Efficient Explanations from Empirical Explainers

Robert Schwarzenberg[1] **Nils Feldhus**[1] **Sebastian Möller**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI)
[2]Technische Universität Berlin (TU Berlin)
{firstname.lastname}@dfki.de

Abstract

Amid a discussion about Green AI in which we see explainability neglected, we explore the possibility to efficiently approximate computationally expensive explainers. To this end, we propose the task of feature attribution modelling that we address with Empirical Explainers. Empirical Explainers learn from data to predict the attribution maps of expensive explainers. We train and test Empirical Explainers in the language domain and find that they model their expensive counterparts well, at a fraction of the cost. They could thus mitigate the computational burden of neural explanations significantly, in applications that tolerate an approximation error.

1 Introduction

In recent years, important works were published on the ecological impacts of artificial intelligence and deep learning in particular, e.g. Strubell et al. (2019), Schwartz et al. (2020), Henderson et al. (2020). Research is focused on the energy hunger of model training and subsequent inference in production. Besides training and in-production inference, explainability has become an integral phase of many neural systems.

In the ongoing discussion about Green AI we see explainability neglected. Conversely, in the explainability community, even though research on efficiency is an active area, apparently the discussion is currently shaped by other aspects, such as faithfulness and plausibility (Jacovi and Goldberg, 2020). This is surprising because to explain a single model output, many prominent explanation methods, in particular many feature attribution methods (cf. below), require a multiple of computing power when compared to the prediction step.

1.1 Motivation: Expensive Explainers

Take, for instance, the demonstrative but arguably realistic case of a classifier that was trained on $100k$ instances for 10 epochs. The training thus amounts to at least $1M$ forward passes and $1M$ backward passes. To produce explanations, in this paper, we consider attribution methods and focus on Integrated Gradients (IG) (Sundararajan et al., 2017) and Shapley Values (SV) (Castro et al., 2009), which are popular and established but also computationally expensive. To compute the exact IG or SV is virtually intractable, which is why sampling-based approximations were devised. For IG

$$\phi_{f,i}(x) = \frac{x_i - \bar{x}_i}{s} \sum_{k=1}^{s} \frac{\partial f(\bar{x} + \frac{k}{s}(x - \bar{x}))}{\partial x_i}$$

is computed, where x is the input to model f, $\bar{x}$ is a user-defined baseline, s denotes the number of samples (a hyperparameter), and $\phi_{f,i}(x)$ denotes the attribution score of feature i. For SV, s permutations of the input data $O_1, O_2, \ldots O_s$ are drawn and then features from x are added to a user-defined baseline,[1] in the order they occur in the permutation. Let $\text{Pre}^i(O)$ denote the baseline including the features that were added to the baseline prior to i. The Shapley value can then be approximated by

$$\phi_{f,i}(x) = \frac{1}{s} \sum_{k=1}^{s} f(\text{Pre}^i(O_k) \cup x_i) - f(\text{Pre}^i(O_k))$$

Sundararajan et al. (2017) report that s between 20 and 300 is usually enough to approximate IG. Let us set $s := 20$. This requires 40 passes (forward and backward) through model f to explain a

[1]There are several variants. This sampling method is based on PyTorch's Captum (Kokhlikyan et al., 2020) library, that we also use for our experiments: https://captum.ai/api/shapley_value_sampling.html, last accessed March 26, 2021.

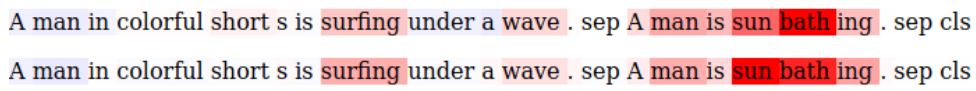

Figure 1: Explanations for an XLNet-based NLI classification. The input is taken from the test split and was classified into `Contradiction`. **Top**: Shapley Value Samples ($s = 20$, 380 passes required). **Bottom**: Empirical Shapley Values (1 pass required). Attribution scores were normalized on sequence level.

single instance in production and furthermore, after only 50k explanations the computational costs of training are also already surpassed. In the case of SV, again setting $s := 20$ and assuming only 512 input features (i.e. tokens to an NLP model), one already needs to conduct $20 * 512 = 10240$ passes to generate an input attribution map for a single classification decision. This means that SV surpasses the training costs specified above after only 195 explanations.

This may only have a small impact if the number of required explanations is low. However, there are strong indications that explainability will take (or retain) an important role in many neural systems: For example, there are legal regulations, such as the EU's GDPR which hints at a "right to explanation" (Goodman and Flaxman, 2017). For such cases, a 1:1 ratio in production between model outputs and explanations is realistic. If the employed explainability method requires more than one additional pass through the model (as many do, cf. below), there then is a tipping point at which the energy need of explanations exceeds the energy needs of both model training and in-production inference.

IG and SV are not the only tipping point methods. Other expensive prominent and recent methods and variants are proposed by Zeiler and Fergus (2014); Ribeiro et al. (2016); Lundberg and Lee (2017); Smilkov et al. (2017); Chen et al. (2019); Dhamdhere et al. (2020); Erion et al. (2019); Covert and Lee (2020); Schwarzenberg and Castle (2020); Schulz et al. (2020); Harbecke and Alt (2020).

All of the above listed explainers require more than one additional pass through the model. This is why in general the following should hold across methods: *The smaller the model, the greener the explanation.* In terms of energy efficiency, explainability therefore benefits from model compression, distillation, or quantization. These are dynamic fields with a lot of active research which is why in the remainder of this paper we instead focus on something else: The mitigation of the ecological impact of tipping-point methods that dominate the cost term in the example cited in this section.

These are our contributions in this paper:

1. We propose feature attribution modelling, a novel task that we solve with trainable models, which we coin Empirical Explainers.
2. We evaluate their performance qualitatively and quantitatively in the language domain and establish them as an efficient alternative to computationally expensive explainers.

2 Method: Empirical Explainers

Informally, an EMPIRICAL EXPLAINER *is a model that has learned from data to efficiently model the feature attribution maps of an expensive explainer. For training, one collects sufficiently many attribution maps from the expensive explainer and then maximizes the likelihood of these target attributions under the Empirical Explainer.*

An expensive explainer may, for instance, be a costly attribution method such as Integrated Gradients that is used to return attributions for the decisions of a classifier, say, a BERT-based (Devlin et al., 2019) sentiment classifier. The corresponding Empirical Explainer could be a separate neural network, similar in size to the sentiment classifier, consuming the same input tokens as the sentiment model, but instead of predicting the sentiment class, it is trained to predict the integrated gradients for each input token.

Whereas the original Integrated Gradients explainer requires multiple passes through the classifier, the empirical integrated gradients require just one pass through the efficient explainer. Empirical explanations come with an accuracy-efficiency trade-off that we discuss in the course of a more formal definition of Empirical Explainers.

For the more formal definition, we need to fix notation first. Let $E_f : \mathbb{R}^d \to \mathbb{R}^d$ be the expensive explainer that maps inputs onto attributions. Furthermore, let an Empirical Explainer be a function $e_\theta : \mathbb{R}^d \to \mathbb{R}^d$, parametrized by θ, which also returns attribution maps. Let $||\cdot||$ be a penalty for the inefficiency of a computation, e.g. a count of floating point operations, energy consumption or number of model passes needed. Furthermore,

let us assume, without the loss of generality, that $||E_f(x)|| >= ||e_\theta(x)||$ always holds; i.e., the Empirical Explainer – which we develop and train – is never more inefficient than the original, expensive explainer. Let $D : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$ be a similarity measure, where $D(l, m) = 0$ if $l = m$, for $l, m \in \mathbb{R}^d$ and $\alpha, \beta \in [0, 1]$ with $\alpha + \beta = 1$. For data $\mathbf{X}$, we define a β-optimal Empirical Explainer by the $\arg\min_{\theta \in \Theta}$

$$\frac{1}{|X|} \sum_{x \in \mathbf{X}} \overbrace{\alpha D(E_f(x), e_\theta(x))}^{\text{accuracy}} + \overbrace{\beta \left(\frac{||e_\theta(x)||}{||E_f(x)||} \right)}^{\text{efficiency}}. \tag{1}$$

2.1 Properties

The first term describes how accurately the Empirical Explainer e_θ models the expensive explainer E_f. The second term compares the efficiency of the two explainers. For $\alpha = 1$, efficiency is considered unimportant and $e_\theta := E_f$ can be set to minimize Eq. 1. $\alpha < 1$ allows to optimize efficiency at the cost of accuracy, which brings about the trade off: One may not succeed in increasing efficiency while maintaining accuracy. In fact, there is generally no exact guarantee for how accurately e_θ models E_f for new data.

Furthermore, while several expensive explainers, such as Integrated Gradients or Shapley Values, were developed axiomatically to have desirable properties, Empirical Explainers are derived from data – empirically. Consequently, the evidence and guarantees Empirical Explainers offer for their faithfulness to the downstream model are empirical in nature and upper-bound by the faithfulness of the expensive explainer used to train them.

We point this out explicitly because we would like to emphasize that we do *not* regard an Empirical Explainer a new explainability method, nor do we argue that it can be used to replace the original expensive explainer everywhere. There are certainly situations for which Empirical Explainers are unsuitable for any $\alpha \neq 1$; critical cases in which explanations must have guaranteed properties.

Nevertheless, we still see a huge potential for Empirical Explainers where approximation errors are tolerable: Consider, for instance, a search engine powered by a neural model in the back-end. Without the need to employ the expensive explainer, Empirical Explainers can efficiently provide the user with clues about what the model probably considers relevant in their query (according to the expensive explainer).

3 Experiments

In this section, we report on the performance of Empirical Explainers that we trained and tested in the language domain.[2] We conducted tests with two state-of-the-art language classifiers, trained on two tasks, explained with two expensive explainers.

The experiments address the question of whether or not it is feasible – in principle – to train efficient Empirical Explainers while achieving significant accuracy. All experiments, code, models and data are open source and can be retrieved following `https://github.com/dfki-nlp/emp-exp`. The most important choices are documented in the following paragraphs. Before going into greater detail, it is noteworthy that Eq. 1 provides a theoretical framework which does not have to be used directly for optimization and does not prescribe a specific model.

We trained two Empirical Explainers: The first one (EmpExp-IMDB) was trained to predict integrated gradients w.r.t. the input tokens to a BERT-based IMDB movie review (Maas et al., 2011) classifier. For the second Empirical Explainer (EmpExp-SNLI), we varied the downstream model architecture, task and expensive explainer: EmpExp-SNLI predicts the Shapley Values for the inputs of an XLNet-based (Yang et al., 2019) natural language inference classifier that was trained on the SNLI (Bowman et al., 2015) dataset. Both Empirical Explainers consume only the input tokens to the downstream model and return an attribution score for each token.

The two Empirical Explainers were trained on target attributions that we generated with IG and SV with $s := 20$ samples. Explanations were generated for the output neuron with the maximal activation. EmpExp-IMDB was trained with early stopping using the IG attribution maps for the IMDB train split. EmpExp-SNLI was trained with around 100k SV attribution maps for the SNLI train split with early stopping, for which we used the 10k attribution maps for the validation split. We did not use all training instances in the split for EmpExp-SNLI, due to the computational costs of Shapley

[2] The authors of this paper primarily research NLP models, which is why Empirical Explainers are first tested in the language domain.

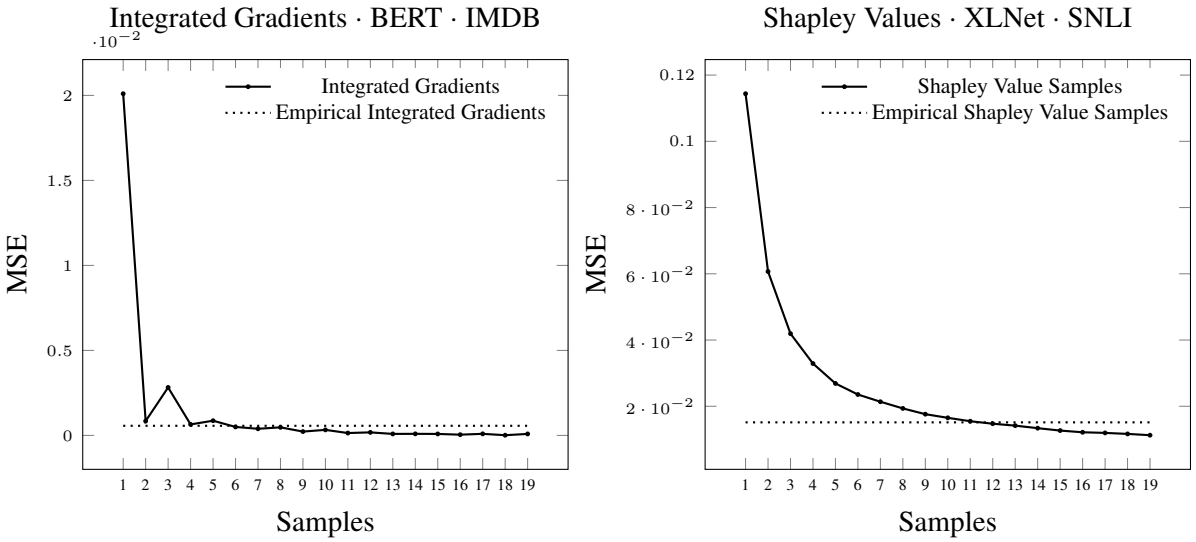

Figure 2: Performance of Empirical Explainers (dashed lines) and convergence curves of expensive explainers (solid lines), averaged across test sets. The attribution maps returned by the expensive explainers with $s = 20$ samples were regarded the target explanations. MSEs were computed on a per-sequence basis and then averaged across the test set. **Left**: Explanations for the decision of a BERT-based sentiment classifier were compared, using the inputs from the IMDB test split. **Right**: Explanations for the decisions of an XLNet-based NLI classifier were compared, using the inputs from the SNLI test split.

Value Sampling.

As mentioned above, the expensive explainers require user-defined baselines. For the baselines we replaced all non-special tokens in the input sequence with pad tokens. For the expensive IG, we produced attribution maps for the embedding layer and projected the attribution scores onto tokens by summing them over the token dimension. For the expensive SV, the input ids were perturbed. During perturbation, we grouped and treated special tokens (CLS, SEP, PAD, ...) in the original input as one feature to accelerate the computation.

In architectural terms, the Empirical Explainers are very similar to the downstream models: We heuristically decided to copy the fine-tuned BERT and XLNet encoders from the text classifiers and instead of the classification layers on top, we initialized new fully connected layers with T output neurons. T was lower bound by the maximum input token sequence length to the downstream model in the respective dataset: $T = 512$ for BERT/IMDB and $T = 130$ for XLNet/SNLI. All input sequences were padded to T and we did not treat padding tokens different from other tokens, when training the Empirical Explainers.

We trained the Empirical Explainers to output the right (in accordance with the expensive explainers) attribution scores for the input tokens, using an MSE loss between $E_f(x_1) \dots E_f(x_T)$ and $e_\theta(x_1) \dots e_\theta(x_T)$ where $x = x_1, x_2, \dots x_T$ is a sequence of input tokens.[3]

To put the performance of an Empirical Explainer into perspective, we propose the following baseline, which is the strongest we can think of: We take the original expensive explainer with a reduced number of samples ($s < 20$) as the baseline. To position the Empirical Explainer against this alternative energy saving strategy, we compute convergence curves. Starting with $s = 1$, we incrementally increase the number of samples until $s = 19$ and collect attribution maps from the expensive explainer for the different choices of s. We then compute the MSEs of these attribution maps when compared to the target attributions with $s = 20$. We average the MSEs across the test split. The same is done for the Empirical Explainer.

4 Results & Discussion

In the following, we report the experimental results, divided into the aspects of task performance, explanation efficiency and explanation accuracy.

Task Performance On the test splits, the classifiers we trained achieved weighted F_1 scores of .93

[3]Even though we do not solve for Eq. 1 directly, please note that for evaluation we can normalize the attribution scores to $[0, 1]$ prior to computing the MSE and this way force the MSE into the interval $[0, 1]$ to comply with the constraints for Eq. 1.

(BERT/IMDB) and .90 (XLNet/SNLI).

Explanation Efficiency Regarding the efficiency term in Eq. 1, in terms of model passes, the Empirical Explainers have a clear advantage over their expensive counterparts. For IG with $s = 20$, 40 model passes are required. For SV with $s = 20$, assuming a token sequence length of 100 for the purpose of discussion, 2000 model passes are required. For the empirical explanations, only one (additional) forward pass through a similarly sized model (the Empirical Explainer) is necessary.

Contrary to runtime (and energy consumption) measures, the number of required model passes is largely invariant of available hardware and implementation details. For the sake of completeness, we nevertheless also report our run times: Generating the expensive target explanations for the official IMDB test split (25k longer instances) with Integrated Gradients and $s = 20$ samples, took us 7:17 hours on a NVIDIA Titan V GPU with around 12 GB of VRAM available (06:22 hours for the 22500 training instances). Generating the expensive Shapley Values for the SNLI test split ($\sim$ 10k shorter instances), again with $s = 20$ samples, took us 48:22 hours on an NVIDIA QUADRO GPU w/ around 16 GB of VRAM (over 600 GPU hours for under 100k training instances). In contrast, generating the empirical explanations took us only 07:14 *minutes* for the IMDB test split on the Titan GPU and only 02:05 *minutes* for SNLI test split on the Quadro GPU.

The runtimes above are not definitive, however. We were unable to establish a fair game for the explainers. For example, due to implementation details and memory issues we explained the data instance-wise with the expensive explainers while our Empirical Explainers easily allowed batch processing. We expect that the expensive explainers can be accelerated but due to the larger number of model passes required, they will very likely not outperform their empirical counterparts.

Explanation Accuracy Regarding the accuracy term in Eq. 1, Fig.1 depicts an example of a target attribution map and its approximated empirical counterpart. Alongside this paper, we provide two files with around 25k (IMDB) and 10k (SNLI) lines, each of which contains an HTML document that depicts a target attribution and its empirical counterpart from the test set. An initial qualitative inspection suggests that the Empirical Explainers model their expensive counterparts well, with varying degrees of approximation errors across the instances, however.

A more objective and quantitative analysis is provided in Fig. 2. The left side depicts the MSE line of IG for an increasing number of samples. We observe that IG converges fast. (This may be due to saturation effects in Integrated Gradients, reported on by Miglani et al. (2020).) We also observe that the empirical integrated gradients (dashed line) perform favourably: To outperform the Empirical Explainer by decreasing s, one needs to choose $s > 5$ which entails 10 model passes as opposed to the single additional pass for the empirical explanations. Furthermore, the approximation error is already marginal at the intersection of expensive and empirical line.

A similar trend can be observed for the (empirical) Shapley Values on the right side of Fig. 2. This time, however, the intersection happens only after $s = 10$ which means that the Empirical Explainer needs only $\frac{1}{11*100} = 0.9\%$ of the computational cost of the next best expensive explainer, again assuming 100 input tokens for the purpose of discussion. We take this as a strong indication that Empirical Explainers can be considered an efficient alternative to expensive explainers, at least in the language domain.

5 Related Work

The computational burden of individual explainability methods was addressed in numerous works. As mentioned above, Integrated Gradients can only be computed exactly in limit cases and for all other cases, the community relies on the approximate method proposed by Sundararajan et al. (2017). Similarly, Shapley Values can rarely be computed precisely which is why Shapley Value Sampling was investigated, e.g. by Castro et al. (2009); Štrumbelj and Kononenko (2010). Shapley Value Sampling was later unified with other methods under the SHAP framework (Lundberg and Lee, 2017) which yielded the method KernelSHAP that showed improved sample efficiency. Covert and Lee (2020) then analysed the convergence behaviour of KernelSHAP and again further improved runtime. Chen et al. (2019) introduced L-Shapley and C-Shapley which accelerate Shapley Value Sampling for structured data, such as dependency trees in NLP. Thus, computational feasibility appears to be a driving force in the research community, already. To the best of our knowledge,

however, we are the first to propose the task of feature attribution modelling and the class of Empirical Explainers that may be used to model the above mentioned explainers efficiently.

Technically, we took inspiration from models that efficiently generate an explanation alongside their classification decision, as proposed by Camburu et al. (2018). The authors trained their model to return a natural language rationale which fundamentally differs from the target explanations Empirical Explainers are trained with, however. Another source of inspiration is the technique of gradient matching, for which a network's (integrated) gradients are compared to a target attribution, i.e. a human prior, and then the network's parameters are updated, s.t. the gradients move closer to the target, as done e.g. by Ross et al. (2017); Erion et al. (2019); Liu and Avci (2019). Apart from the loss on an alignment with target attributions, our method and goals diverge from theirs significantly. Very recently, Rajagopal et al. (2021) proposed local interpretable layers as a means to generate concept attributions which in parts aligns with our method, even though their target attributions and task objectives are very different again.

6 Conclusion & Future Work

In this paper, we take a step towards greener XAI by proposing energy efficiency as an additional criterion by which to judge an explainability method, alongside important aspects such as faithfulness and plausibility. In this context, we propose the novel task of feature attribution modelling that we address with efficient Empirical Explainers. In the language domain, we were able to generate empirical attributions with significant accuracy. We take this as a strong indication that Empirical Explainers could be a viable alternative to expensive explainers, where approximation errors are tolerable.

The Empirical Explainers we trained are our concrete model choices. The framework we propose allows many other approaches, too. We would like to note that we tested our Empirical Explainers only on in-domain data but their behaviour on out-of-domain data should be investigated, too. Lastly, as noted in Sec. 5, there are more sample efficient alternatives to the expensive Shapley Value Sampling explainer that we stressed tested our Empirical Explainers with. These more efficient versions should be considered in the future, too.

Acknowledgements

We would like to thank Christoph Alt, Steffen Castle, Jonas Mikkelsen and David Harbecke for their helpful feedback. This work has been supported by the German Federal Ministry of Education and Research as part of the project XAINES (01IW20005).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31:9539–9549.

Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2019. L-shapley and c-shapley: Efficient model interpretation for structured data. *International Conference on Learning Representations 2019*.

Ian Covert and Su-In Lee. 2020. Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv preprint arXiv:2012.01536*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Kedar Dhamdhere, Ashish Agarwal, and Mukund Sundararajan. 2020. The shapley taylor interaction index. In *ICML 2020*.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2019. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*.

Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

David Harbecke and Christoph Alt. 2020. Considering likelihood in nlp classification explanations with occlusion and language modeling. *ACL 2020 SRW*.

P. Henderson, Jie-Ru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *ArXiv*, abs/2002.05651.

Alon Jacovi and Y. Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *ACL 2020*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6274–6283.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Andrew L. Maas, Raymond E. Daly, P. T. Pham, D. Huang, A. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL 2011*.

Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. 2020. Investigating saturation effects in integrated gradients. *arXiv preprint arXiv:2010.12697*.

Dheeraj Rajagopal, Vidhisha Balachandran, E. Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. *arXiv preprint arXiv:2103.12279*.

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*, pages 1135–1144.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*.

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. 2020. Restricting the flow: Information bottlenecks for attribution. *ICLR 2020*.

Roy Schwartz, Jesse Dodge, N. A. Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*, 63:54 – 63.

Robert Schwarzenberg and Steffen Castle. 2020. Pattern-guided integrated gradients. *ICML 2020 Workshop on Human Interpretability in Machine Learning (WHI) arXiv preprint arXiv:2007.10685*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. pages 3645–3650.

E. Štrumbelj and I. Kononenko. 2010. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328.

Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *13th European Conference on Computer Vision, ECCV 2014*, pages 818–833. Springer Verlag.