Can We Really Trust Explanations? Evaluating the Stability of Feature Attribution Explanation Methods via Adversarial Attack

Zhao Yang[1,2], **Yuanzhe Zhang**[1,2], **Zhongtao Jiang**[1,2],
Yiming Ju[1,2], **Jun Zhao**[1,2], **Kang Liu**[1,2,3*]
[1]School of Artificial Intelligence, University of
Chinese Academy of Sciences / Beijing, 100049, China
[2]National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences / Beijing, 100190, China
[3]Beijing Academy of Artificial Intelligence / Beijing, 100084, China
{zhao.yang,yzzhang,zhongtao.jiang}@nlpr.ia.ac.cn
{yiming.ju,jzhao,kliu}@nlpr.ia.ac.cn

Abstract

Explanations can increase the transparency of neural networks and make them more trustworthy. However, can we really trust explanations generated by the existing explanation methods? If the explanation methods are not stable enough, the credibility of the explanation will be greatly reduced. Previous studies seldom considered such an important issue. To this end, this paper proposes a new evaluation frame to evaluate the *stability* of current typical feature attribution explanation methods via textual adversarial attack. Our frame could generate adversarial examples with similar textual semantics. Such adversarial examples will make the original models have the same outputs, but make most current explanation methods deduce completely different explanations. Under this frame, we test five classical explanation methods and show their performance on several stability-related metrics. Experimental results show our evaluation is effective and could reveal the *stability* performance of existing explanation methods.

1 Introduction

Fueled by recent rapid development in deep learning, NLP systems have obtained promising results in several fields, such as medical, law and commerce (Rudin, 2019; Bommasani et al., 2021). However, besides the predicted results, users concern more on how these results are generated (Lipton, 2018). To this end, lots of emphases have been set upon the explanation methods for neural networks (Ribeiro et al., 2016; Li et al., 2016; Simonyan et al., 2013; Bastings et al., 2019).

Although the current explanation methods have increased the transparency of the neural networks and provided explanations as supports for predicted results, most of them ignored important questions: *are these methods reliable and the generated explanations really trustful?* Besides the widely used focused properties of explanation methods, such as faithfulness, plausibility (Adebayo et al., 2018; Jacovi and Goldberg, 2020; Atanasova et al., 2020), readableness (Bastings et al., 2019) and compactness (Miller, 2019; Jiang et al., 2021), we believe *stability* is an important but often overlooked property (Robnik-Šikonja and Bohanec, 2018). When we put a small perturbation on the input, which would not change the input semantic and the output of the original model, we believe that the explanation method is not stable enough when we obtain the same outputs with quite different explanations. For example, Figure 1 shows all results of major explanation methods would change when we just replace `fine` by `refined`, including LIME (Ribeiro et al., 2016), Leave-one-out (Li et al., 2016), Vanilla Gradient (Simonyan et al., 2013), Smooth Gradient (Smilkov et al., 2017), Integrated Gradient (Sundararajan et al., 2017).

To fulfill the *stability* testing, we intuitively consider existing word-substitution based textual adversarial attack methods[0](Ren et al., 2019; Zang et al., 2020), since it is under the black-box[1] settings and

*Corresponding author.

[0]Feature attribution based explanation methods show the importance of each token to the prediction. Therefore, paraphrase-based attack methods do not fit because they would modify too many parts of inputs at once.

[1]Black-box refers to we can only utilize the outputs of the model during the attack. However, some explanation methods are

The movie exists for its soccer action and its fine acting.
Label: Positive **Attribution Order:** 1st 2nd 3rd

LeaveOneOut **Label: Positive**
Ori: The movie exists for its soccer action and its fine acting.
Adv: The movie exists for its soccer action and its terrific acting.

LIME **Label: Positive**
Ori: The movie exists for its soccer action and its fine acting.
Adv: The special exists for its soccer action and its fine acting.

Vanilla Gradient **Label: Positive**
Ori: The movie exists for its soccer action and its fine acting.
Adv: The movie exists for its soccer action and its refined acting.

Smooth Gradient **Label: Positive**
Ori: The movie exists for its soccer action and its fine acting.
Adv: The movie exists for its soccer action and its gorgeous acting.

Integrated Gradient **Label: Positive**
Ori: The movie exists for its soccer action and its fine acting.
Adv: The movie exists for its soccer behavior and its good acting.

Figure 1: An example of the result of our adversarial attack. We select a sentence from SST-2 and show the adversarial examples for explanation method **Vanilla Gradient** (Simonyan et al., 2013). `Ori` and `Adv` stand for original sentence and corresponding adversarial example respectively. We show the three most important tokens and sign them in different colors.

no need for the transparency of the model framework. However, we could not directly extend the current adversarial attack on the explanation methods. In our explanation stability test setting, the attack method should ensure the original prediction model has unchanged outputs for the adversarial examples, but the explanations vary, which is obviously different from the target of the common textual adversarial attacks. Thus, the main challenge is, for such adversarial examples, how to ensure the explanations are different but the outputs of the original model are the same. To this end, we modified the target of the standard textual adversarial attack to keep the prediction label of the adversarial examples unchanged. At the same time, we define two criteria to measure the difference between two explanations and add them respectively to the score function. Such explanation difference measurements are used to help the judgment of the adversarial examples' qualities in the attacking procedure.

Finally, we put the attack on five typical feature attribution explanation methods. Experimental results show their performance on *stability*. We find perturbation-based explanation methods perform better on *stability* than gradient-based methods. All of the source code and data will be available soon.

2 Related Work

2.1 Feature Attribution Explanation Method

Feature attribution explanation methods score each token of the input based on its contribution to the prediction label. We can easily find the key tokens according to the attribution value. These explanation methods can be simply classified as below two categories: perturbation-based methods and gradient-based methods.

Perturbation-based get the attribution score by perturbing the input sequence: **LIME** (Ribeiro et al., 2016) sampled enough new sequences from the neighbor of the input sequence and fit the output logits of these sampled sequences by a linear function, the coefficients of the fitted function are the attribution

not black-box such as gradient-based methods. Whether the explanation method is black-box has nothing to do with our black box attack method.

score for each token. **Leave-one-out** (Li et al., 2016) observed the probability change on the predicted class when erasing some certain word and the value of probability change is the attribution score for the removed word. Gradient-based methods compute the attribution score according to the gradient of the input: **Vanilla Gradient** (Simonyan et al., 2013) simply computed the gradient of the loss with respect to each token. **Smooth Gradient** (Smilkov et al., 2017) added small Gaussian noise to every embedding and take the average gradient value as the final attribution score for each token. **Integrated Gradient** (Sundararajan et al., 2017) integrated the gradient along the path from a sequence of all-zero embeddings to the original input and take the integral value as the attribution score.

2.2 Evaluation of Explanation Methods

Recently, a collection of explanation methods has emerged exploring to interpret neural networks. To compare these explanation methods, various explanation metrics have been proposed. Faithfulness refers to how accurately the explanation reflects the true reasoning process of the model (Herman, 2017; Wiegreffe and Pinter, 2019; Jacovi and Goldberg, 2020). Plausibility refers to how convincing the explanation is to humans by comparing explanations that generated by explanation methods and human annotated explanations (Atanasova et al., 2020; DeYoung et al., 2019). Besides, readableness measures whether human could understand the explanations (Molnar, 2020) and compactness requires a explanation should be short or selective (Miller, 2019; Jiang et al., 2021). However, these evaluation metrics ignore whether the explanation method is reliable.

To evaluate the reliability of existing explanation methods, *consistency* and *stability* have been proposed. However, *consistency* is quite different from *stability* actually. To evaluate *consistency*, existing studies usually modified original model to generate different explanations when the inputs and outputs keep unchanged. Jain and Wallace (2019) modified the attention value and maintain the output unchanged to illustrate attention is not explanation. Heo et al. (2019) applied adversarial model manipulation to generate different explanations. Slack et al. (2020) aims to sample based explanation methods. They modified the original classifier into two parts: original classifier for original instances and another model for instances in neighbor. Wang et al. (2020) construct a new model which has similar outputs with original model but definitely different gradient. They added this model on original model and the added model shows similar prediction but totally different gradient-based explanations. Indeed, they all try to modified the original model to generate different explanations. However, for *stability*, we just put perturbation on inputs not on model, which is extremely different with *consistency*.

For *stability*, though existing works defined its specific meanings, only a few work design corresponding experiments to evaluate the performance of *stability*. (Ghorbani et al., 2019) applied pixel-level perturbations to evaluate the stability. However, pixel-level perturbations can not be easily transferred in NLP. In NLP only (Ding and Koehn, 2021) evaluated this property by manually constructing similar instances, which is much time-consuming and expensive. Therefore, in this paper, we automatically construct similar instances by learning from textual adversarial attack.

3 Formulation

In this section, we first introduce the basic information of the common textual adversarial attack in Section 3.1. Then we introduce how to formulate explanation adversarial attack in Section 3.2.

3.1 Textual Adversarial Attack

Formally, suppose that a sentence $x_k = \omega_1 \omega_2 \cdots \omega_n$, where ω_i is the i-th word in x_k. For a given classifier $P(y|x)$ and label set $Y = (y_1, y_2, ..., y_m)$, the model prediction y_k for x_k can be formulated as $y_k = arg\max_{y \in Y} P(y|x_k)$. The target is to find $x_k^{'}$, which can be formulated as:

$$\begin{aligned} & x_k^{'} \\ s.t. \quad & y_k \neq y_k^{'}, \left\| x_k^{'} - x_k \right\| < \epsilon \end{aligned} \tag{1}$$

where $x_k^{'}$ is the adversarial example of x_k. The core constraint is to ensure the difference between x_k and $x_k^{'}$ is small enough. In this paper, we ensure the semantics of x_k and $x_k^{'}$ to be as similar as possible,

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

which has been shown more imperceptible for human (Zhang et al., 2020).

3.2 Explanation Adversarial Attack

Feature attribution explanation method can generate an explanation $e_k = (s_1, s_2, \cdots, s_n)$ according to x_k and its prediction y_k, where s_i is the attribution score of ω_i. Therefore, the target is to find $x_k^{'}$, which can be formulated as follow:

$$\begin{aligned} & x_k^{'} \\ s.t. \qquad & e_k \neq e_k^{'}, y_k^{'} = y_k, \left\| x_k^{'} - x_k \right\| < \epsilon \end{aligned} \tag{2}$$

We also follow the common textual adversarial attack to keep the semantics of x_k and $x_k^{'}$ to be similar. And the most important difference is an extra constraint $y_k^{'} = y_k$, we must ensure this constraint should be satisfied because of the definition of *stability*. Obviously, the constraint is contrary to the target of common textual attack, where $y_k^{'} \neq y_k$. By contrast, our target is to ensure the explanations are different. Therefore, we will define how to measure explanation difference in the following section.

4 Attack Method

According to Section 3.2, we need to measure the explanation difference. Therefore, we propose two metrics in Section 4.1. Then we present our detailed attack strategies to attack existing explanation methods in Section 4.2.

4.1 Measuring the Explanation Difference

For feature attribution methods, people usually do not care the specific attribution score of each token but the relative importance ranks of these tokens. Therefore, we consider the rank differences between explanations. We can easily get the corresponding rank sequence R_k for explanation E_k in descending order, where $R_k = (r_1^k, r_2^k, ..., r_n^k)$, r_i^k stands for the descending rank of the i-th token in x_k. We can also get the corresponding position sequence $P_k = (p_1^k, p_2^k, ..., p_n^k)$ via `argsort`, p_i^k stands for the index of the i-largest attribution score in x_k. Based on this, we design two quantitative criteria to measure the difference between explanations.

Rank-count: In this setting, we compute the number of positions whose rank has changed:

$$d_{count}(E_i, E_j) = \sum_{k=1}^{n} ||r_k^i - r_k^j||_0 \tag{3}$$

where $||\cdot||_0$ refers to the L0 norm.

Rank-topk: In this setting, we compute the size of intersection set of two position set of the top-k rank. The top-k set for e_i is the first k elements of position sequence r_i: $E_{topk}^i = \{p_1^i, p_2^i, ..., p_k^i\}$.

$$d_{topk}(E_i, E_j) = |E_{topk}^i \cap E_{topk}^j| \tag{4}$$

where $|\cdot|$ refers to the size of a set.

For example, given $E_1 = \{0.1, 0.5, 0.3, 0.2\}$ and $E_2 = \{0.6, 0.3, 0.4, 0.2\}$. We get the rank sequence $R_1 = \{3, 0, 1, 2\}$ and $R_2 = \{0, 2, 1, 3\}$, then we can get the position sequence $P_1 = \{1, 2, 3, 0\}$ and $P_2 = \{0, 2, 1, 3\}$. Accordingly, we compute $d_{count}(E_1, E_2) = 3$ and $d_{topk}(E_1, E_2) = 2$ when $k = 3$.

4.2 Attack Strategies

Word-substitution based textual adversarial attack methods usually consist of two main steps: determining substitution order and selecting substitution words. In different steps, we employ different strategies. To determine the substitution order, we modify Samanta and Mehta (2017) as an example. To select substitution words, we utilize OpenHowNet (Qi et al., 2019) as the substitution resource (Zang et al., 2020). Notably, other word-substitution based adversarial attack methods (Ren et al., 2019; Alzantot et al., 2018; Zang et al., 2020) are also applicable.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

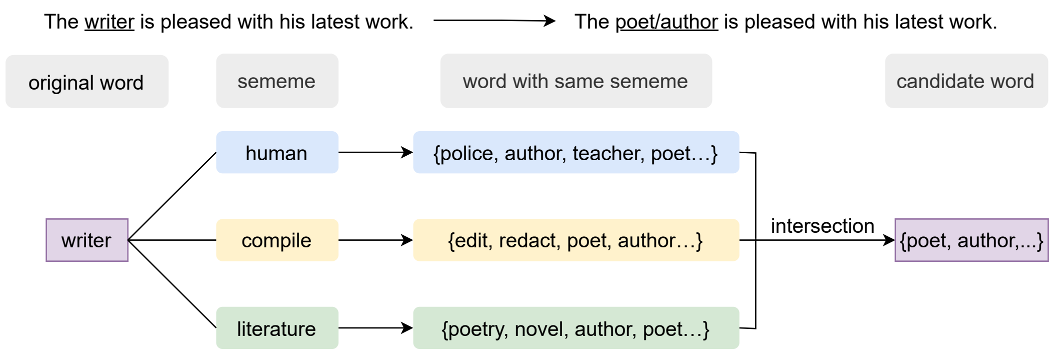

Figure 2: An example of how to construct candidate substitution word set for the word `writer` by its sememes `human`, `compile` and `literature`.

4.2.1 Determining Substitution Order

Formally, for a sentence $x = \omega_1\omega_2\cdots\omega_i\cdots\omega_n$, to determine the substitution order, we compute the word saliency WS_i for token ω_i first. To compute WS_i , we should get $\hat{x}_i = \omega_1\omega_2\cdots\ \mathbf{0}\ \cdots\omega_n$ by replacing ω_i with $\mathbf{0}$.

$$WS_i = P(y_{ori}|x) - P(y_{ori}|\hat{x}_i) \tag{5}$$

where y_{ori} refers to the original output label. We calculate the word saliency WS_i for all $\omega_i \in x$ and then we sort all of the tokens in descending order based on their saliency value. Then we substitute the words in this order (Samanta and Mehta, 2017).

4.2.2 Selecting Substitution Words

We construct candidate substitution set via sememes and utilize OpenHowNet (Qi et al., 2019) as the resource. Sememe is the minimum semantic unit of language (Bloomfield, 1926) and the sememes of one word can composite the meaning of this word. Therefore, words that have the same sememe can substitute for each other (Zang et al., 2020). As shown in Figure 2, when we want to find substitution words for the original word `writer`. We utilize OpenHowNet to get its sememes `human`, `compile` and `literature`. Then we get three word sets that has these three sememes respectively. Finally, we compute the intersection of these three word sets and get the substitution word `poet` and `author` for the original word `writer`. According to Qi et al. (2019) and Zang et al. (2020), when we replace the word with the obtained substitution word, the semantic of the original sentence would not change.

After getting substitution set for the original word by above method, we still have to choose which word to substitute the original word. Therefore, we also need a quantitative criterion to help us to find the most suitable substitution word from the whole substitution set. Specifically, we define our score function as follow:

$$score(x_1, x_2) = d(e_1, e_2) \times (1 - ||y_1 - y_2||_0) \tag{6}$$

where $d(e_1, e_2)$ represent the explanation difference for x_1, x_2 and we directly employ the Equation (3) and Equation (4). y_1, y_2 are the prediction label for x_1, x_2. We directly force the labels must be same, otherwise the score would be zero.

With this score function, we can get the substitution word ω_i^* for ω_i in $x_i = \omega_1\omega_2\cdots\omega_i\omega_n$. This process can be formulate as follow:

$$\omega_i^* = \arg\max_{\omega_i \in L_{\omega_i}} score(x, x_i^{'}) \tag{7}$$

where $x_i^{'} = \omega_1\omega_2\cdots\omega_i^{'}\cdots\omega_n$ and L_{ω_i} is the candidate set for the word ω_i. Finally, ω_i^* is the substitution word for ω_i is x.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.

5 Experiments

5.1 Datasets and Models

Following previous explanation studies (DeYoung et al., 2019; Atanasova et al., 2020), we also select sentiment analysis as the target task. In specific, we choose SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011) as the test benchmark dataset and select the base version of BERT (Devlin et al., 2018) and BiLSTM (Conneau et al., 2017) as the target model.

For BERT, we utilize the base version of BERT. For BiLSTM, the hidden states are 256-dimensional and we utilize the 300-dimensional pre-trained Glove (Pennington et al., 2014) word embeddings. Our reproduced BERT can achieve accuracy of 91.28% and 91.36% on SST-2 and IMDB respectively. And BiLSTM can achieve accuracy of 85.50% and 90.38% on SST-2 and IMDB respectively.

To improve evaluation efficiency, we randomly sample 500 correctly classified instances with the length of 10-100 from the test set.

5.2 Explanation Methods

We select five classical feature attribution explanation methods in the two mainstream types to conduct our experiments:

A. Perturbation-based Explanation Method:

LIME (Ribeiro et al., 2016) sampled enough sentences from the neighbor of the input and fit the output logits of these samples by a linear function. The coefficients of the obtained linear function is the corresponding attribution scores.

LeaveOneOut (LOO) (Li et al., 2016) observed the probability change on the predicted class when erasing each word one by one and take this change value as the atribution score.

B. Gradient-based Explanation Method:

VanillaGradient (VG) (Simonyan et al., 2013) simply computed the gradient of the model loss with respect to the token and multiply with its embedding as its corresponding attribution score.

$$a_i = x_i \cdot \frac{\partial f(x_i)}{\partial x_i} \tag{8}$$

SmoothGradient (SG) (Smilkov et al., 2017) added small Gaussian noise to every embedding N times and average these N VanillaGradient value as the final attribution score.

$$a_i = \frac{1}{N}\sum_{i=1}^{N}(x_i + \mathcal{N}(0,1)) \cdot \frac{\partial f(x_i + \mathcal{N}(0,1))}{\partial (x_i + \mathcal{N}(0,1))} \tag{9}$$

where $\mathcal{N}(0,1)$ refers to the Gaussian noise.

IntegratedGradient (IG) (Sundararajan et al., 2017) integrated the gradient along the path from a basic sequence $x_i^{'}$ to the original input x_i and take the integral value a_i as the attribution.

$$a_i = (x_i - x_i^{'}) \int_{\alpha=0}^{1} \frac{\partial f(x_i^{'} + \alpha \times (x_i - x_i^{'}))}{\partial \alpha} d\alpha \tag{10}$$

Specifically, it is time-consuming to compute intergral value. To improve computation efficiency, we divide the integral area into K parts and obtain the approximate value of a_i (Sundararajan et al., 2017).

$$a_i = (x_i - x_i^{'}) \sum_{m=1}^{K} \frac{\partial f(x_i^{'} + \frac{m}{K} \times (x_i - x_i^{'}))}{\partial x_i} \times \frac{1}{K} \tag{11}$$

5.3 Experimental Settings and Results

5.3.1 Explanation Similarity

Firstly, we fix m modified words to generate corresponding adversarial examples whose explanations are the most different. Then we use explanation similarity to evaluate the stability of explanation methods.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.

Model	Dataset	Explanations	m=1			m=2			m=3		
			change↓	*spearman*↑	*inte*↑	*change*↓	*spearman*↑	*inte*↑	*change*↓	*spearman*↑	*inte*↑
BERT	SST-2	LIME	**79.87**	**0.80**	**3.87**	**84.03**	**0.78**	**3.81**	**86.52**	**0.76**	**3.75**
		LOO	89.13	0.64	3.14	92.62	0.62	3.09	94.12	0.61	3.03
		VG	92.99	0.48	2.83	95.65	0.45	2.71	97.11	0.42	2.64
		SG	92.86	0.55	2.92	95.71	0.53	2.87	96.70	0.52	2.83
		IG	86.79	0.71	3.45	90.01	0.69	3.38	91.69	0.67	3.37
	IMDB	LIME	**84.60**	**0.92**	**4.23**	**88.65**	**0.90**	**4.08**	**90.04**	**0.88**	**3.87**
		LOO	90.10	0.84	3.48	93.47	0.79	3.12	95.22	0.76	2.91
		VG	92.75	0.79	3.23	95.44	0.73	2.88	96.65	0.69	2.66
		SG	92.48	0.82	3.29	95.26	0.76	2.89	96.60	0.73	2.67
		IG	85.49	0.91	4.07	89.58	0.89	3.90	91.37	0.87	3.81
BiLSTM	SST-2	LIME	**71.18**	**0.81**	**4.02**	**80.38**	**0.74**	**3.78**	**84.22**	**0.68**	**3.63**
		LOO	75.76	0.77	3.89	84.07	0.71	3.70	86.96	0.67	3.60
		VG	78.20	0.75	3.78	85.04	0.62	3.52	88.50	0.56	3.36
		SG	77.83	0.77	3.85	84.49	0.68	3.55	87.21	0.64	3.40
		IG	73.55	0.79	3.99	81.73	0.72	3.75	85.39	0.67	3.61
	IMDB	LIME	**81.44**	**0.90**	**4.24**	**86.36**	**0.86**	**4.07**	**88.25**	**0.84**	**3.92**
		LOO	84.96	0.86	4.11	89.48	0.82	3.91	90.78	0.81	3.85
		VG	86.25	0.85	3.72	90.42	0.80	3.41	91.88	0.77	3.27
		SG	86.22	0.86	4.08	90.00	0.81	3.89	91.45	0.79	3.80
		IG	82.80	0.88	4.21	87.41	0.84	4.02	89.19	0.83	3.89

Table 1: Results of similarity of explanations between original instances and their adversarial examples by replacing m words for BERT and BiLSTM. *change* is defined as the percentage of positions whose corresponding ranks have changed. *spearman* is the spearman's rank order correlation between two explanations. *inte* is defined as the size of the intersection of the 5 most important tokens before and after perturbation.

More stable explanation methods could get higher explanation similarity. In specific, we employ three specific criteria including *change*, *spearman* and *inte*. *change* refers to the percentage of positions whose corresponding rank has changed, *spearman* refers to the spearman's rank order correlation efficient between the ranks of two explanations (Spearman, 1961), and *inte* refers to the size of the intersection of the 5 most important tokens before and after perturbation (Ghorbani et al., 2019). Table 1 presents the experimental results of the five explanation methods that conduted on BERT and BiLSTM on the two datasets SST-2 and IMDB.

To evaluate *stability*, following its definition, we should ensure the same output and keep semantics of adversarial examples unchanged. For output consistency, we test the consistency of predictions between all test instances and their adversarial examples, which can achieve 100%. It means our methods satisfy the requirement of the same outputs. As for input semantic consistency, we perform human evaluation to check the semantic similarity between the adversarial example and the original example. Specifically, We invite 4 postgraduates score ranges 1 to 3 according to the semantic similarity between original instances and their adversarial examples. Scores of 1,2 and 3 indicate low, medium and high semantic similarity, respectively. Higher scores mean better consistency. Table 2 shows the results of human evaluation. These results show that our generated examples could keep semantics unchanged. Therefore, our experiment satisfies the definition of *stability* and the experimental results in Table 1 are convincing.

From the experimental results in Table 1, we find the *stability* performance of the five typical explanation methods keep same on different models and different datasets. And the *stability* performance (from good to bad) of these explanation methods is as follow: **LIME**, **Integrated Gradient**, **LeaveOneOut**, **Smooth Gradient**, **Vanilla Gradient**.

According to the results for different m in Table 1, when we replace more words, explanation difference obviously increases. However, from the human evaluation results in Table 2, we find the semantic consistency also decreases as m increases. Therefore, one thing must be pointed out, to satisfy the semantic consistency of input, we should control the modification rate when we evaluate the *stability* of explanation methods.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.

Model	Dataset	Explanation	m=1	m=2	m=3
BERT	SST-2	LIME	2.75	2.48	2.23
		LOO	2.74	2.46	2.18
		VG	2.73	2.42	2.12
		SG	2.74	2.44	2.14
		IG	2.75	2.47	2.21
	IMDB	LIME	2.82	2.67	2.41
		LOO	2.79	2.63	2.36
		VG	2.77	2.60	2.34
		SG	2.77	2.61	2.33
		IG	2.80	2.65	2.39
BiLSTM	SST-2	LIME	2.76	2.48	2.25
		LOO	2.73	2.44	2.19
		VG	2.72	2.41	2.13
		SG	2.72	2.44	2.16
		IG	2.75	2.46	2.23
	IMDB	LIME	2.81	2.67	2.37
		LOO	2.75	2.47	2.18
		VG	2.74	2.44	2.15
		SG	2.74	2.46	2.16
		IG	2.75	2.50	2.22

Table 2: Results of human evaluation. The human evaluation score is not an objective metric and the higher score does not stand for the better method. We list it here just to show the adversarial examples in Table 1 keep the semantic unchanged.

5.3.2 Attack Success Rate

Secondly, following the common textual adversarial attack, We design a series of success conditions to check the attack success rate for different explanation methods. Combining with the finding in Section 5.3.1 that we should control the modification rate when evaluating *stability*, we set the maximum modification rate 20%. And existing textual adversarial attack also usually control the modification rate less than 20% (Ren et al., 2019; Alzantot et al., 2018; Zang et al., 2020).

Then we illustrate our formulated success conditions. We utilize the quantitative criteria introduced in Sec 4.1 and then define the success conditions as $d_{count} > \alpha * length$ and $d_{topk} < \beta$ for different α, β. $d_{count} > \alpha * length$ refers to the proportion of positions whose ranks have changed in should bigger than α and we select α from $\{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$. $d_{topk} < \beta$ refers to the size of intersection of the top-5 important tokens should smaller than β and we choose β from $\{1, 2, 3, 4, 5\}$. Obiviously, bigger α and smaller β mean more difficult success conditions, and a smaller attack success rate on the same condition means a more stable explanation method. Given a sentence, if achieving the success condition with the modification rate less than 20% , we define this is a successful attack. Otherwise, when the success condition can not be achieved even on the maximum modification rate, we define this is a unsuccessful attack. Then we calculate the corresponding attack success rate on all examples.

Figure 3 shows the results of BERT on SST-2. Under the two type of success conditions, we find the relative rank of the five explanation methods appears the same. And more difficult success condition would cause lower attack success rate. The *stability* performance (from good to bad) is the same as the results in §5.3.1: **LIME**, **Integrated Gradient**, **LeaveOneOut**, **Smooth Gradient**, **Vanilla Gradient**.

In summary, in our different experiment settings (Table 1 and Figure 3), all experimental results consistently show that the *stability* performance (from good to bad) of the five methods is as follows: **LIME**, **Integrated Gradient**, **LeaveOneOut**, **Smooth Gradient**, **Vanilla Gradient**. Besides, we also observe perturbation-based methods have better performance on *stability* than gradient-based methods.

6 Discussion

Beyond the above experiments, our discussions would address the following research questions:

- **RQ1** How do the evaluation results change when replacing the two steps in the proposed attack

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

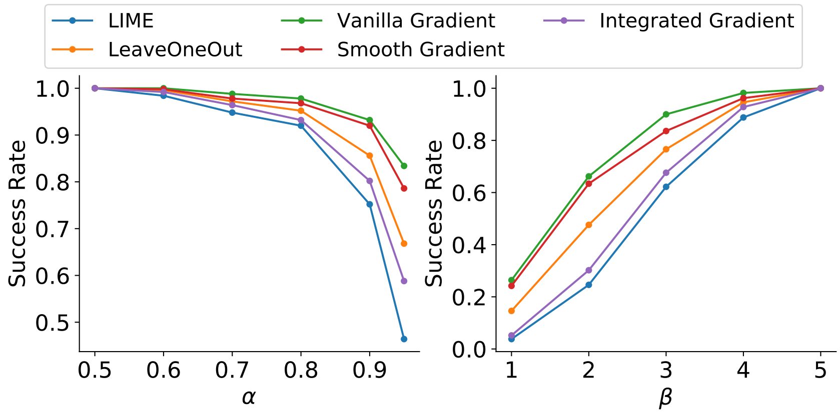

Figure 3: Success rate for different success conditions. Left part shows the condition $d_{count} > \alpha * length$ for $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$. Right part shows the condition $d_{topk} < \beta$ for $\beta \in \{1, 2, 3, 4, 5\}$. Success rate is the percentage of instances whose explanation difference could satisfy the condition. Bigger α and smaller β indicate more different explanations. A smaller success rate on the same success condition indicates a more stable method.

	m=1						m=2					
	change↓		*spearman*↑		*inte*↑		*change*↓		*spearman*↑		*inte*↑	
	ori	rand	ori	rand	ori	rand	ori	rand	ori	rand	ori	rand
LIME	**79.87**	**76.00**	**0.80**	**0.84**	**3.87**	**4.03**	**84.03**	**82.71**	**0.78**	**0.79**	**3.81**	**3.89**
LOO	89.13	84.25	0.64	0.76	3.14	3.48	92.62	90.40	0.62	0.69	3.09	3.25
VG	92.99	89.82	0.48	0.62	2.83	3.20	95.65	94.58	0.45	0.55	2.71	2.99
SG	92.86	89.13	0.55	0.65	2,92	3.23	95.71	94.20	0.53	0.55	2.87	2.99
IG	86.79	79.39	0.71	0.80	3.45	3.89	90.01	86.12	0.69	0.75	3.38	3.69

Table 3: Results of explanation similarity for BERT on SST-2. `ori` refers to the results based on the word substitution order in §4.2.1 and `rand` refers to the results based on the random substitution order.

strategy with othe the existing methods?

- **RQ2** How can we improve the stability of explanation methods?

6.1 Correlation Analysis Between The Two Attack Steps and The Evaluation Results

To address **RQ1**, we modify the two steps in Section 4.2 to conduct experiments in the following parts:

Effect of Substitution Order To verify whether the other substitution order is effective to evaluate the *stability* of explanation methods, we utilize a random order to replace the substituion order in Section 4.2.1. Specifically, following experiments settings in Section 5.3.1, we select SST-2 and conduct experiments on BERT model. To improve efficiency, we only choose $m = 1$ and $m = 2$.

Table 3 shows the corresponding results. Compare to results in Table 1, all of the attack performance have dropped. In specific, for same explanation method on same setting, the *change* metric decreases and the $spearman$ and $inte$ metrics both increases, which stands for the higher explanation similarity. And this is consistent with the common textual adversarial attack, which has been shown the random order would much decrease the attack performance (Ren et al., 2019). Besides, we find the *stability* performance of these five explanation methods still keep same as the previous findings.

Effect of Substitution Set To verify whether the other substitution set is effective, we utilize WordNet (Miller, 1995) to construct substitution word set. We can easily find synonyms for a given word via WordNet. Following experiments settings in Section 5.3.1, we select IMDB and conduct experiments on BiLSTM model. To improve efficiency, we also only choose $m = 1$ and $m = 2$.

Similar to replacing the substitution order with random order, the attack performance also drop. And the *stability* performance of these five explanation methods also keep same.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

	m=1						m=2					
	change↓		*spearman*↑		*inte*↑		*change*↓		*spearman*↑		*inte*↑	
	ori	WN	ori	WN	ori	WN	ori	WN	ori	WN	ori	WN
LIME	**81.44**	**78.89**	**0.90**	**0.92**	**4.24**	**4.41**	**86.36**	**83.21**	**0.86**	**0.89**	**4.07**	**4.09**
LOO	84.96	82.18	0.86	0.89	4.11	4.18	89.48	86.32	0.82	0.85	3.91	3.98
VG	86.25	83.79	0.85	0.87	3.72	4.02	90.42	88.14	0.80	0.83	3.41	3.85
SG	86.22	83.72	0.86	0.87	4.08	4.14	90.00	87.97	0.81	0.84	3.89	3.95
IG	82.80	79.97	0.88	0.90	4.21	4.27	87.41	84.56	0.84	0.87	4.02	4.05

Table 4: Results of explanation similarity for BiLSTM on IMDB. `ori` refers to utilizing OpenHowNet to construct substitution set and `WN` refers to utilizing WordNet to construct substitution set.

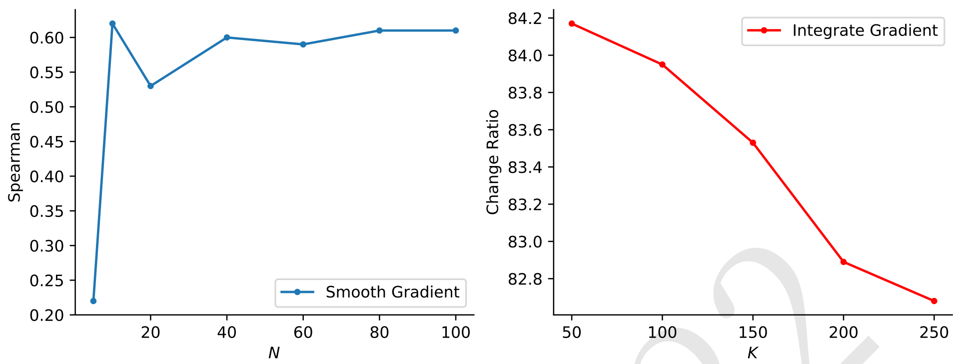

Figure 4: The left figure shows the relation between Spearman's rank order correlation and the number of the added noise M in **Smooth Gradient**. The right figure shows the relation between change ratio and the number of the divided parts K in **Integrated Gradient**.

In summary, our evaluation frame is independent to the specific substitution order and how to construct substitution set. These specific steps only influence the attack performance and could get the similar results of existing explanation methods when evaluating *stability*.

6.2 Simply Improving Stability of Explanation Method

To address **RQ2**, we try to explore how to improve the *stability* of two explanation methods.

Adding more noise We explore the influence of the number of the added noise N (Equation (9)) in Smooth Gradient. We select Spearman's rank order correlation as the evaluation metric. Figure 4 (left) shows the results. We find adding appropriate noises is useful and adding more noises is not meaningful.

More robust mechanism Integrated Gradient is a more robust mechanism compared to Simple Gradient and Smooth Gradient, because it satisfy *sensitivity* and *implementation invariance* these two important axiom (Sundararajan et al., 2017). We explore the influence of the divided parts K in Equation (11). Figure 4 (right) shows the results of change rate. We find adding the number of the divided parts K is useful. The bigger K is, the more accurate the integral value is, which means more robust mechanism. Therefore, more robust mechanism could improve the *stability* of explanation methods.

Therefore, we can try to add appropriate noises and seek more robust mechanisms to make explanation methods more stable. And we take the further exploration of improving *stability* as our future work.

7 Conclusion

This paper proposes a new evaluation frame to evaluate the *stability* of typical feature attribution explanation methods via adversarial attack. Various experimental results on different experimental settings reveal their performance on *stability*, which also show the effectiveness of our proposed evaluation frame. We also conduct experiments to show the proposed frame is dependent of specific step. Therefore, we hope the proposed evaluation frame could be applied to evaluating the *stability* of feature attribution explanation methods in the future and attract more research on this important but often overlooked property.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.

8 Limitations

The proposed evaluation frame only focus on the rank of the feature attribution explanation methods. These explanation methods also provide specific attribution scores and these scores may further refine the proposed frame.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61922085, 61831022, 61906196), the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006) and the Youth Innovation Promotion Association CAS. This work was also supported by Yunnan provincial major science and technology special plan projects, under Grant:202103AA080015.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. *arXiv preprint arXiv:2009.13295*.

Joost Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. *arXiv preprint arXiv:1905.08160*.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. *arXiv preprint arXiv:2104.05824*.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688.

Juyeon Heo, Sunghwan Joo, and Taesup Moon. 2019. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems*, pages 2925–2936.

Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. *arXiv preprint arXiv:1711.07414*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao, and Kang Liu. 2021. Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5372–5387, Online, August. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

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*, pages 142–150, Portland, Oregon, USA, June. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Qiang Dong, Maosong Sun, and Zhendong Dong. 2019. Openhownet: An open sememe-based lexical knowledge base. *arXiv preprint arXiv:1901.09957*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

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.

Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In *Human and machine learning*, pages 159–175. Springer.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

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*, pages 180–186.

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

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.

Charles Spearman. 1961. The proof and measurement of association between two things.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of nlp models is manipulable. *arXiv preprint arXiv:2010.05419*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 932-944, Nanchang, China, October 14 - 16, 2022.