Robustness of Explanation Methods for NLP Models

Shriya Atmakuri, Tejas Chheda, Dinesh Kandula, Nishant Yadav, Taesung Lee, Hessel Tuinhof

Robustness of Explanation Methods for NLP Models

Shriya Atmakuri[1⋆], Tejas Chheda[1], Dinesh Kandula[1], Nishant Yadav[1], Taesung Lee[2], and Hessel Tuinhof[2]

[1] University of Massachusetts Amherst
Manning College of Information and Computer Sciences
[2] IBM Research

Abstract. Explanation methods have emerged as an important tool to highlight the features responsible for the predictions of neural networks. There is mounting evidence that many explanation methods are rather unreliable and susceptible to malicious manipulations. In this paper, we particularly aim to understand the robustness of explanation methods in the context of text modality. We provide initial insights and results towards devising a successful adversarial attack against text explanations. To our knowledge, this is the first attempt to evaluate the adversarial robustness of an explanation method. Our experiments show the explanation method can be largely disturbed for up to 86% of the tested samples with small changes in the input sentence and its semantics.

1 Introduction

Large and complex neural network models have become state-of-the-art in many computer vision and natural language processing tasks. However, the complexity that results in their effectiveness also causes a lack of interpretability. This is a major disadvantage of these models and makes it difficult to deploy them in sensitive applications where 'black box' solutions do not suffice. To combat this, a number of explainability methods have been developed. As deep neural networks (DNN) are being deployed in critical fields like autonomous driving and healthcare, explanations can help satisfy regulatory requirements [10], detect adversarial inputs, help practitioners debug their model, and reveal bias or other unintended effects learned by a model.

Intensive research on improving the DNN explainability has resulted in several either model-level or instance-level explanation methods. Prominent among these are gradient-based methods such as saliency mapping. As these methods are widely adopted, so too does the need to ensure that they behave in a reliable manner. Unfortunately, recent research has shed doubt on the validity and exposed vulnerability of explanation methods [1,2,3,5,9,21,22]. The latter work only considers continuous inputs such as image data. Similarly, work on improving

⋆ The first three authors have equal contribution.

the robustness of explanation methods ([6,7,13]) often focuses solely on continuous inputs.

In this work, the adversarial robustness of an explanation for discrete input data will be evaluated exemplified by an NLP model, taking a saliency mapping method as an example. In particular, we analyze how saliency maps change when inputs (*e.g.*, "a gorgeous, high-spirited musical") are perturbed to create new inputs (*e.g.*, "a resplendent, high-spirited musical") that maintain the semantics and the model prediction (*e.g.*, positive sentiment). We expect robust explanations to be invariant to such perturbations. Most of the prior work on evaluating the scope and quality of explanation only considers the image domain, which is continuous, and thus cannot be directly applied to text data with its discrete input nature. To the best of our knowledge, ours is the first attempt to understand the robustness of explanation methods towards adversarial text explanations in the NLP domain. We therefore hope that our work makes a first contribution towards improving the robustness of explanations in the case of discrete inputs such as text and tabular data.

Our preliminary experiments show the saliency mapping method is vulnerable to such an attack where up to 86% of the tested samples with small input perturbations have significant shift in the saliency map. We consider four different transformations (misspellings, word deletion, synonym substitution, and word inflection), and show the robustness of the saliency mapping for varying criteria to cover different application scenarios.

2 Related Work

In this paper, we consider an explanation method that receives a model and an input, and maps them to an attribution object of the same size as the input. Each entry in the output of the explanation method describes the relevance of the corresponding entry in the input for predicting the class.

For example, Integrated Gradients (IG) [19] uses the gradient of the output to compute the importance of the input. SmoothGrad (SG) [18] seeks to alleviate noise diffusion for saliency maps by averaging over explanations of noisy copies of an input.

Past work on understanding the adversarial robustness of explanation methods focuses on image data. [1] proposes an actionable methodology to evaluate explanations. They rely on visual information to support their findings. They find that Guided BackProp and Guided GradCAM are invariant to higher layer parameters. Nevertheless, the paper does not try to understand the explanation methods by perturbing the input features. They instead change the model parameters and input labels.

Authors in [11] introduce an input invariance axiom and propose that the axiom needs to be satisfied by explanation methods to ensure reliable explanation of the input's contribution to the model prediction. Although the work deals with transformations of the input, it only experiments with image data. Also the paper limits itself to only one simple transformation of the input: a constant shift

Fig. 1. Input attribution generated using Captum (IG). IG assigns very high attribution score to the word *happy* for positive sentiment.

Fig. 2. The word *not* is semantically important but is given very little weight by IG.

of the input. The constant shift transformation cannot be directly used on text inputs and we devise techniques to attack explanation methods using various transformations of the inputs.

Authors in [21] propose a new class of attacks that generate adversarial inputs not only misleading a target DNN but also deceiving its coupled interpreter. Work of [12] also proposed a similar attack method that generates an adversarial example using projected gradient descent. Both methods generate adversarial examples by perturbing the input which cannot be directly translated into the text domain as the perturbed embedding may not map to any word. Moreover, we aim to maintain the model prediction to focus on the robustness of the explanation method, not as a side-effect of deceiving the model.

3 Methods

3.1 Overview

To evaluate the adversarial robustness of an NLP model, we generate adversarial examples using a saliency-based explanation method, an access to the model needed by the explanation method (often white-box for gradient-based approaches), and transformations. We consider a text classifier such as sentiment classification and apply perturbations to change the input sentence such as synonym substitution, word deletion, and misspelling. Such transformations should result in perturbed inputs that are semantically equivalent to the original input and indistinguishable to a human observer. The transformations should also not alter the output of the model itself to reduce confounding factors. We then measure the change in the saliency mapping, which provides the attribution score, which show a positive or negative contribution to the predicted class. For example, in Figure 1, an explanation method assigns very high attribution score to the word *happy*, whereas in Figure 2, when the sentence is rephrased, the word *not* is given negative attribution while it is very important to model's prediction of positive sentiment.

Formally, for an input X consisting of a sequence of words $x_1, x_2, \ldots, x_n$ and a model M that produces a prediction P, the explanation E is defined as the vector of attributions $e_1, e_2, ..., e_n$ produced by applying an explanation method S to X and M where e_i indicates the importance of the word to M in producing P. A positive value of e_i indicates that x_i supported the prediction while a negative value indicates that x_i contradicted the prediction.

We then apply one or more transformations $t_1, \ldots, t_k \in T$ to generate a perturbed input $X' = t_k \circ \ldots \circ t_1(X)$, the corresponding explanation E', and prediction P' by M. We measure the attribution shift score $\text{Score}(E, E')$ using two methods: cosine similarity or L^∞ distance of E and E'. Cosine similarity captures the overall shift of the attribution. L^∞ instead focuses on the most important word. The attack is successful when $P = P'$ and $\text{Score}(E, E') > \theta$ where θ is an application specific threshold parameter.

3.2 Transformations

We consider four types of transformations for T to test the adversarial robustness of an explanation method: misspelling, synonym substitution, word inflection and word deletion. Each transformation applies necessary constraints to maintain the semantics of the sentence.

Misspellings Misspellings refers to replacing the original sentence with a perturbed one such that few of the words are incorrectly spelled due to orthographic changes akin to common human errors in pronunciation or wrongly understood phonetic structure of the word or typographical errors.

Word Deletion We delete a word and verify how the attribution across the perturbed text changes. The attack is again valid only if the meaning of the sentence remains same after deleting a word.

Synonym Substitution Synonym substitution involves replacing some of the words in the sentence with semantically equivalent words. The replacement words must be chosen such that the semantics of the sentence remain identical or nearly identical to the original sentence. In contrast to the word inflection attack, only synonyms with a different lemma are considered.

Word Inflection The word inflection attack involves replacing words in the sentence with different inflections of the same word. Inflections are words with the same lemma but can have different tense, quantity, etc.

3.3 Similarity-based Greedy Search

In order to effectively apply the above transformations, we choose the transformations that generate worst-case adversarial perturbations based on a similarity-based greedy search. The main idea of this algorithm is to find a perturbation

that has the highest effect on the attribution similarity score, while maintaining the constraints on how much the sentence can be changed. The constraints we have chosen to apply are as follows:

1. The model's prediction for the sentence's label must not change.
2. A maximum threshold is set for the fraction of words per sentence that can be transformed. (Generally 30% of the words)

Given a transformation and a candidate word, the algorithm performs the transformation on the word and recomputes the attribution score distribution. This process is repeated until no further words can be changed without violating the constraints or the attribution similarity score drops below the set threshold θ. The sentence obtained as a result of this process is the final perturbed sentence X' for a given transformation. This method can also be used for combined attacks, e.g. a different attack upon every iteration but that combination is out of the scope of this paper.

3.4 Semantic Similarity

To further ensure that the meaning of the sentences has not been changed after applying the transformations, we measure the semantic similarity between the original and sentence X and perturbed sentence X'. We use the sentence embeddings generated by S-BERT [17] for this purpose.

4 Experiments

4.1 Dataset

We run experiments on the Stanford Sentiment Treebank 2 (SST-2) which is a binary sentiment classification dataset released as part of the GLUE Benchmark [20]. It consists of extracted sentences from movie reviews (not the whole review) and either a positive or negative label assigned by a human annotator. For our experiments, we use the validation split which has 872 examples.

4.2 Model

The model we used is a RoBERTa base model finetuned on SST-2 [3]. The model was publicly shared on the HuggingFace model hub by TextAttack [15]. The model has an accuracy of 94.04% on the validation data.

[3] https://huggingface.co/textattack/roberta-base-SST-2

4.3 Attack Implementation

Misspellings To generate perturbations of the original sentence for the misspelling transformation, we take the most attributed word (positive) and replace if with a misspelled word. For the task of finding the misspelled word, we use the "birkbeck" dataset [14] which contains misspellings of 6000+ commonly used words. If the given word is not present in the dataset, we perform a QWERTY substitution. The QWERTY substitution attack is provided by the TextAttack library and contains common keyboard based human errors.

Table 1 shows examples of a few misspelling perturbations. The words that are perturbed are shown in italics.

Table 1. Examples of misspellings attack with the misspelled word in *italics*.

	Original Sentence	Perturbed Sentence
1	the acting , costumes , music , cinematography and sound are all *astounding* given the production 's austere locales .	the acting , costumes , music , cinematography and sound are all *astoumding* given the production 's austere locales .
2	a sequence of ridiculous shoot - 'em - *up* scenes .	a sequence of ridiculous shoot - 'em - *jp* seens .

Word Deletion We choose the least attributed words as candidates for word deletion. Since the sentence is tokenized into sub-words, the explanation methods assigns attribution scores to each token rather than the word. So, to calculate the score at the word level, we averaged the scores of all the tokens present in the word. As outlined in Section 3.4, we use S-BERT to ensure that the deletion does not significantly change the semantics of the sentence which is very important with this transformation.

Synonym Substitution Three different approaches were attempted for synonym substitution. The first approach involved choosing synonyms for the word using WordNet [8]. This turned out to be unsuitable for the task as WordNet does not perform word-sense disambiguation. Some of the substitutions produced were of good quality and retained semantic similarity (example 1.2 in Table 2) but others did not. In example 1.1 in Table 2, the original sentence uses the word *last* to mean *previous* but the substitution is the word *endure* which is a different sense of *last*.

The second approach considered used BERT [4]. As BERT's original training task is predicting masked out words from sentences, this can be adapted to generating substitutions. The word to be substituted is masked out and the masked sentence is passed to the model. The model's top predictions for the masked out word can be used as substitutes in the perturbed sentence. However,

Table 2. Examples of synonym substitution attack based on WordNet and word embeddings.

	Original Sentence	Perturbed Sentence
	WordNet	
1.1	Or doing *last* year's taxes with your ex-wife	Or doing *endure* year's taxes with your ex-wife
1.2	Unflinchingly *bleak* and desperate	Unflinchingly *stark* and desperate
	Embeddings	
2.1	a *gorgeous* , high-spirited musical from india that exquisitely blends music , dance , song , and high drama .	a *resplendent* , high-spirited musical from india that exquisitely blends music , dance , song , and high drama .

Table 3. Percentage of test samples with changes in the predicted label and semantics of the input sample along with average number of words perturbed and success rate for each type of attack.

Attack	Δ Label (%)	Δ Semantics (%)	$\varnothing$ Perturbations	Success (%)
Word Deletion	5.5	5.0	3.0	32.1
Synonym Substitution	9.0	5.6	1.6	67.1
Inflection	1.2	1.3	2.0	39.5
Misspelling	4.3	8.5	1.8	86.0

since the original word is masked out, BERT only looks at the semantics of the sentence and not the word itself. This leads to it often producing substitutions that are completely unrelated to the original word although they fit in the context of the sentence.

Finally, we perform synonym substitution using counter-fitted word embeddings [16]. Unlike traditional word embeddings, these embeddings are trained with linguistic constraints to ensure that antonyms are not nearest neighbors. This method produced appropriate results for most of the examples and was the final choice.

Word Inflection Word inflection was performed using the LemmInflect[4] library. LemmInflect uses a dictionary approach to lemmatize English words and inflect them. It works with out-of-vocabulary (OOV) words by applying neural network techniques to classify word forms and choose the appropriate morphing rules. LemmInflect has a 95.6% accuracy on the AGID database [5]

4.4 Results

Table 3 shows summary of the various attacks. Figure 3 shows the number of successful attacks for each bucket of cosine similarity between the explanation

[4] https://github.com/bjascob/LemmInflect
[5] http://wordlist.aspell.net/other/

vectors before and after an adversarial perturbation. Figure 4 shows the number of successful attacks for each bucket of L_∞ distance between the explanation vectors before and after an adversarial perturbation. An attack is referred to as being *successful* if the prediction does not change, *and* the explanation vectors differ significantly for the original and the perturbed sentence *i.e.*, having low cosine similarity.

Misspellings As shown in Figure 3, majority of the samples have cosine similarity between explanation vectors around 0.4 for the misspelling attack. Around 770 out of 872 samples had explanations with a similarity of less than the 0.6 threshold with respect to the original sentences. This implies that even though the predicted label was the same, the explanation method is not robust of misspelling based perturbation. A potential reason for this could be that since we are misspelling the most attributed word in the original sentence, the attribution scores get distributed over other words in the perturbed sentence. This may lead to significantly different explanation vector than the original and lead to low cosine similarity between explanation vectors before and after the attack.

It is also important to check whether or not the perturbations themselves significantly change the semantics of the original sentences which can cause the explanation method to fail. Figure 5a shows scatter plot of BERT-based semantic similarity between original and perturbed sentences and cosine similarity between corresponding explanation vectors. A successful attack would have most datapoints in the top-left corner which correspond to high semantic similarity between the original and perturbed sentence along with low cosine similarity between corresponding explanations. For misspellings attack, 705 examples out of the total 872 examples have sentence similarity greater than 0.7 and cosine similarity less than 0.5.

Word Deletion Figure 3 shows that majority of the samples have high cosine similarity between explanation vectors before and after deleting the least important word from the sentence. This is apparently because we are deleting *least* important word in sentence according to the attribution scores, and this causes marginal changes to overall attribution scores for the sentence. We experimented with deleting one or two least important words in a single attack as well as applying greedy search with threshold of 0.5 as explained in Section 3.3. The attack was successful for only 7% of the sentences for single word deletion and the attack was successful for 65 sentences out of 755 when deleting two words. With greedy search, the attack was successful for 32% of the samples. Overall, Integrated gradients explanation method is largely robust against our proposed word deletion attack.

Synonym Substitution The synonym substitution attack automated with greedy search produced similar results to misspelling although it was slightly less effective. In Table 3 we can see that it has the second highest percentage

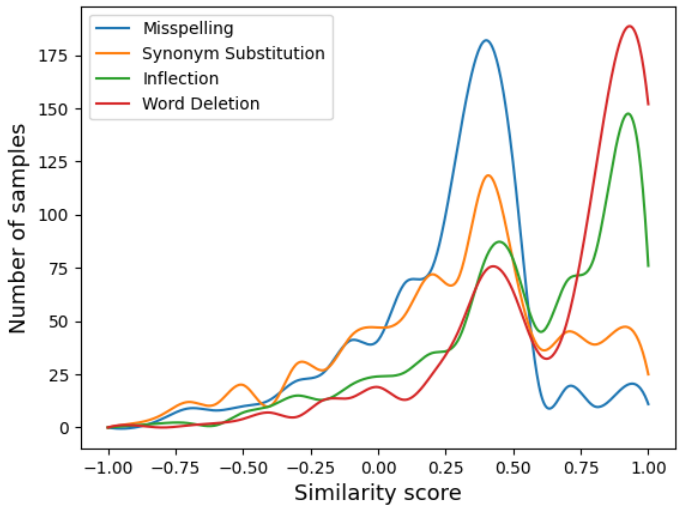


Fig. 3. Number of test samples v. cosine similarity between explanation vectors before and after adversarial perturbation to the input.

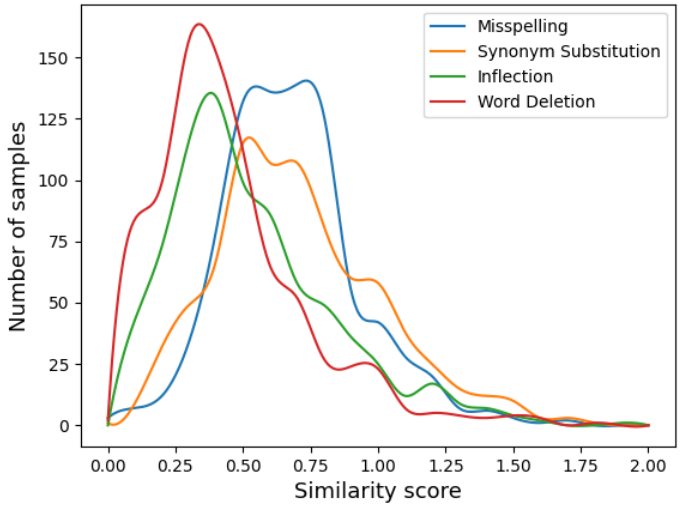
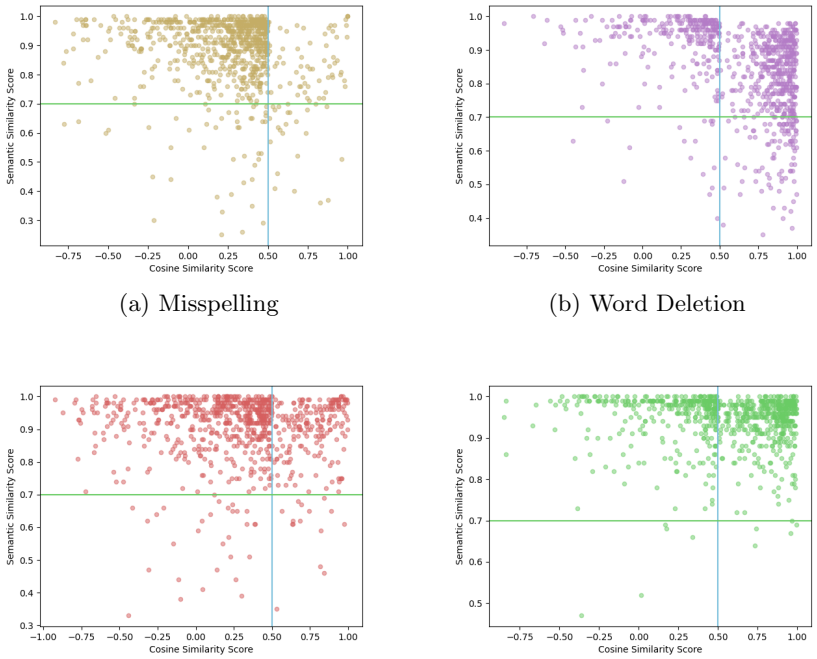


Fig. 4. Number of test samples v. L_∞ distance between explanation vectors before and after adversarial perturbation to the input.



(a) Misspelling (b) Word Deletion

(c) Synonym Substitution (d) Word Inflection

Fig. 5. Semantic similarity versus cosine similarity of explanation vectors of original and perturbed input samples.

of succesful attacks at 67%. In Figure 3, it can be observed that the synonym substitution peak is slightly to the right of the misspelling peak, indicating slightly higher cosine similarity in the attribution vectors. However, it remains much to the left of the inflection and word deletion peaks. In Figure 5c we can also note that the number of points in the top left corner is lower than for the misspelling plot but higher than for the remaining transformations.

Word Inflection The word inflection attack is only slightly more effective than the word deletion one. It has a success percentage of 39.44. Unsurprisingly, most of the points in Figure 5d are well above the 0.7 sentence similarity line as this attack produces only a very minor change in each word affected. However, it is also not very effective at changing the attribution vectors, possibly for the same reason.

5 Conclusion

Explanation methods are crucial going forward into the future where ML models will be deployed in critical applications. The goal of our work is to devise methods to automatically test robustness of the explanation methods specifically in text domain. We performed multiple types of attacks on text under constraints that ensured meaning remained consistent and devised methods to measure the change in explanations. We primarily focused on Integrated Gradients explanation method but our work could easily be extended to other methods as well. We found Integrated Gradients method is not robust against Misspelling and Synonym substitution attacks as the explanation changed heavily upon attack. We believe this is a small step in the right direction towards testing robustness of explanation methods.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada. (2018)
2. Adebayo, J., Muelly, M., Abelson, H., Kim, B.: Post hoc explanations may be ineffective for detecting unknown spurious correlation. In: International Conference on Learning Representations (2021)
3. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049 (2018)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Dombrowski, A.K., Alber, M., Anders, C., Ackermann, M., Müller, K.R., Kessel, P.: Explanations can be manipulated and geometry is to blame. Advances in Neural Information Processing Systems **32** (2019)
6. Dombrowski, A.K., Anders, C.J., Müller, K.R., Kessel, P.: Towards robust explanations for deep neural networks. Pattern Recognition **121**, 108194 (2022)
7. Etmann, C., Lunz, S., Maass, P., Schoenlieb, C.: On the connection between adversarial robustness and saliency map interpretability. In: International Conference on Machine Learning. pp. 1823–1832. PMLR (2019)
8. Fellbaum, C.: Wordnet. In: Theory and applications of ontology: computer applications, pp. 231–243. Springer (2010)
9. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: AAAI Conference on Artificial Intelligence. pp. 3681–3688 (2019)
10. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI magazine **38**(3), 50–57 (2017)
11. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 267–280. Springer (2019)
12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)

13. Mishra, S., Dutta, S., Long, J., Magazzeni, D.: A survey on the robustness of feature importance and counterfactual explanations. arXiv preprint arXiv:2111.00358 (2021)
14. Mitton, R.: Corpora of misspellings for download (1985)
15. Morris, J., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 119–126 (2020)
16. Mrkšić, N., Séaghdha, D.O., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.H., Vandyke, D., Wen, T.H., Young, S.: Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892 (2016)
17. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
18. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
19. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
20. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
21. Zhang, X., Wang, N., Shen, H., Ji, S., Luo, X., Wang, T.: Interpretable deep learning under fire. In: The 29th USENIX Security Symposium (2020)
22. Zhou, Y., Booth, S., Ribeiro, M.T., Shah, J.: Do feature attribution methods correctly attribute features. In: AAAI Conference on Artificial Intelligence (2022)