Evaluating the Explanatory Capabilities of Transformers

Hannes Lindbäck
Uppsala University, Sweden
Department of Linguistics and Philology
hannes.lindback.7493@student.uu.se

Abstract

Getting large language models to provide explanations for their prediction is a key task in the field of NLP as large models essentially function as black boxes, making their outputs difficult to interpret. One way of solving this is to provide a model with examples of such explanations, and let the model learn to provide explanations on their own. Explanations such as these are provided in the e-SNLI dataset, an extension of the Stanford Natural Language Inference dataset, where the task is to classify sentence relations. By training the model on generating explanations as it learns to classify the sentences in the e-SNLI dataset, it is possible to get a model that can explain its own label predictions. In this study we investigate the quality of explanations possible to achieve by training a transformer model on the e-SNLI dataset and get mixed results.

1 Introduction

As modern computational models for natural language processing have become better and better and more capable of accurately understanding and modelling human language, they have too grown in size. The huge size of modern models means that it has become difficult to understand the reasoning behind the model's predictions. The models are so called black boxes. They can provide an output based on the input, but it is not possible to see what is going on inside the box. It is therefore a key task in NLP to get large language models to provide explanations that make it possible to understand how the model came to make a certain prediction. For instance, in a natural language inference task (NLI) a model reads a premise and a hypothesis, and predicts a label of the type of relation between the premise and the hypothesis, but due to the black box-nature of NLI-models, the reason to why it predicted a certain label for a certain relation is unknown. This issue can be circumvented by getting the model to generate explanations to the relation of the premise and hypothesis while also predicting a label.

This extension of the NLI-task is provided in the e-SNLI dataset[1], where apart from a premise and an hypothesis, there is also a natural language explanation of the premise-hypothesis relationship. Camburu et al. (2018) trained an LSTM model on this dataset and got a model capable of providing natural language explanations together with the predicted label. By letting the model explain its own predictions the model becomes much more interpretable than would otherwise have been the case.

The network architecture used in Camburu et al. (2018) is that of an LSTM. While effective, this architecture is no longer the state-of-the-art for natural language processing, this is instead the transformer architecture (Vaswani et al., 2017). Transformers are very capable at processing natural language, as is evident by the leaderboard of the GLUE benchmark (Wang et al., 2018), where models using transformer architecture are all in the top. Extending the experiments of Camburu et al. (2018) to the transformer architecture is therefore of interest in order to keep the research up to date. But of even greater interest is the transformer's contextual understanding capabilities, given by its self-attention mechanism, which have been demonstrated to make transformers excel at natural language understanding and text generation tasks (Devlin et al., 2019; Brown et al., 2020). As the goal within explainable AI is to achieve more interpretable and explanatory models, applying transformer networks to the e-SNLI task (a task of both NLU and text generation), should provide more interpretable models.

The purpose of this study is therefore to eval-

[1] https://github.com/OanaMariaCamburu/e-SNLI

uate how capable transformer models are at generating explanations to a model's label prediction. As baseline for this experiment we use two LSTM models - one LSTM model with an additive attention module (Bahdanau et al., 2014), and one LSTM model without attention. Since attention mechanisms have been demonstrated to make a model better understand context, we expect the models with attention to perform better than the model without, and that the model with the largest attention mechanism, the transformer to perform the best.

2 Related work

Numerous studies have previously examined and experimented with ways to procure natural language processing models that are more interpretable and capable of giving explanations to their predictions. These previous attempts can largely be divided into categories of *natural language explanations* and *non-natural language explanations*. A non-natural language explanation uses instead of natural language typically scores of some kind to provide some interpretability to the output. Natural language explanations instead learns to output explanations in natural language along side with their learned predictions. Examples of both are given below.

2.1 Non-natural language explanations

A common way of attempting to explain machine learning models' predictions without using natural language is through the means of attention. As attention essentially provides a weighted representation of the input which is then used to generate the output, it has been hypothesized that it is possible to get explanations for an output by looking at the attention weights. Xie et al. (2017) used attention to model relations between concepts in a knowledge base. By using the weights of attention vectors they could also visualize these relations, making the model more interpretable than would otherwise have been the case. In an aspect-level sentiment classification task, Wang et al. (2016) similarly used attention as a way of providing a more interpretable model. By using heatmaps of the attention weights for the input sequences they could show which tokens had mattered more when predicting the label for the input sequence.

Using attention weights for explaining model predictions has been questioned however. Serrano and Smith (2019) found that there appears to be a low correlation between changes in attention weights and model decision flips and that it is difficult to find the input representations most important to the models' predictions. They conclude that attention is unreliable to use for providing explainability and interpretability to a model. Jain and Wallace (2019) examined the correlation between attention weights and other measures of feature importance; in a one-hot encoded vector, if an element is measured to carry high feature importance, then the attention weights for that element should also be high. They found that this was not the case. The correlation between attention weights and other measurements of feature importance was very low. Similarly, there seemed to be little correlation in changes in attention weights and model outputs. Adversarial attention weights (weights that differ as much as possible from the original), did not lead to similar adversarial model outputs.

Yin and Neubig (2022) provide explanations by giving scores not only to the tokens most responsible for the prediction, but also for the tokens most responsible for one prediction instead of for another prediction; the explanations are contrastive. The scores are calculated from different types of saliency scores, where the contrastive scores are achieved by calculating how much an input token influences the model to not predict a certain output.

2.2 Natural language explanations

One type of natural language explanation is to let a predicted target sequence have source sequence word order and then use a sequence of operations to realign the target sequence to the correct word order (Stahlberg et al., 2018). In a machine translation task, a source sequence is given as input to an encoder-decoder model which then outputs a target sequence, ideally with correct word-for-word translations and with the words in a grammatical order. But as many source-target language pairs do not have exact word-for-word translations, the interpretability of the model decreases since it is not clear which token in the input sequence that produced a certain word or grammatical feature in the output sequence. By letting the output sequence consist of tokens in the same order as the input sequence interspersed with operations that would realign output to have the pre-

dicted word order, it becomes easier to see the relationship between the source and target tokens.

In a knowledge-base question-answering task, a model takes a query and outputs an answer based on its knowledge base. A way to let the model explain its answer to the query is to let the output also contain an explanation to how the model interpreted the query. Abujabal et al. (2017) shows, along with the answer, to which knowledge base entities and predicates the query were mapped. This gives the user a look at the internal derivation process of the model explained in natural language.

Another option is to let the model train on datasets where human-provided explanations are given to the each data entry. This enables a model to learn to predict explanations along with the label prediction the model makes (Rajani et al., 2019). The e-SNLI dataset (Camburu et al., 2018) explained in detail below is also an example of this explanation method.

3 Experimental setup

3.1 Data

The dataset used for the experiments in this report is the e-SNLI dataset (Camburu et al., 2018). The dataset is an extension of the Stanford Natural Language Inference dataset (Bowman et al., 2015). Each datapoint in the SNLI dataset consists of a premise and a hypothesis. The premise and hypothesis can have three relationships between each other *entailment*, *neutral*, *contradiction*. The *entailment* relation means that the hypothesis is a logical continuation of the premise. The *contradiction* relation means that if the hypothesis is true, then the premise has to be false and vice-versa. The *neutral* relation means that the hypothesis neither contradicts nor entails the premise. For each premise-hypothesis pair a label is provided for encoding the relationship for that pair.

e-SNLI provides an extension to SNLI in the form of a human annotated explanation describing why the label between the premise-hypothesis pairs hold true. These explanations where sourced from Amazon Mechanical Turk with 6325 workers providing explanations. When a worker annotated an explanation they were also required to highlight words in the premise and the hypothesis that they considered essential for understanding the relation. These highlighted words where then requiered to be used in the explanation. After the crowd sourcing some postprocessing was made in order to filter out low quality or uninformative explanations. In total the dataset consists of 569 034 datapoints, with 9800 datapoints each for the validation and test set and the rest for the training set.

Premise	*A skier slides along a metal rail.*
Hypothesis	*A skier is away from the rail.*
Explanation	*A slider can slide along or away from the rail.*
Label	*contradiction*
Highlighted words	**along* *away**

Figure 1: An example of an entry in the e-SNLI dataset.

3.2 Training setup

The model architecture (referred to here on out as `Transformer` used in both experiments for generating explanations is a transformer model (Vaswani et al., 2017) with six encoder-decoder layers and with a dimensionality of 512. The feed-forward sublayers have an internal dimension of 1024. Eight attention heads are used. Instead of applying layer normalization after each sublayer as in the original implementation, a pre-layer normalization setup is used as warm up steps are not required then (Xiong et al., 2020). For label classification, an MLP consisting of three linear layers with an internal dimension of 1024 is used.

The models for generating explanations were trained on an Nvidia T4 GPU for 10 epochs with a batch size of 64 (the largest that fit on the memory). and 13 hours for experiment PredictThenExplain. The models for predicting labels were trained on an Nvidia V100 GPU for 10 epochs with a batch size of 512. The training lasted about one hour for both experiments. Due to time constraints and the long training time for the models, little to no hyperparameter tuning has been done. For both experiments, the following hyperparameters were used:

LR	Weight decay	Dropout	Optimizer
0.0001	5e-5	0.5	Adam

3.3 Experiments

Camburu et al. (2018) performed five experiments on the e-SNLI dataset in order to test the explanation generation possible after training on the dataset. In this study experiment 1, 2 and 3 were

replicated, but due to time constraints the results of experiment 2 were not considered and are not described in this paper. Experiments 4 and 5 were considered out of scope and are therefore also not described here. As one of the purposes of this study is to compare the explanations of LSTMs and transformers, the main differences of this study and that of Camburu et al. (2018) is in the model architecture.

Accuracy, BLEU score and perplexity are the evaluation methods used by Camburu et al. (2018). Accuracy for the generated explanations is calculated based on the amount of required arguments that are in the generated explanation, giving scores for each sentence between 0 and 1, with partial scores of k/n if k out of n arguments are in the generated explanation. For sentence pairs with the entailment label, the explanation has to contain justifications of all parts of the hypothesis that do not appear in the premise. For neutral and contradiction sentence pairs, the explanation has to contain at least one of the elements that make up the relation. Accuracy for the predicted labels is calculated as $n\_correct/n\_total$. The accuracy scores are in Camburu et al. (2018) and in this study calculated manually over the first 100 examples in the test dataset.

As BLEU and perplexity were found to be unsuitable for rating explanations by Camburu et al. (2018), the only quantitative measurement for the explanations in this study is the accuracy score described above.

3.3.1 Experiment PremiseAgnostic

Gururangan et al. (2018) show that a model trained only on the SNLI-hypothesis can predict the correct label with an accuracy of 67 % due to artifacts in the hypothesis, i.e. certain words in the hypothesis that are strong indicators of the label. These words can be used by the model to predict the correct label despite not having learned a good semantic representation of the hypothesis.

Camburu et al. (2018) test these assumptions on the e-SNLI dataset by training a model with a bidirectional LSTM encoder with a hidden size of 2048 and a one layer LSTM decoder. The encoder takes the hypothesis as the source input and the decoder takes the explanation as the target input. Seperately a classifier for label prediction is trained with the hypothesis as input, using the same encoder and a 3-layered MLP with 512 as internal size for prediciting the label.

For the replication of the experiment the `Transformer` model described in the previous section was used together with the described MLP for label classification. The exact same training setup was used, with the hypothesis given to the transformer's encoder and the explanation to the transformer's decoder.The training lasted approximately 11 hours for `Transformer` and approximately 1 hour for the label classifier.

3.3.2 Experiment PredictThenExplain

In this experiment Camburu et al. (2018) argue that a more natural way of solving the task is to first generate an explanation from both the premise and the hypothesis and based on that explanation then predict a label. They therefore first train a model for generating explanations from the premise and hypothesis and then train a model for predicting labels from explanations. The encoded premise u and hypothesis v are concatenated into a feature vector f along with the magnitude of the difference and the dot product of u and v:

$$f = [u, v, |u - v|, u \cdot v]$$

The decoder takes f both as an initial state and concatenated to the word embedding at each time step. Again a bidirectional LSTM with internal size 2048 is used as encoder and a one-layered LSTM is used as decoder. This setup is dubbed `Seq2Seq`. In addition to this setup, the same model is used together with attention, where two additive attention modules are separately applied on the premise and the hypothesis (dubbed `Attention`). The label classifier is again a 3-layered MLP with 512 internal size.

In our replication of `PredictThenExplain` the `Transformer` setup is used, but instead of creating a feature vector with absolute difference and the dot product, the premise and the hypothesis are separately encoded and then concatenated and passed to the decoder as the memory. The label classifier is in our replication also a 3-layered MLP with internal size 1024.

4 Results and Analysis

Table 1 shows the label and explanation accuracy for the two experiments replicated in this study. For the first experiment, `PremiseAgnostic`, the model has only access to the hypothesis and is to predict labels and explanations solely from this input. Here the transformer setup massively

***Label Accuracy* `PremiseAgnostic`**			***Explanation accuracy* `PremiseAgnostic`**		
`Transformer`	`LSTM`		`Transformer`	`LSTM`	
67%	66%		**17.28%**	6.83%	
***Label Accuracy* `PredictThenExplain`**			***Explanation Accuracy* `PredictThenExplain`**		
`Transformer`	`Seq2Seq`	`Attention`	`Transformer`	`Seq2Seq`	`Attention`
61.29%	81.59%	81.71%	**54.16%**	49.8%	64.27%

Table 1: Label and explanation accuracy for experiment 1 PremiseAgnostic, and experiment 3 PredictAndExplain. The model *LSTM* is the setup described in section 3.2.1. *Seq2Seq* and *Attention* are described in section 3.2.2.

outperforms the LSTM setup, achieving an explanation accuracy of 17.28% compared to 6.83%. The label accuracy for this experiment are comparable for both model setups: 67% for the transformer architecture and 66% for the LSTM. These label prediction results match the results of Gururangan et al. (2018) for the same hypothesis-only label prediction, indicating that 67% is close to the limit when predicting labels solely based on the hypothesis. Camburu et al. (2018) state that it is approximately 10x more difficult to predict explanations than labels with the LSTM architecture. This evidently does not hold for the transformer architecture, showing that transformers appear to be better at predicting explanations than at predicting labels, relative to the LSTM's performance. This comparison should be taken with a grain of salt however, as the accuracy scores for label vs explanation are not the same: the label accuracy score is an objective measurement of many times the model outputs the same label as the gold; the explanation accuracy score is instead a subjective measurement of what the annotator deems is a correct explanation. To compare the scores in this sense is therefore not optimal as they are not measurements of the same thing.

For experiment `PredictThenExplain`, the model encodes the premise and the hypothesis and a concatenation of these are passed to the decoder. The explanation accuracy scores for this experiment are 54.16% for model `transformer`, 49.8% for model `Seq2Seq` and 64.27% for `Attention`. While the transformer performs better than the non-attention LSTM model, the LSTM model with attention modules outperform the transformer by a large margin. Considering that in experiment `PremiseAgnostic`, the transformer is much better than the LSTM model, these results are rather surprising. The label accuracy results are if anything even more surprising: both LSTM setups get 20 percentage points higher accuracy than the transformer. We find this surprising for three reasons. First, a difference of 20 percentage points is a massive difference considering that transformers typically outperform LSTMs for this task; second, this difference is much higher than the differences in explanation accuracy between the three models; third, the label accuracy for `Seq2Seq` and `Attention` are almost the same while there is a big gap between their explanation accuracies. The third point we especially find strange. In this experiment, the labels are predicted from the generated explanations. If the generated explanations are of poorer quality, then it stands to reason that it also should be more difficult for a classifier to predict the correct label from these. `Seq2Seq` can therefore be expected to have a lower label accuracy than `Attention` as the explanation accuracy is worse. Instead the accuracy scores are almost the same.

4.1 Quantitative analysis

For experiment PredictThenExplain there is a discrepancy between the accuracy of the labels and the accuracy of the explanations in the sense that the label accuracy does not seem to depend on the accuracy of the explanations. A reason for this discrepancy could be because of the template-like structure of the e-SNLI explanations. It is often the case throughout the dataset that explanations for the same type of relation are constructed in the same manner. For instance, many explanations of *neutral* premise-hypothesis pairs have the structure:

- "Just because [element of premise] doesn't mean that [element of hypothesis]".

For explanations of *entailment* relations a typical structure is:

- "[element of premise] is [element of hypothesis]",

and for explanations of *contradiction* relations a typical structure is:

- "One cannot [element of premise] and [element of hypothesis] at the same time".

This means that label predictions would not be so much dependant on the generated explanation accurately justifying the relation of the premise and hypothesis as much as it would be dependant on the generated containing a typical syntactic structure for its relation type. The examples of generated explanations provided by Camburu et al. (2018) (figure 2) strengthen this theory as all follow the above mentioned pattern. This would also explain why the label accuracy for `Transformer` is so much worse than for the LSTM-models as the explanations generated by `Transformer` often generate explanations with correct arguments, but with the wrong structure, which then confuses the label classifier. For instance, frequently, entailment explanations have the structure "[element of premise] is *not* [element of hypothesis]", giving instead a typical *contradiction* structure. That models trained on the e-SNLI dataset learn to generate explanations with generic structures could potentially be an issue as they therefore do not learn to explain, but rather fill templates with different words depending on the context. However, even with this template structure we note that the generated explanations can still be good at explaining the sentence pairs' relation and as such we do not think that this is too much of an issue for evaluating the quality of explanations. This does however make explanations unsuitable for label prediction, as the predictions are then based more on syntactic structure rather than on the model's actual understanding of language.

Another issue is with the explanation accuracy as the transformer model performs worse than the LSTM model with an additive attention module. A reason for this could be that transformer models tend to need both a larger dataset and a larger batch size during training in order to perform well (Xu et al., 2021). Additionally, scaling up the size of the model tends to improve performance. We noticed this for our experiments as well, where tentative results with a model of 256 dimensions and four encoder-decoder layers were substantially worse than the final results with the full size transformer model. An even larger model would therefore probably improve explanation accuracy more. Finally, the lack of hyperparameter testing would most likely have improved explanation quality. ***As mentioned in section 3.2, due to time constraints it was not possible to perform any extensive hyperparameter testing.***

4.2 Qualitative analysis

The explanations generated by our model vary greatly in quality. For sentence pair (1) in figure 2, `Transformer` generated the explanation

- *three people are not three people.*,

to which it predicted the label *contradiction*, with both label and explanation being incorrect. This is a good example of the phenomenon mention earlier, where the model learns patterns in explanation structure, but does not learn to generate explanations with correct structure. Here the word "not" turns the explanation in to a *contradiction* explanation. Had it left out that single word, both label and explanation would have been (partially) correct.

For sentence pair (2), `Transformer` generated the explanation

- *three firefighters are putting out a fire station inside a subway station.*,

and predicted the label *entailment*. Here the explanation is much farther away from being correct. The explanation is using the *entailment* structure however, meaning that the predicted label is logical based on the explanation.

For sentence pair (3), `Transformer` generated

- *a doctor is not a man.*,

and the prediction *contradiction*, giving both a correct label and explanation (for *contradiction*, only one argument needs to be correct, and in this sentence pair, a doctor is not a man, making one of the arguments correct.).

As mentioned the quality of the explanations is very varied. Some are excellent such as the **Good explanations** below, both in terms of accuracy score and as a natural language explanation of the sentence relation; and some are very poor such as the **Bad explanation**.

- **Good explanations:**
  - Premise: A woman working long hours.

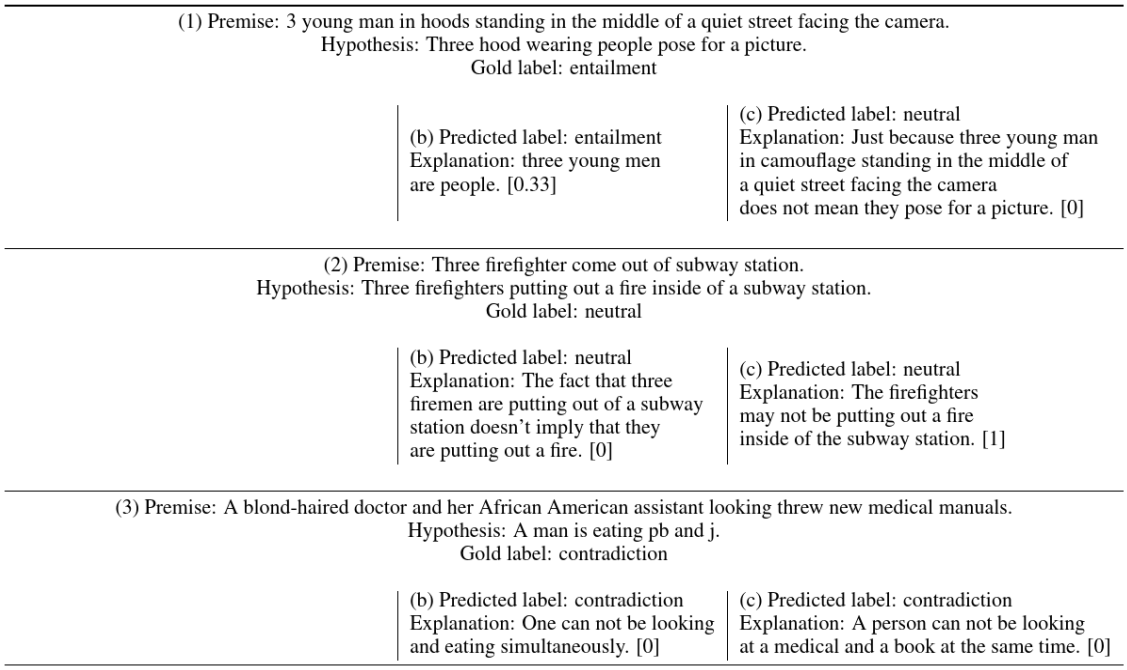

Figure 2: Generated explanations by Camburu et al. (2018). b) were generated with model `Seq2Seq` and c) were generated with model `Attention`. The number in the brackets is the accuracy score.

  - Hypothesis: A woman is sleeping.
  - Gold label: *contradiction*
  - Predicted label: *contradiction*
  - Explanation: *working and sleeping are different actions*
  - ———————————————————
  - Premise: A man playing an electric guitar on stage.
  - Hypothesis: A man is performing for cash.
  - Gold label: *neutral*
  - Predicted label: *neutral*
  - Explanation: *just because a man is playing an electric guitar on stage does not mean he is performing for cash.*

- **Bad explanation:**

  - Premise: A person with a purple shirt is painting an image of a woman on a white wall.
  - Hypothesis: A woman paints a portrait of a monkey.
  - Gold label: *contradiction*
  - Predicted label: *contradiction*
  - Explanation: *a woman is not a woman*

5 Conclusion

Before beginning the experiments, we expected the two models with attention modules `Transformer` and `Attention` to be able to generate better explanations than the model without attention `Seq2Seq`. This hypothesis is confirmed by our experiments to be true. The other hypothesis, that `Transformer` should perform better at this task than the LSTM model with additive attention, `Attention`, did not realize. We expect however that a larger transformer model trained with a larger batch size should perform better than the smaller current version. As is evident by the qualitative analysis, `Transformer` is still able to formulate good explanations despite its quantitative shortcomings. We therefore conclude that a transformer is capable of generating natural language explanations to a label prediction task, but further steps need to be taken in order to ensure more consistent quality in the generated explanations.

6 Ethical considerations

Potential ethical issues with this study concern the dataset, or rather the creation of the dataset. The explanations in the e-SNLI dataset were created by crowdworkers sourced from Amazon Mechanical Turk, a service where one can outsource a sim-

ple task, such as annotation, to several workers. There are some ethical issues with AMT however. The payment for completing tasks is very low, 2 dollars an hour. Many workers state that AMT is their primary source of income, making the service very exploitative. Another issue is the fact that AMT-workers are not employed by Amazon or any other party, meaning that they do not get benefits that typically come with an employment such as insurances, unemployment payments etc.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2017. QUINT: Interpretable question answering over knowledge bases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–66, Copenhagen, Denmark. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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 (EMNLP)*. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Oana-Maria Camburu, Tim Rocktaschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Felix Stahlberg, Danielle Saunders, and Bill Byrne. 2018. An operation sequence model

for explainable neural machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 175–186, Brussels, Belgium. Association for Computational Linguistics.

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*, volume 30. Curran Associates, Inc.

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, Brussels, Belgium. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021. Optimizing deeper transformers on small datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2089–2102, Online. Association for Computational Linguistics.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.