An Empirical Comparison of Instance Attribution Methods for NLP

Pouya Pezeshkpour*
University of California, Irvine
pezeshkp@uci.edu

Sarthak Jain*
Northeastern University
jain.sar@northeastern.edu

Byron C. Wallace
Northeastern University
b.wallace@northeastern.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Abstract

Widespread adoption of deep models has motivated a pressing need for approaches to interpret network outputs and to facilitate model debugging. *Instance attribution* methods constitute one means of accomplishing these goals by retrieving training instances that (may have) led to a particular prediction. Influence functions (IF; Koh and Liang 2017) provide machinery for doing this by quantifying the effect that perturbing individual train instances would have on a specific test prediction. However, even approximating the IF is computationally expensive, to the degree that may be prohibitive in many cases. Might simpler approaches (e.g., retrieving train examples most similar to a given test point) perform comparably? In this work, we evaluate the degree to which different potential instance attribution agree with respect to the importance of training samples. We find that simple retrieval methods yield training instances that differ from those identified via gradient-based methods (such as IFs), but that nonetheless exhibit desirable characteristics similar to more complex attribution methods. Code for all methods and experiments in this paper is available at: https://github.com/successar/instance_attributions_NLP.

1 Introduction

Interpretability methods are intended to help users understand model predictions (Ribeiro et al., 2016; Lundberg and Lee, 2017; Sundararajan et al., 2017; Gilpin et al., 2018). In machine learning broadly and NLP specifically, such methods have focused on feature-based explanations that highlight parts of inputs 'responsible for' the specific prediction. Feature attribution, however, does not communicate a key basis for model outputs: training data. Recent work has therefore considered methods for surfacing training examples that were influential for a specific prediction (Koh and Liang, 2017; Yeh et al., 2018; Pezeshkpour et al., 2019; Charpiat et al., 2019; Barshan et al., 2020; Han et al., 2020). While such *instance-attribution* methods provide an appealing mechanism to identify sources that led to specific predictions (which may reveal potentially problematic training examples), they have not yet been widely adopted, at least in part because even approximating influence functions (Koh and Liang, 2017)—arguably the most principled attribution method—can be prohibitively expensive in terms of compute. Is such complexity necessary to identify 'important' training points? Or do simpler methods (e.g., attribution scores based on similarity measures between train and test instances) yield comparable results? In this paper, we set out to evaluate and compare instance attribution methods, including relatively simple and efficient approaches (Rajani et al., 2020) in the context of NLP (Figure 1). We design qualitative evaluations intended to probe the following research questions: (1) How correlated are rankings induced by gradient and similarity-based attribution methods (assessing the quality of more efficient approx-

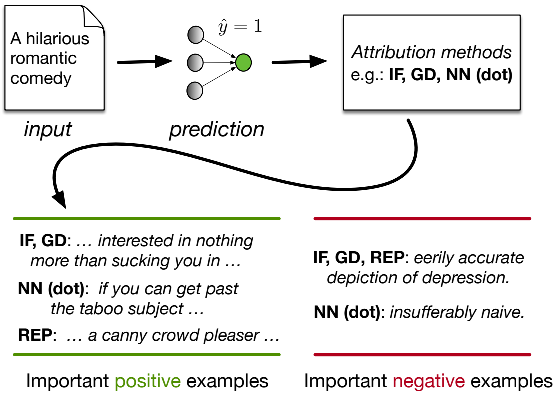

A hilarious romantic comedy
input
$\hat{y} = 1$
prediction
Attribution methods e.g.: IF, GD, NN (dot)
IF, GD: ... interested in nothing more than sucking you in ...
NN (dot): if you can get past the taboo subject ...
REP: ... a canny crowd pleaser ...
IF, GD, REP: eerily accurate depiction of depression.
NN (dot): insufferably naive.
Important positive examples
Important negative examples

Figure 1: Attribution methods score train examples in terms of their importance to a particular prediction. In this work, we compare several such methods, e.g., Influence Functions (IF) and its variants (GD), Representer Points (REP) and similarity measures (NN).

[*] Equal contribution

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 967–975
June 6–11, 2021. ©2021 Association for Computational Linguistics

imations)? (2) What is the quality of explanations in similarity methods compared to gradient-based ones (clarifying the necessity of adopting more complex methods)?

We evaluate instance-based attribution methods on two datasets: binarized version of the Stanford Sentiment Treebank (SST-2; Socher et al. 2013) and the Multi-Genre NLI (MNLI) dataset (Williams et al., 2018). We investigate the correlation of more complex attribution methods with simpler approximations and variants (with and without use of the Hessian). Comparing explanation quality of gradient-based methods against simple similarity retrieval using leave-one-out (Basu et al., 2020) and randomized-test (Hanawa et al., 2021) analyses, we show that simpler methods are fairly competitive. Finally, using the HANS dataset (McCoy et al., 2019), we show the ability of similarity-based methods to surface artifacts in training data.

2 Attribution Methods

Similarity Based Attribution Consider a text classification task in which we aim to map inputs x_i to labels $y_i \in Y$. We will denote learned representations of x_i by f_i (i.e., the representation from the penultimate network layer). To quantify the importance of training point x_i on the prediction for target sample x_t, we calculate the similarity in embedding space induced by the model.[1] To measure similarity we consider three measures: *Euclidean* distance, *Dot* product, and *Cosine* similarity. Specifically, we define similarity-based attribution scores as: **NN EUC** = $-\|f_t - f_i\|^2$, **NN COS** = $\cos(f_t, f_i)$, and **NN DOT** = $\langle f_t, f_i\rangle$.

To investigate the effect of fine-tuning on these similarity measures, we also derive rankings based on similarities between untuned sentence-BERT (Reimers et al., 2019) representations.

Gradient Based Attribution *Influence Functions (IFs)* were proposed in the context of neural models by Koh and Liang (2017) to quantify the contribution made by individual training points on specific test predictions. Denoting model parameter estimates by $\hat{\theta}$, the IF approximates the effect that upweighting instance i by a small amount—ϵ_i—would have on the parameter estimates (here H is the Hessian of the loss function with respect to our parameters): $\frac{d\hat{\theta}}{d\epsilon_i} = -H_{\hat{\theta}}^{-1}\nabla_\theta \mathcal{L}(x_i, y_i, \hat{\theta})$. This estimate can in turn be used to derive the effect on a specific test point x_{test}: $\nabla_\theta \mathcal{L}(x_{\text{test}}, y_{\text{test}}, \hat{\theta})^T \cdot \frac{d\hat{\theta}}{d\epsilon_i}$.

Aside from IFs, we consider three other similar gradient-based variations:

(1) RIF = $\cos(H^{-\frac{1}{2}}\nabla_\theta \mathcal{L}(x_{\text{test}}), H^{-\frac{1}{2}}\nabla_\theta \mathcal{L}(x_i))$.
(2) GD = $\langle \nabla_\theta \mathcal{L}(x_{\text{test}}), \nabla_\theta \mathcal{L}(x_i)\rangle$, and
(3) GC = $\cos(\nabla_\theta \mathcal{L}(x_{\text{test}}), \nabla_\theta \mathcal{L}(x_i))$.

RIF was proposed by Barshan et al. (2020), while GD and GC by Charpiat et al. (2019).

Representer Points (REP; Yeh et al. 2018) introduced to approximate the influence of training points on a test sample by defining a classifier as a combination of a feature extractor and a ($L2$ regularized) linear layer: $\phi(x_i, \theta)$. Yeh et al. (2018) showed that for such models the output for any target instance x_t can be expressed as a linear decomposition of "data importance" of training instances: $\phi(x_t, \theta^*) = \sum_i^n \alpha_i f_i^\top f_t = \sum_i^n k(x_t, x_i, \alpha_i)$, where $\alpha_i = \frac{1}{-2\lambda_n}\frac{\partial \mathcal{L}(x_i, y_i, \theta)}{\partial \phi(x_i, \theta)}$.

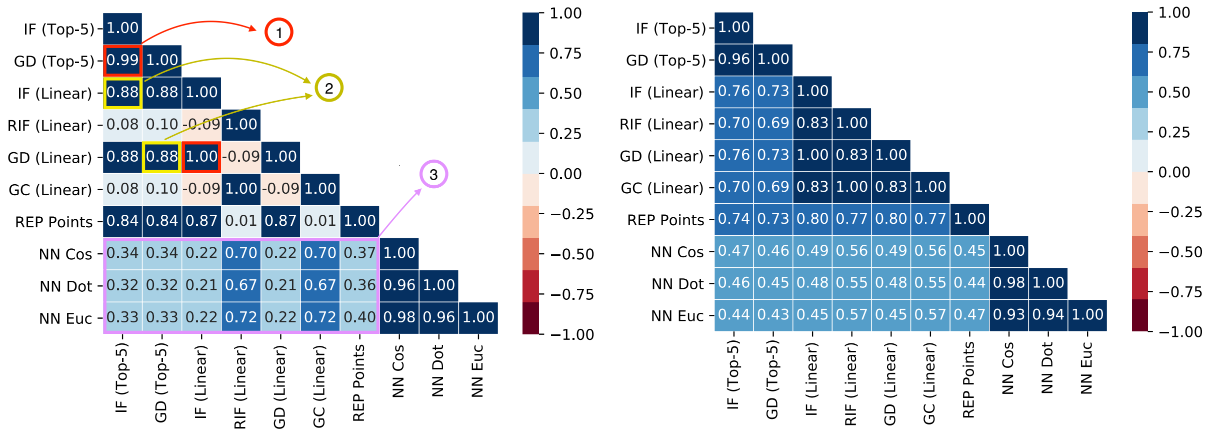
3 Experimental Setup

Datasets To evaluate different attribution methods, we conduct several experiments on sentiment analysis and NLI tasks, following prior work investigating the use of IF specifically for NLP (Han et al., 2020). We adopt a binarized version of the Stanford Sentiment Treebank (SST-2; Socher et al. 2013), and the Multi-Genre NLI (MNLI) dataset (Williams et al., 2018). For fine-tuning on MNLI, we randomly sample 10k training instances. Finally, to evaluate the ability of instance attribution methods to reveal annotation *artifacts* in NLI, we randomly sampled 1000 instances from the HANS dataset (more details in the Appendix).

Models We define models for both tasks on top of BERT (Devlin et al., 2019), tuning hyperparameters on validation data via grid search. Our models achieve 90.6% accuracy on SST and 71.2% accuracy on MNLI (more details in the Appendix).

Computing the IF for BERT Deriving the IF for all parameters θ of a BERT-based model requires deriving the corresponding Inverse Hessian. We compute the Inverse Hessian Vector Product (IHVP) $H^{-1}\nabla_\theta \mathcal{L}(x, y, \theta)$ directly because storing the entire matrix of $|\theta|^2$ elements is practically impossible (requiring $\sim$12 PB of storage). We approximate the IHVP using the LiSSa algorithm

[1] To be clear, there is no guarantee that similarity reflects 'influence' at all, but we are interested in the degree to which this simple strategy identifies 'useful' training points, and whether the ranking implied by this method over train points agrees with rankings according to more complex methods.



(a) Spearman Correlation on SST.

(b) Spearman Correlation on MNLI

Figure 2: The similarity between influence of training samples for different pairs of attribution methods on the SST and MNLI datasets was measured via Spearman Correlation. ① = Using Hessian does not change the ordering of training examples. ② = Using more layers of BERT in IF approximation does not much affect the ordering. ③ = NN metrics are not well correlated with gradient-based ones.

(Agarwal et al., 2017). This method is still expensive to run and is sensitive to the norm of the IHVP approximation. Therefore, for computational reasons we consider IF with respect to the subset of parameters that correspond to the top five layers [IF (Top-5)], and only the last linear layer [IF (linear)], resulting in a few orders of magnitude faster procedure (the algorithm becomes increasingly unstable as we incorporate additional layers). We also use a large scaling factor to aid convergence.

4 Experiments

In this section, we first investigate the correlation between different methods. Then, to study the quality of explanations we conduct leave-some-out experiments, and further analyze attribution methods on HANS data. We consider four evaluations (more analyses and experimental details in the Appendix).

(1) Calculating the *correlation* of each pair of attribution methods, assessing whether simple methods induce rankings similar to more complex ones.

(2) *Removing the most influential samples* according to each method, retrain, and then observe the change in the predicted probability for the originally predicted class, with the assumption that more accurate attribution methods will cause more drop.

(3) We follow *randomized-test* from (Hanawa et al., 2021) and measure the ranking correlation of methods for (a) randomly initialized and (b) trained models, under the assumption that high correlation here would suggest less meaningful attribution.

(4) We measure the degree to which the methods recover examples that exhibit *lexical overlap* when tested on the HANS dataset (McCoy et al., 2019). This extends a prior analysis of IF (Han et al., 2020), considering alternative attribution methods.

Attribution Methods' Correlation We calculate the Spearman correlation between scores assigned to training samples by different methods, allowing us to compare their similarities. More specifically, we randomly sample 100 test and 500 training samples from datasets and calculate the average resultant Spearman correlations.

We report attribution methods' correlation on SST and MNLI datasets in Figure 2 (a more complete version of these figures is in the Appendix). We make the following observations. (1) Gradient methods w/wo normalization appear similar to each other, e.g., GC is similar to RIF and IF is similar to GD, suggesting that Hessian information may not be necessary to provide meaningful attributions (GD and GC do not use the Hessian). (2) There is a high correlation between IF calculated over the top five layers of BERT and IF over only the last linear layer. (3) There is only a modest correlation between similarity-based rankings and gradient-based methods, suggesting that these do differ in terms of the importance they assign to training instances. We report a proportion of common top examples between IF (Top-5) and IF (Linear) in the Appendix, providing further evidence of the high correlation between these methods.

Removing 'Important' Samples In Table 1 we report the average results of removing the top-k

	Method	avg(Δ)-SST		avg(Δ)-MNLI		Spearman	
		Remove-50	Remove-500	Remove-50	Remove-500	SST	MNLI
	Random (50 runs)	-0.028	-0.021	-0.039	-0.029	-	-
Similarity	NN EUC	-0.028	**-0.540**	-0.102	-0.266	0.056	0.023
	NN COS	-0.072	-0.430	-0.088	-0.306	0.045	0.018
	NN DOT	-0.059	-0.513	**-0.106**	-0.273	0.005	-0.002
Gradient	IF	-0.054	-0.526	-0.042	-0.407	-0.296	0.018
	REP	**-0.114**	-0.490	-0.002	-0.230	-0.217	0.053
	RIF	-0.071	-0.537	-0.068	-0.347	-0.021	0.013
	GD	-0.058	-0.516	-0.022	**-0.446**	-0.290	0.017
	GC	-0.082	-0.528	-0.030	-0.279	-0.021	0.012

Table 1: Average difference (Δ) between predictions made after training on (i) all data and (ii) a subset in which we remove the top-50/top-500 most important training points, according to different methods (Random on both of the benchmarks has standard deviation around 0.02). We also report the Spearman correlation between the ranking induced by each approach using a trained model and the same ranking when a randomly initialized model is used.

	Method	Lexical Overlap Rate	
		top-1	top-10
	Random	0.40	0.40
Sen-Bert	NN EUC	0.39	0.41
	NN COS	0.38	0.39
	NN DOT	0.39	0.40
Sim	NN EUC	**0.56**	**0.57**
	NN COS	**0.56**	0.56
	NN DOT	0.44	0.44
Gradient	IF	0.43	0.44
	REP	0.43	0.35
	RIF	0.55	0.56
	GD	0.43	0.44
	GC	0.55	0.56

Table 2: Average lexical overlap rate between premise and hypothesis in top-k most influential samples for test instances mispredicted as entailment.

most important training samples for 50 random test samples using different attribution methods. We only consider the linear version of methods in the remainder of the paper. All methods seem effective, compared to random sampling. Perhaps surprisingly, for both tasks at least one of the similarity-based approaches performs comparably or better than gradient-based methods, in the sense that removing the top examples according to similarity yields reductions in the predicted probability (which is what one would intuitively hope). Finally, it seems that the models applying some form of normalization to the gradient (i.e., RIF and GC) perform more consistently. This is consistent with contemporaneous work of Hanawa et al. (2021) which argues that this is a consequence of large gradient magnitudes for some samples dominating when normalization is not used. Upon investigating high influential training samples, we observed that similarity-based approaches seem to yield more diverse "top" instances compared to gradient-based ones. We also found that normalization in gradient-based methods made a large difference. Generic IF-based ranking tends to be dominated by high loss training examples across test examples, whereas normalization provides more diverse top training examples. Further, proportions of shared top examples between methods is provided in the Appendix, clarifying their similar performance.

Randomized-Test We report the Spearman correlation between trained and random models for SST and MNLI data in Table 1. This would ideally be small in magnitude (non-zero values indicate correlation). Curiously, gradient-based methods (IF, REP, GD) exhibit negative correlations on the SST dataset. Overall, these results suggest that gradient-based approaches without gradient normalization may be inferior to alternative methods. The simple NN-DOT method provides the 'best' performance according to this metric.

Artifacts and Attribution Methods To investigate whether attribution methods can correctly identify training samples with specific artifacts responsible for model predictions we follow Han et al. (2020): This entails randomly choosing 10k samples from MNLI and treating *neutral* and *contradiction* as a single *non-entailment* label for model fine-tuning. More specifically, we are interested in target samples that the model mispredicts as *entailment* because of the lexical overlap artifact (lexical overlap is an artifactual indicator of entailment; McCoy et al. 2019).

The average lexical overlap rate for 1000 random samples from the HANS dataset is provided in Table 2. As a baseline, we also apply similarity-based methods on top of sentence-BERT embeddings, which as expected appear very similar to random correlation. One can observe that similarity-based approaches tend to surface instances with higher lexical overlap, compared to gradient-based instance attribution methods. Moreover, gradient-based methods without normalization (IF, GD, and REP) perform similar to selecting samples randomly and based on sentence-BERT representations, suggesting an inability to usefully identify lexical overlap.

Computational Complexity The computational complexity of IF-based instance attribution methods constitutes an important practical barrier to their use. This complexity depends on the number of model parameters taken into consideration. As a result, computing IF is effectively infeasible if we consider *all* model parameters for modern, medium-to-large models such as BERT.

If we only consider the parameters of the last linear layer—comprising $O(p)$ parameters—to approximate the IF, the computational bottleneck will be the inverse Hessian which can be approximated with high accuracy in $O(p^2)$. There are ways to approximate the inverse Hessian more efficiently (Pearlmutter, 1994), though this results in worse performance. Similarity-based measures, on the other hand, can be calculated in $O(p)$.

With respect to wall-clock running time, calculating the influence of a single test sample with respect to the parameters comprising the top-5 layers of a BERT-based model for SST classification running on a reasonably modern GPU[2] requires $\sim$5 minutes. If we consider the linear variant, this falls to < 0.01 seconds. Finally, similarity-based approaches require < 0.0001 seconds. Extrapolating these numbers, it requires about 6 days to calculate IF (top-5 Layer) for all 1821 test samples in SST, while it takes only around 0.2 seconds for similarity-based methods.

5 Conclusions

Instance attribution methods constitute a promising approach to better understanding how modern NLP models come to make the predictions that they do (Han et al., 2020; Koh and Liang, 2017). However, approximating IF to quantify the importance of train samples is prohibitively expensive. In this work, we investigated whether alternative, simpler and more efficient methods provide similar instance attribution scores.

We demonstrated high correlation between (1) gradient-based methods that consider more parameters [IF and GD (top-5)] and their simpler counterparts [IF and GD (linear)], and (2) methods without Hessian information, i.e., IF vs GD and RIF vs GC. We considered even simpler, similarity-based approaches and compared the importance rankings over training instances induced by these to rankings under gradient-based methods. Through leave-some-out, randomized-test, and artifact detection experiments, we demonstrated that these simple similarity-based methods are surprisingly competitive. This suggests future directions for work on fast and useful instance attribution methods. All code necessary to reproduce the results reported in this paper is available at: `https://github.com/successar/instance_attributions_NLP`.

6 Ethical Considerations

Deep neural models have come to dominate research in NLP, and increasingly are deployed in the real world. A problem with such techniques is that they are opaque; it is not easy to know why models make specific predictions. Consequently, modern models may make predictions on the basis of attributes we would rather they not (e.g., demographic categories or ‘artifacts’ in data).

Instance attribution—identifying training samples that influenced a given prediction—provides a mechanism that might be used to counter these issues. However, the computational expense of existing techniques hinders their adoption in practice. By contrasting these complex approaches against simpler alternative methods for instance attribution, we contribute to a better understanding and characterization of the tradeoffs in instance attribution techniques. This may, in turn, improve the robustness of models in practice, and potentially reduce implicit biases in their predictions.

Acknowledgements

We thank the anonymous NAACL reviewers for their detailed feedback. Further, we also thank Matt Gardner, Sanjay Subramanian, and Dylan Slack for their useful comments. This work was sponsored in part by the Army Research Office (W911NF1810328), and funded in part by the NSF grants #IIS-1756023 and #IIS-2008956. The views expressed are those of the authors and do not reflect the policy of the funding agencies.

[2] Maxwell Titan GPU (2015).

References

Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. 2020. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909.

Samyadeep Basu, Philip Pope, and Soheil Feizi. 2020. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*.

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. 2019. Input similarity from the neural network perspective. In *Advances in Neural Information Processing Systems*, pages 5342–5351.

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.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana 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. IEEE.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. 2021. Evaluation of similarity-based explanations. *The Ninth International Conference on Learning Representations (ICLR)*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894.

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.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Barak A Pearlmutter. 1994. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160.

Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2019. Investigating robustness and interpretability of link prediction via adversarial modifications. 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 3336–3347.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint arXiv:2010.09030*.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. 2018. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9291–9301.

A Experimental Details

Datasets To evaluate different attribution methods, we conduct several experiments on sentiment analysis and NLI tasks, following prior work investigating the use of influence functions specifically for NLP (Han et al., 2020). We adopt a binarized version of the Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), consisting of 6920 training samples and 1821 test samples. As our NLI benchmark, we use the Multi-Genre NLI (MNLI) dataset (Williams et al., 2018), which contains 393k pairs of premise and hypothesis from 10 different genres. For model fine-tuning, we randomly sample 10k training instances. To evaluate the utility of different instance attribution methods in helping to unearth annotation artifacts in NLI, we use the HANS dataset (McCoy et al., 2019), which comprises examples exhibiting previously identified NLI artifacts such as lexical overlap between hypotheses and premises.We randomly sampled 1000 instances from this benchmark as test data to analyze the behavior of different attribution methods.

Models As discussed in the paper, we define models for both tasks on top of BERT, tuning hyperparameters on validation data via grid search. These hyperparameters include the regularization parameter $\lambda = [10^{-1}, 10^{-2}, 10^{-3}]$; learning rate $\alpha = [2 \times 10^{-3}, 2 \times 10^{-4}, 2 \times 10^{-5}, 2 \times 10^{-6}]$; number of epochs $\in \{3, 7, 10, 15\}$; and the batch size $\in \{8, 16\}$. Our final models achieve 90.6% accuracy on SST and 71.2% accuracy on MNLI

B Attribution Methods' Correlation

The complete version of Spearman correlation between attribution methods (containing the sentence-BERT) is provided in Figure 3. As expected, similarity-based approaches based on sentence-BERT show a very small correlation with other methods.

We also provide the proportion of shared examples in the top samples retrieved by IF (top-5) and IF (linear) in Figure 4. One can see that there is a very high correlation between these methods in top samples, validating the high quality of simpler version of IF (IF (linear)) in comparison to the more complex method (IF (top-5)).

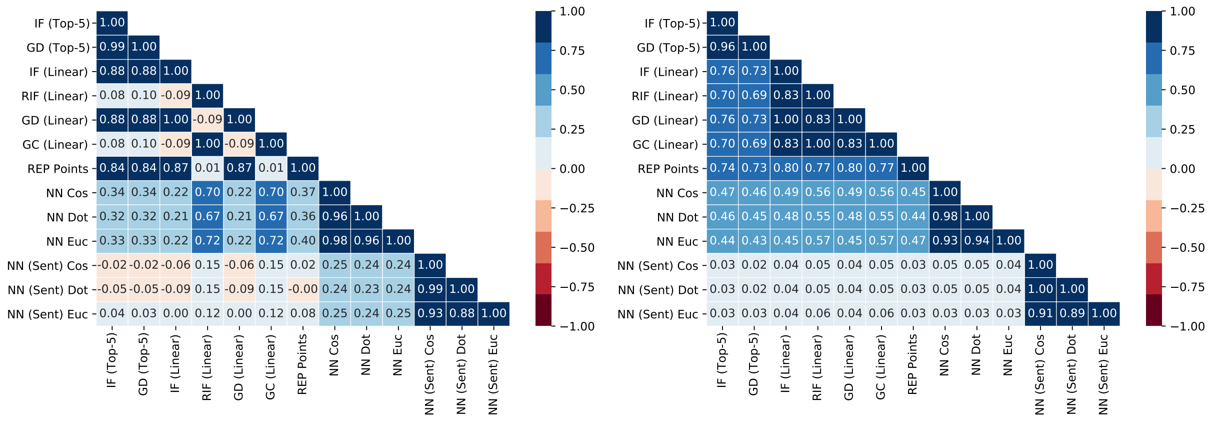
C Removing 'Important' Samples

In this experiment, we first select 50 random test samples (for both MNLI and SST). Then, for each one of these instances, we separately remove top-k (we consider k = 50 and 500) training instances for that test sample, retrain the model, and calculate the change in the model's prediction for that sample. We report the average changed over the prediction of the selected 50 random test samples in Table 1. Moreover, the proportion of common examples in top samples between pairs of attribution methods is depicted in Figures 5 and 6. The very high rate between IF vs GD, RIF vs GC, and NN-EUC vs NN-COS pairs, clarify the reason behind the similar performance of these pairs of methods in leave-some-out experiments.

D Near Training Samples Explanations

To further investigate the quality of the most influential sample based on different attribution methods, we conjecture that a data point very similar to a training sample should recover that sample as the most influential instance. We consider four scenarios to create target points similar to training data: (1) using training samples themselves as the target instances for attribution methods; (2) adding a random token to a random place in each training samples; (3) randomly removing a token from each training samples, and; (4) replacing a random token in each training samples with a random token from the dictionary of tokens. In the MNLI dataset, we apply each modification to both the premise and hypothesis in each training sample.

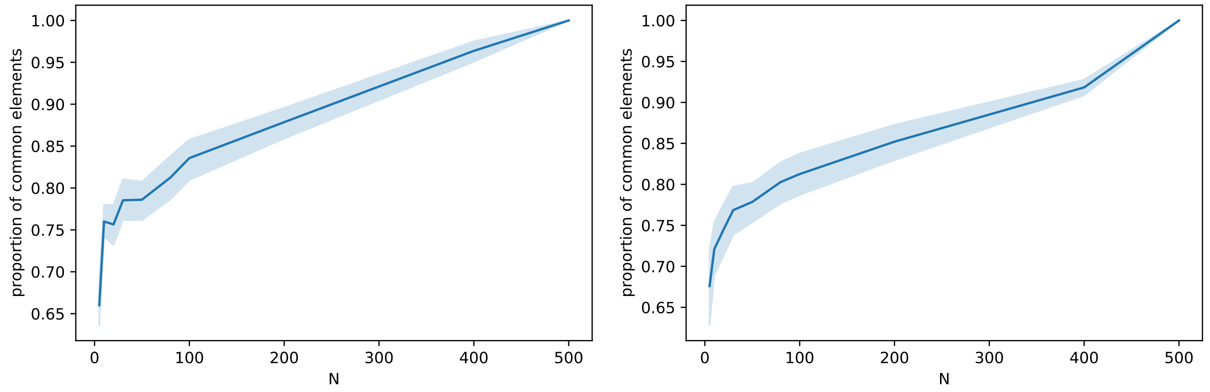
The result of this analysis is provided in Tables 3 and 4. We observe that similarity-based methods demonstrate a greater ability to recover the original training samples corresponding to the different targets. Moreover, the very low performance of IF, GC, and REP methods is due to the fact that there are training points with high magnitude gradient, which these methods choose as top instances for *any* target sample.



(a) SST.

(b) MNLI.

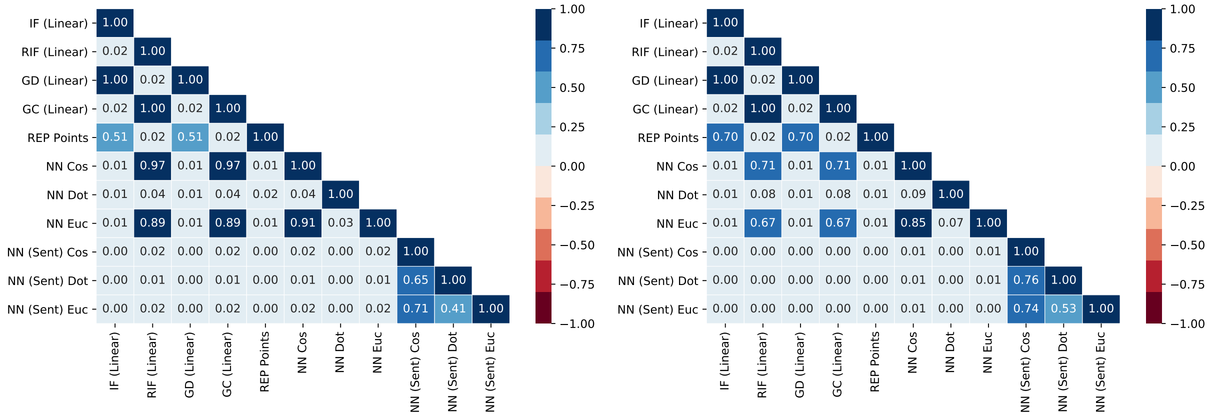
Figure 3: Complete version of correlation matrices.



(a) SST.

(b) MNLI.

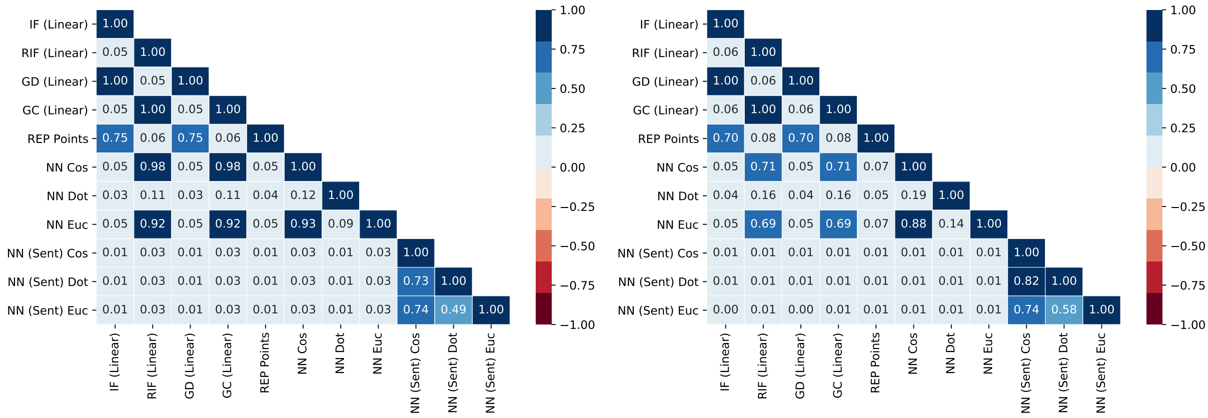
Figure 4: Proportion of common top examples between IF (Top-5) and IF (Linear) Methods. We selected 100 test examples and 500 training examples to compute the attributions over.



(a) Top-10 in SST.

(b) Top-10 in MNLI.

Figure 5: Proportion of common examples in top 10 samples between pairs of attribution methods.



(a) Top-50 in SST.

(b) Top-50 in MNLI.

Figure 6: Proportion of common examples in top 50 samples between pairs of attribution methods.

	Method	**Train**		**ADD**		**Remove**		**Replace**	
		HIT@1	HIT@10	HIT@1	HIT@10	HIT@1	HIT@10	HIT@1	HIT@10
Sim	NN EUC	100	100	99.9	100	66.5	73.7	99.9	100
	NN COS	100	100	99.8	100	67.3	74.6	99.8	100
	NN DOT	0.73	2.06	0.73	2.06	0.47	2.19	0.73	2.06
Gradient	IF	0.01	0.34	0.01	0.35	0.04	0.25	0.01	0.35
	REP	0.01	0.27	0.01	0.27	0.04	0.22	0.01	0.27
	RIF	95.8	96.0	95.9	96.0	65.0	72.2	95.8	96.0
	GD	0.01	0.38	0.01	0.38	0.04	0.23	0.01	0.37
	GC	95.9	96.0	95.9	96.0	65.3	72.3	95.9	96.0

Table 3: Treating the training samples and their modifications as the target samples for attribution methods over the SST dataset.

	Method	**Train**		**ADD**		**Remove**		**Replace**	
		HIT@1	HIT@10	HIT@1	HIT@10	HIT@1	HIT@10	HIT@1	HIT@10
Sim	NN EUC	100	100	100	100	36.7	45.8	100	100
	NN COS	100	100	100	100	38.1	46.8	100	100
	NN DOT	1.30	6.44	1.30	6.44	3.49	10.7	1.30	6.44
Gradient	IF	0.0	0.01	0.0	0.01	0.02	0.10	0.0	0.01
	REP	0.0	0.01	0.0	0.01	0.01	0.09	0.0	0.01
	RIF	92.5	92.5	92.5	92.5	32.6	41.2	92.5	92.5
	GD	0.0	0.01	0.0	0.01	0.10	0.50	0.0	0.01
	GC	92.5	92.5	92.5	92.5	32.8	41.2	92.5	92.5

Table 4: Treating the training samples and their modifications as the target samples for attribution methods over the MNLI dataset.