On the Lack of Robust Interpretability of Neural Text Classifiers

Muhammad Bilal Zafar
Amazon
zafamuh@amazon.com

Michele Donini
Amazon
donini@amazon.com

Dylan Slack*
University of California, Irvine
dslack@uci.edu

Cédric Archambeau
Amazon
cedrica@amazon.com

Sanjiv Das
Amazon & Santa Clara University
sanjivda@amazon.com

Krishnaram Kenthapadi
Amazon
kenthk@amazon.com

Abstract

With the ever-increasing complexity of neural language models, practitioners have turned to methods for understanding the predictions of these models. One of the most well-adopted approaches for model interpretability is *feature-based interpretability*, i.e., ranking the features in terms of their impact on model predictions. Several prior studies have focused on assessing the fidelity of feature-based interpretability methods, i.e., measuring the impact of dropping the top-ranked features on the model output. However, relatively little work has been conducted on quantifying the robustness of interpretations. In this work, we assess the robustness of interpretations of neural text classifiers, specifically, those based on pretrained Transformer encoders, using two randomization tests. The first compares the interpretations of two models that are identical except for their initializations. The second measures whether the interpretations differ between a model with trained parameters and a model with random parameters. Both tests show surprising deviations from expected behavior, raising questions about the extent of insights that practitioners may draw from interpretations.

1 Introduction

In recent years, large scale language models like BERT and RoBERTa have helped achieve new state-of-the-art performance on a variety of NLP tasks (Devlin et al., 2019; Liu et al., 2019). While relying on vast amounts of training data and model capacity has helped increase their accuracy, the reasoning of these models is often hard to comprehend. To this end, several techniques have been proposed to interpret the model predictions.

Perhaps the most widely-adopted class of interpretability approaches is that of *feature-based interpretability* where the goal is to assign an importance score to each of the input features. These scores are also called *feature attributions*. Several methods in this class (e.g., SHAP (Lundberg and Lee, 2017), Integrated Gradients (Sundararajan et al., 2017)) possess desirable theoretical properties making them attractive candidates for interpretability.

Benchmarking analyses often show that these methods possess *high fidelity*, i.e., removing features marked important by the interpretability method from the input indeed leads to significant change in the model output as expected (Atanasova et al., 2020; Lundberg and Lee, 2017).

However, relatively few investigations have been carried out to understand the *robustness* of feature attributions. To explore the robustness, we conduct two tests based on randomization:

Different Initializations Test: This test operationalizes the *implementation invariance* property of Sundararajan et al. (2017). Given an input, it compares the feature attributions between two models that are identical in every aspect—that is, trained with same architecture, with same data, and same learning schedule—except for their randomly chosen initial parameters. If the predictions generated by these two models are also identical, one would also expect the feature attributions to be the same for such *functionally equivalent* models. If the attributions in two cases are not the same, two users examining the *same input* may deem the same features to have *different importance* based on the model that they are consulting.

Untrained Model Test: This test is similar to the test of Adebayo et al. (2018). Given an input, it compares the feature attributions generated on a fully trained model with those on a randomly initialized untrained model. The test evaluates whether feature attributions on a fully trained model differ from the feature attributions computed on an

*Work done during internship at Amazon.

untrained model as one would expect.

We conduct the two tests on a variety of text classification datasets. We quantify the feature attribution similarity using *interpretation infidelity* (Arras et al., 2016) and *Jaccard similarity* (Tanimoto, 1958). The results suggest that: (i) Interpretability methods fail the different initializations test. In other words, two functionally equivalent models lead to different ranking of feature attributions; (ii) Interpretability methods fail the untrained model test, i.e., the fidelity of the interpretability method on an untrained model is better than that of random feature attributions.

These findings may have important implications for how the prediction interpretations are shown to the users of the model, and raise interesting questions about reliance on these interpretations. For instance, if two functionally equivalent models generate different interpretations, to what extent can a user act upon them, e.g., investing in a financial product or not. We discuss these implications and potential reasons for this behavior in §4.

Related work. Model interpretability has different aspects: local (e.g. Lundberg and Lee, 2017) vs. global (e.g. Tan et al., 2018), feature-based (e.g. Lundberg and Lee, 2017) vs. concept-based (e.g. Kim et al., 2018) vs. hidden representation-based (Li et al., 2016). See Gilpin et al. (2018); Guidotti et al. (2018) for an overview. In this paper, we focus on feature-based interpretability, which is a well-studied and commonly used form (Bhatt et al., 2020; Tjoa and Guan, 2020). Specifically, this class consists of a relatively large number of methods, of which some have been shown to satisfy desirable theoretical properties (e.g., SHAP (Lundberg and Lee, 2017), Integrated Gradients (Sundararajan et al., 2017), and DeepLIFT (Shrikumar et al., 2017)).

There are several important aspects of interpretation robustness. Some prior studies have considered interpretability in the context of *adversarial robustness* where the goal often is to actively fool the model to generate misleading feature attributions. See for instance Anders et al. (2020); Dombrowski et al. (2019); Ghorbani et al. (2019); Slack et al. (2020). In this work, rather than focusing on targeted changes in the input or the model, we explore robustness of feature attribution methods to various kinds of randomizations.

Several prior works have focused on quantifying quality of interpretations. See for instance, Adebayo et al. (2020); Alvarez-Melis and Jaakkola (2018); Chalasani et al. (2020); Chen et al. (2019); Hooker et al. (2019); Lakkaraju et al. (2020); Meng et al. (2018); Ribeiro et al. (2016); Tomsett et al. (2020); Yang and Kim (2019). Closest to ours is the work of Adebayo et al. (2018), which is based on checking the saliency maps of randomly initialized image classification models. However, in contrast to Adebayo et al., we consider text classification. Moreover, while the analysis of Adebayo et al. is largely based on visual inspection, we extend it by considering automatically quantifiable measures. We also extend the analysis to non-gradient based methods (SHAP).

2 Setup

We describe the datasets, models, and interpretability methods considered in our analysis.

Datasets. We consider four different datasets covering a range of document lengths and number of label classes. The datasets are: (i) **FPB:** The Financial Phrase Bank dataset (Malo et al., 2014) where the task is to classify news headlines into one of three sentiment classes, namely, positive, negative, and neutral. (ii) **SST2:** The Stanford Sentiment Treebank 2 dataset (Socher et al., 2013). The task is to classify single sentences extracted from movie reviews into positive or negative sentiment classes. (iii) **IMDB:** The IMDB movie reviews dataset (Maas et al., 2011). The task is to classify movie reviews into positive or negative sentiment classes. (iv) **Bios:** The Bios dataset of De-Arteaga et al. (2019). The task is to classify the profession of a person from their biography. Table 5 in Appendix A shows detailed dataset statistics.

2.1 Models

We consider four pretrained Transformer encoders: BERT (**BT**), RoBERTa (**RB**), DistilBERT (**dBT**), and DistilRoBERTa (**dRB**). The encoder is followed by a pooling layer to combine individual token embeddings, and a classification head. Appendix B.1 describes the detailed architecture, training and hyperparameter tuning details. After training and hyperparameter tuning, the best model is selected based on validation accuracy and is referred to as `Init#1`.

Different Initializations Test. Recall from §1 that this test involves comparing two identical models trained from different initializations. The second model, henceforth referred to as `Init#2`, is

trained using the same architecture, hyperparameters and training strategy as `Init#1`, but starting from a different set of initial parameters. Since we start from pretrained encoders, the encoder parameters are not intialized. For each layer in the rest of the model, a set of initial parameters different from those in `Init#1` is obtained by calling the parameter initialization method of choice for this layer—*He initialization* (He et al., 2015) in this case—but with a different random seed.

Untrained Model Test. Recall that this test involves comparing the trained model (`Init#1`) with a randomly initialized untrained model, henceforth called `Untrained`. To obtain `Untrained`, we start from the `Init#1`, and randomly initialize the fully connected layers attached on top of the Transformer encoders (the encoder weights are not randomized). The initialization strategy is the same as in Different Initializations Test.

2.2 Interpretability methods

We consider a mix of gradient-based and model agnostic methods. Specifically: Vanilla Saliency (**VN**) of Simonyan et al. (2014), SmoothGrad (**SG**) of Smilkov et al. (2017), Integrated Gradients (**IG**) of Sundararajan et al. (2017), and KernelSHAP (**SHP**) of Lundberg and Lee (2017). We also include random feature attribution (**RND**) which corresponds to each feature being assigned an attribution from the uniform distribution, $\mathcal{U}(0, 1)$. Appendix B.2 provides details about the parameters chosen for the interpretability methods.

Given an input text document, we tokenize the text using the tokenizer of the corresponding encoder. Finally, for each input feature (that is, token), the feature attribution of the gradient-based methods is a vector of the same length as the token input embedding. For scalarizing these vector scores, we use the L2-norm strategy of Arras et al. (2016) and the Input $\odot$ Gradient strategy of Ding et al. (2019).

2.3 Interpretability Metrics

To compare the feature attributions by various interpretability methods, we use the following metrics.

(In)Fidelity: Given an input text which has been split into L tokens, $\mathbf{t} = [t_1, \ldots, t_L]$, get the vector $\mathbf{\Psi}(\mathbf{t}) = [\psi(t_1), \ldots, \psi(t_L)]$ of feature attributions of the corresponding tokens using the interpretability method to be evaluated. Drop the features from $\mathbf{t}$ in the decreasing order of attribution score until the model prediction changes from the original prediction (with all tokens present). Infidelity is defined as the % of features that need to be dropped until the prediction changes. A better interpretability method is expected to need a lower fraction of features to be dropped until the prediction change. We simulate feature dropping by replacing the corresponding input token with the model's unknown vocabulary token.

	BT	**RB**	**dBT**	**dRB**
FPB	0.83	0.85	0.82	0.84
SST2	0.87	0.91	0.88	0.90
IMDB	0.92	0.95	0.93	0.94
Bios	0.86	0.86	0.86	0.86

Table 1: Test accuracy with `Init#1`. For any given dataset, all encoders lead to a similar accuracy.

The infidelity metric has appeared in many closely related forms in a number of studies evaluating model interpretability (Arras et al., 2016; Atanasova et al., 2020; DeYoung et al., 2020; Fong et al., 2019; Lundberg and Lee, 2017; Samek et al., 2017). All of these forms operate by iteratively hiding features in the order of their importance and measuring the change in the model output, e.g., in predicted class probability, or the predicted label itself. We chose number of tokens to prediction change, which is closely aligned with (Arras et al., 2016), due to its simplicity as compared to more involved metrics relying on AUC-style measures (Atanasova et al., 2020; Samek et al., 2017).

Jaccard Similarity: It is common to show top few most important features to users as model interpretations. See for instance, Ribeiro et al. (2016) and Schmidt and Biessmann (2019). In order to measure the similarity between feature attributions generated by different methods, we use the Jaccard@K% metric. Given an input $\mathbf{t}$, let $\mathbf{s}_i$ be the set of top-K% tokens, when the tokens are ranked based on their importance as specified by an attribution output $\mathbf{\Psi}_i$. Then, given two attribution outputs $\mathbf{\Psi}_i$ and $\mathbf{\Psi}_j$, Jaccard@K% measures the similarity between them as: $J(i, j) = \frac{|\mathbf{s}_i \cap \mathbf{s}_j|}{|\mathbf{s}_i \cup \mathbf{s}_j|}$. If the top-K% tokens by the two attributions $\mathbf{\Psi}_i$ and $\mathbf{\Psi}_j$ are the same, then $J(i, j) = 1$. In case of no overlap in the top-K% tokens, $J(i, j) = 0$.

Appendix D shows some examples of Jaccard@K% computation.

	FPB				SST2				IMDB				Bios			
	BT	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**
VN	68	64	61	59	50	54	53	54	46	53	52	50	28	22	27	25
SG	68	63	61	58	46	50	53	51	45	52	52	48	27	23	27	23
IG	68	62	61	59	38	48	50	49	39	47	46	47	24	20	17	22
SHP	60	54	52	43	22	25	30	27	10	19	11	13	15	15	15	14
RND	72	71	68	70	61	67	66	69	58	58	63	66	51	51	49	56

Table 2: Mean infidelity of different interpretability methods for `Init#1` (shown as %). Lower values are better.

3 Results

Table 1 shows the test set accuracy of `Init#1` model with different encoders on all the datasets. For all the datasets, different encoders lead to a very similar accuracy. Tables 7 in Appendix C.1 shows the prediction accuracy for `Untrained`.

Infidelity. Table 2 shows the infidelity of different interpretability methods on the best performing models (`Init#1`). The table shows that: (i) As expected, the infidelity of all interpretability methods is better than RND; (ii) SHP provides the best performance, followed by IG; (iii) For a given dataset, even though different encoders have very similar accuracy (Table 1), the infidelity of the same interpretability method for different encoders can vary widely, e.g., SHP on IMDB; (iv) There is no particularly discernable correlation between the models and their infidelity, for instance, with FPB dataset, the distilled Transformers provides same or lower infidelity as compared to the original counterparts (BERT, RoBERTa), whereas the trend is reversed for SST2 data.

Moreover, gradient-based methods in Table 2 use the L2-norm reduction (§2.2). Table 8 in Appendix C.2 shows that in most cases, the performance is much worse when using the Input $\odot$ Gradient reduction. Hence, for the rest of the analysis, we only use L2-norm reduction.

Different Initializations Test. Comparing `Init#1` and `Init#2` in Table 6 in Appendix C.1—two otherwise identical models with only difference being the random initial parameters, shows that a vast majority of predictions are common between the two models: meaning that the *two models are almost functionally equivalent*.

We now compare the similarity in feature attributions of two functionally equivalent models. Since the feature attributions are generated w.r.t. the predicted class, our similarity analysis is limited to samples where both models generate the same prediction. Figure 1 shows Jaccard@25% when comparing the feature attributions of `Init#1` vs. `Init#2`, and `Init#1` vs. `Untrained`. The figure shows that (i) for the functionally equivalent models `Init#1` vs. `Init#2`, Jaccard@25% is far from the ideal value of 1.0 — in fact, for IG, the value drops to almost 0.5; (ii) When comparing the top-ranked feature attributions of `Init#1` vs. `Init#2`, and `Init#1` vs. `Untrained`, the former should show a much bigger overlap than latter, but this is not the case, except for SHP.

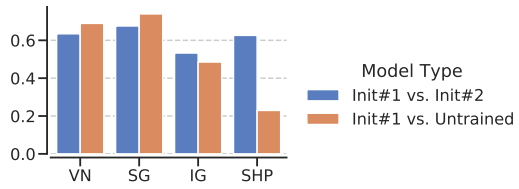

Figure 1: [**BT** on SST2 data] Comparing the mean Jaccard@25 between different model types (`Init#1` vs. `Init#2` and `Init#1` vs. `Untrained`).

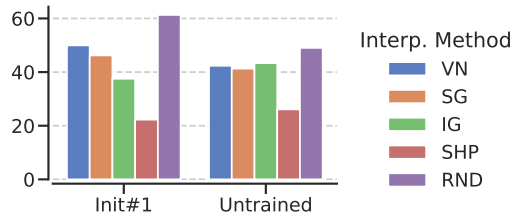

Figure 2: [**BT** on SST2 data] Mean infidelity of interpretability methods on `Init#1` and `Untrained`.

Tables 3 and 4 show the results for rest of the cases, revealing similar insights. Specifically, the Jaccard@25% between `Init#1` and `Untrained` for VN averaged over all 16 cases (four datasets, four models in Table 3) is 68, whereas the same is 69 when comparing `Init#1` and `Untrained` (Table 4). The same comparison yields 67 vs. 64 for SG, 56 vs. 50 for IG and 65 vs. 29 for SHP. Moving beyond averages, we also counted for each of the cases in Table 3, the number of times Jaccard@25% between `Init#1` and `Init#2` is within 10 units (Jaccard@25% ranges from 0-100) of the Jaccard@25% between `Init#1` and `Untrained` for the corresponding

	FPB				SST2				IMDB				Bios			
	BT	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**
VN	65	52	77	67	63	53	73	63	72	47	78	72	76	70	81	75
SG	53	66	70	62	68	63	70	66	57	45	76	75	75	70	81	78
IG	46	35	68	51	53	37	71	50	65	36	82	68	49	51	73	68
SHP	57	55	64	55	63	65	66	62	68	50	75	67	72	71	75	69

Table 3: Jaccard@25% between the feature attributions for `Init#1` vs. `Init#2` models (shown as %).

	FPB				SST2				IMDB				Bios			
	BT	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**
VN	65	60	75	71	69	63	73	69	64	79	71	66	71	68	70	74
SG	47	70	67	61	74	72	70	75	44	53	63	52	66	71	60	75
IG	41	40	55	49	49	45	60	49	37	75	63	51	40	41	55	47
SHP	25	30	19	29	23	22	14	33	15	85	23	18	32	45	24	27

Table 4: Jaccard@25% between the feature attributions for `Init#1` vs. `Untrained` models (shown as %).

pair in Table 4. The numbers are 14/16 for VN, 12/16 for SG, 7/16 for IG and 0/16 for SHP.

We selected K=25 as it corresponds to comparing the 25% most important features between the two models. Selecting K=10 (comparing 10% most important features) leads to similar outcomes: 13/16 for VN, 11/16 for SG, 6/16 for IG and 0/16 for SHP.

In other words, *the attribution overlap between two functionally equivalent models can be similar to that between a trained vs. an untrained model.*

Untrained Model Test. Figure 2 shows the infidelity of different methods on SST2 dataset with a **BT** model. We note that the performance of RND is better (lower infidelity) for the untrained model (`Untrained`) than for the trained model (`Init#1`). Furthermore, even for the untrained model (`Untrained`), all interpretability methods have a better fidelity than RND. In fact, for SHP, the infidelity is almost half of RND. Table 9 in Appendix C shows a similar pattern for the rest of the datasets and models.

In short, *even for an untrained model, the interpretability methods lead to better-than-random-attribution fidelity.* The insights highlight the need for baselining the fidelity metric with untrained models before using it as an evaluation measure.

4 Conclusion & Future Work

We carried out two tests to assess robustness of several popular interpretability methods on Transformer-based text classifiers. The results show that both gradient-based and model-agnostic methods can fail the tests.

These observations raise several **interesting questions**: if the fidelity of the interpretations is reasonably high on even an untrained model, to what extent does the interpretability method reflect the data-specific vs. data-independent behavior of the model? If two functionally equivalent models lead to different feature attributions, to what extent can the practitioners rely upon these interpretations to make consequential decisions?

One cause of the non-robust behavior could be the redundancy in text where several input tokens may provide evidence for the same class (e.g., several words in input review praising the movie). Another reason, related to the first, could be the pathologies of neural models where dropping most of the input features could still lead to highly confident predictions (Feng et al., 2018).[1] Dropping individual features can also lead to out-of-distribution samples, further limiting the effectiveness of methods and metrics that rely on simulating feature removal (Kumar et al., 2020; Sundararajan and Najmi, 2020). Systematically analyzing the root causes, and designing interpretability measures that are cognizant of the specific characteristics of text data—preferably with human involvement (Chang et al., 2009; Doshi-Velez and Kim, 2017; Hase and Bansal, 2020; Nguyen, 2018; Poursabzi-Sangdeh et al., 2021; Schmidt and Biessmann, 2019)—is a promising research direction. Similarly, extending the Untrained Model Test to study the effect of randomization of pre-trained embedding models on interpretability is another direction for exploration.

[1] Note, however, that the insights of Feng et al. relate to *dropping least important tokens first*, whereas when computing infidelity, one drops the most important first.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9525–9536, Red Hook, NY, USA. Curran Associates Inc.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. 2020. Debugging Tests for Model Explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards Robust Interpretability with Self-explaining Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 7786–7795, Montréal, Canada. Curran Associates Inc.

Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. 2020. Fairwashing Explanations with Off-manifold Detergent. In *International Conference on Machine Learning*, pages 314–323. PMLR.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining Predictions of Non-Linear Classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2020. Explainable Machine Learning in Deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 648–657, Barcelona, Spain. Association for Computing Machinery.

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. 2020. Concise Explanations of Neural Networks using Adversarial Training. In *International Conference on Machine Learning*, pages 1383–1391. PMLR.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems*, volume 22, pages 288–296.

Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. 2019. Robust Attribution Regularization. In *Advances in Neural Information Processing Systems*, volume 32.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 120–128, New York, NY, USA. Association for Computing Machinery.

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 {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven Word Alignment Interpretation for Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32:13589–13600.

Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*. ArXiv: 1702.08608.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of Neural Models Make Interpretations Difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Ruth Fong, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding Deep Networks via Extremal Perturbations and Smooth Masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2950–2958.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of Neural Networks Is Fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688.

L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. 2018. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5):93:1–93:42.

Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, USA. IEEE Computer Society.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A Benchmark for Interpretability Methods in Deep Neural Networks. *Advances in Neural Information Processing Systems*, 32:9737–9748.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A Unified and Generic Model Interpretability Library for PyTorch. *arXiv:2009.07896 [cs, stat]*. ArXiv: 2009.07896.

I. Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR.

Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. Robust and Stable Black Box Explanations. In *International Conference on Machine Learning*, pages 5628–5638. PMLR.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, USA. Association for Computational Linguistics.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Qingjie Meng, Christian Baumgartner, Matthew Sinclair, James Housden, Martin Rajchl, Alberto Gomez, Benjamin Hou, Nicolas Toussaint, Veronika Zimmer, Jeremy Tan, Jacqueline Matthew, Daniel Rueckert, Julia Schnabel, and Bernhard Kainz. 2018. Automatic Shadow Detection in 2D Ultrasound Images. In *Data Driven Treatment Response Assessment and Preterm, Perinatal, and Paediatric Image Analysis*, Lecture Notes in Computer Science, pages 66–75, Cham. Springer International Publishing.

Dong Nguyen. 2018. Comparing Automatic and Human Evaluation of Local Explanations for Text Classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1069–1078, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32:8026–8037.

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and

Hanna Wallach. 2021. Manipulating and Measuring Model Interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ArXiv: 1802.07810.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673.

Philipp Schmidt and Felix Biessmann. 2019. Quantifying Interpretability and Trust in Machine Learning Systems. In *AAAI-19 Workshop on Network Interpretability for Deep Learning*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *2nd International Conference on Learning Representations (Workshop Track)*.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pages 180–186, New York, NY, USA. Association for Computing Machinery.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: Removing Noise by Adding Noise. In *ICML Workshop on Visualization for Deep Learning*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? In *Chinese Computational Linguistics*, Lecture Notes in Computer Science, pages 194–206, Cham. Springer International Publishing.

Mukund Sundararajan and Amir Najmi. 2020. The Many Shapley Values for Model Explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*.

Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. 2018. Learning Global Additive Explanations for Neural Nets Using Model Distillation. *arXiv:1801.08640 [cs, stat]*. ArXiv: 1801.08640.

T. T. Tanimoto. 1958. *An elementary mathematical theory of classification and prediction*. New York: International Business Machines Corporation.

Erico Tjoa and Cuntai Guan. 2020. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21. ArXiv: 1907.07374.

Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. 2020. Sanity Checks for Saliency Metrics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6021–6029.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models. In *Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations)*, pages 7–12.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengjiao Yang and Been Kim. 2019. Benchmarking Attribution Methods with Relative Feature Importance. *arXiv:1907.09701 [cs, stat]*. ArXiv: 1907.09701.

Appendix A Datasets

	N	K	D	Δ_K
FPB	4,846	3	22±10	47
SST2	9,613	2	19±9	3
IMDB	50,000	2	227±168	0
Bios	397,907	28	61±28	30

Table 5: Dataset details. The columns are: # of samples (N), # of classes (K), average ± standard deviation # of words per document (D), and Class Imbalance (Δ_K). Class imbalance is measured as % prevalence of the most prevalent minus the least prevalent class.

Appendix B Reproducibility

B.1 Architecture, data & training details

We insert a classification head on top of the pre-trained encoder. The end-to-end classifier has the following architecture: Encoder → Avg. pooling → FC-layer (512-units) → RELU → FC-layer (K units), where K is the number of classes. The maximum sequence length of the encoder is set to 128 for FPB and SST2 datasets, 512 for IMDB reviews and 200 for the Bios data.

Each dataset is split into a $80\% - 20\%$ train-test set. 10% of the training set is used as a validation set for hyperparameter optimization. Accuracy and overlap statistics are reported on the test set.

We used the following hyperparameter ranges: learning rate $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and the number of last encoder layers to be fine-tuned $\{0, 2\}$. Fine tuning last few layers of the encoder, as opposed to all the layers, has been shown to lead to superior test set performance (Sun et al., 2019).

We use the AdamW optimizer (Loshchilov and Hutter, 2019). The maximum number of training epochs is 25. We use early stopping with a patience of 5 epochs: if the validation accuracy does not increase for 5 consecutive epochs, we stop the training. The model training was done using PyTorch (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020) libraries.

B.2 Interpretability methods implementation

Owing to the large runtime of methods like SHAP and Integrated Gradients, interpretations are only computed for a randomly chosen 1000 subsample from the test set. Consequently, metrics like fidelity and Jaccard@K% are reported only on this subset.

	BT	RB	dBT	dRB
FPB	88	94	90	89
SST2	94	96	93	94
IMDB	98	97	97	98
Bios	95	95	95	95

Table 6: A vast percentage of predictions are common between the `Init#1` and `Init#2`, indicating that the models are almost *functionally equivalent*.

	BT	RB	dBT	dRB
FPB	0.19	0.19	0.34	0.34
SST2	0.21	0.22	0.30	0.80
IMDB	0.27	0.51	0.20	0.46
Bios	0.03	0.02	0.01	0.01

Table 7: Test accuracy with `Untrained`.

Vanilla Saliency and SmoothGrad are implemented using the PyTorch `autograd` function. Integrated Gradients and SHAP are implemented using Captum (Kokhlikyan et al., 2020). The parameters of these methods are:

SmoothGrad. Requires two parameters. (i) Number of iterations: Following AllenNLP Interpret (Wallace et al., 2019), we use a value of 10. (ii) Variance of the Gaussian noise $\mathcal{N}(0, \sigma^2)$. The default value of 0.01 leads to attributions that are almost identical to Vanilla Saliency. So we try different values of $\sigma \in \{0.01, 0.05, 0.1, 0.2\}$ and select the one with the lowest infidelity.

Integrated Gradients. Requires setting two parameters. (i) Number of iterations: we use Captum (Kokhlikyan et al., 2020) default of 50. (ii) Feature Baseline: IG requires specifying a baseline (Sundararajan et al., 2017) that has the same dimensionality as the model input, but consists of 'non-informative' feature values. We construct the baseline by computing the embedding value of the unknown vocabulary token and repeating it N times where N is the maximum sequence length of the model.

KernelSHAP. Requires two parameters. (i) Number of feature coalitions: Following the author implementation,[2] we use a value of $2L + 2^{11}$, where L is the number of input tokens in the text. (ii) Dropped token value: SHAP operates by dropping subsets of tokens and estimating model output on these perturbed inputs. We simulate dropping of a

[2] https://github.com/slundberg/shap

	FPB				SST2				IMDB				Bios			
	BT	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**
VN	70	70	65	71	62	67	57	70	60	58	51	69	52	53	52	58
SG	70	69	64	69	61	67	54	68	60	58	45	68	51	51	47	58
IG	75	73	72	71	71	70	65	66	74	64	68	70	59	55	48	58

Table 8: Mean infidelity of gradient-based interpretability methods when considering *Input* $\odot$ *gradient* reduction of Ding et al. (2019) for `Init#1` (shown as %). In most cases, the performance is worse than when using the L2-norm reduction (Table 2 in Section 3). Lower values are better.

	FPB				SST2				IMDB				Bios			
	BT	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**	**BT**	**RB**	**dBT**	**dRB**
VN	75	30	33	51	42	43	36	33	35	99	42	43	14	14	11	12
SG	74	30	33	51	41	40	36	31	36	99	42	43	13	15	11	12
IG	73	32	31	55	43	42	31	35	49	100	39	42	17	17	11	13
SHP	56	19	23	36	26	40	23	27	9	99	35	19	9	11	7	9
RND	73	36	43	62	49	49	43	51	42	100	49	46	25	32	21	27

Table 9: Mean infidelity of different interpretability methods for `Untrained` (shown as %). Lower values are better.

token by replacing its embedding value with that of the unknown vocabulary token.

Appendix C Additional results

C.1 Accuracy & prediction commonality

Table 6 shows the fraction of predictions common between `Init#1` and `Init#2`.

Table 7 shows the accuracy of `Untrained`. As expected, the accuracy of `Untrained` is much smaller than with trained models.

C.2 Input $\odot$ Gradient reduction

Table 8 shows the infidelity of gradient-based interpretability methods when using the Input $\odot$ Gradient dot product reduction of Ding et al. (2019). When comparing the results to those with L2 reduction in Table 2, we notice that in *all except two cases* (VN and SG on **dBT** with IMDB data), the performance is worse.

C.3 Infidelity with `Untrained` model

Table 9 shows the infidelity for the `Untrained` model. Much like Figure 2, the table shows that in several cases, the performance of the feature attribution methods (most notably SHP) can be much better than random attribution (RND).

The table also shows an exception for **dBT** on IMDB dataset where for all methods, the infidelity is near 100. This behavior is likely an artefact of the particular initial parameters due to which the model always predicts a certain class irrespective of the input.

Appendix D Examples of top-ranked tokens

We now show some examples of Jaccard@K% computation. The examples show the input text, different models, and top-K% tokens ranked w.r.t. their importance. The attribution method used was VN.

Example 1: SST2 data. Comparing `Init#1` and `Init#2`. Both models predict the sentiment to be positive.
Text. at heart the movie is a deftly wrought suspense yarn whose richer shadings work as coloring rather than substance
Top-25% w.r.t. `Init#1`. {'substance', 'rather', 'at', 'yarn', 'coloring', 'movie'}
Top-25% w.r.t. `Init#2`. {'heart', '##tly', 'suspense', 'at', 'yarn', 'def'}
Jaccard@25%. 20

Example 2: SST2 data. Comparing `Init#1` and `Init#2`. Both models predict the sentiment to be positive.
Text. an infectious cultural fable with a tasty balance of family drama and frenetic comedy
Top-25% w.r.t. `Init#1`. {'fable', 'infectious', 'cultural', 'balance', 'an'}
Top-25% w.r.t. `Init#2`. {'cultural', 'balance', 'infectious', 'fable', 'an'}
Jaccard@25%. 100

Example 3: FPB data. Comparing `Init#1` and `Untrained`. Both models predict the sentiment to be negative.

Text. nokia shares hit 13.21 euros on friday , down 50 percent from the start of the year in part because of the slow introduction of touch-screen models
Top-25% w.r.t. `Init#1`. { ‘,’, ‘.’, ‘down’, ‘friday’, ‘shares’, ‘euros’, ‘nokia’, ‘hit’ }
Top-25% w.r.t. `Init#2`. { ‘,’, ‘.’, ‘down’, ‘euros’, ‘friday’, ‘hit’, ‘shares’, ‘nokia’ }
Jaccard@25%. 100

In second and third examples, even though the rankings are different, the set of top-25% tokens is the same leading to a perfect Jaccard@25%.