–359, Oct 2019. [Online]. Available: <http://dx.doi.org/10.1007/s11263-019-01228-7>
- [2] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.
- [3] R. Marcinkevičs and J. E. Vogt, "Interpretability and explainability: A machine learning zoo mini-tour," 2020.
- [4] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in NLP," *arXiv preprint arXiv:1506.01066*, 2015.
- [5] A. M. P. Braşoveanu and R. Andonie, "Visualizing transformers for NLP: A brief survey," in *2020 24th International Conference Information Visualisation (IV)*, 2020, pp. 270–279.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "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*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008. [Online]. Available: <http://papers.nips.cc/paper/7181-attention-is-all-you-need>
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: <https://doi.org/10.18653/v1/n19-1423>
- [8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: <http://arxiv.org/abs/1907.11692>
- [9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: <http://arxiv.org/abs/1910.01108>
- [10] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 5754–5764. [Online]. Available: <http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding>
- [11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: <https://github.com/openai/gpt-2>
- [12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: <https://github.com/openai/gpt-3>
- [13] B. Hoover, H. Strobelt, and S. Gehrmann, "exbert: A visual analysis tool to explore learned representations in transformers models," *CoRR*, vol. abs/1910.05276, 2019. [Online]. Available: <http://arxiv.org/abs/1910.05276>
- [14] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, M. R. Costa-jussà and E. Alfonseca, Eds. Association for Computational Linguistics, 2019, pp. 37–42. [Online]. Available: <https://doi.org/10.18653/v1/p19-3007>
- [15] E. Reif, A. Yuan, M. Wattenberg, F. B. Viégas, A. Coenen, A. Pearce, and B. Kim, "Visualizing and measuring the geometry of BERT," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8592–8600. [Online]. Available: <http://papers.nips.cc/paper/9065-visualizing-and-measuring-the-geometry-of-bert>
- [16] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: pre-training of generic visual-linguistic representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: <https://openreview.net/forum?id=SygXPaEYvH>
- [17] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? an analysis of bert's attention," *CoRR*, vol. abs/1906.04341, 2019. [Online]. Available: <http://arxiv.org/abs/1906.04341>
- [18] X. Han, B. C. Wallace, and Y. Tsvetkov, "Explaining black box predictions and unveiling data artifacts through influence functions," 2020.
- [19] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems*, 2018, pp. 9505–9515.
- [20] Z. Yun, Y. Chen, B. A. Olshausen, and Y. LeCun, "Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors," *arXiv preprint arXiv:2103.15949*, 2021.
- [21] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 740–750. [Online]. Available: <https://www.aclweb.org/anthology/D14-1082>
- [22] <https://nlp.stanford.edu/software/stanford-dependencies.shtml>. [Online]. Available: <https://nlp.stanford.edu/software/stanford-dependencies.shtml>
- [23] S. Jain and B. C. Wallace, "Attention is not explanation," *CoRR*, vol. abs/1902.10186, 2019. [Online]. Available: <http://arxiv.org/abs/1902.10186>
- [24] H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 2019. [Online]. Available: <http://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019>