IDENTIFYING THE SOURCE OF VULNERABILITY IN FRAGILE INTERPRETATIONS: A CASE STUDY IN NEURAL TEXT CLASSIFICATION

Ruixuan Tang

Charlottesville, VA

Bachelor of Engineering, University of Delaware, 2017

Master of Engineering, University of Virginia, 2018

A Thesis submitted to the Graduate Faculty

of the University of Virginia in Candidacy for the Degree of

Master of Science

Department of Computer Engineering

University of Virginia

August 2022

Tom Fletcher, Chair

Yangfeng Ji, Advisor

Jundong Li

Identifying the Source of Vulnerability in Fragile Interpretations: A Case Study in Neural Text Classification

Ruixuan Tang

(ABSTRACT)

Along with the development of machine learning models, although the prediction performance has been improved, it becomes more difficult to understand why the model makes such predictions due to the complexity of neural networks. It causes concern and interest in the trustworthiness and reliability of models. One concrete way to improve the trustworthiness and reliability of the neural model is to generate explanations that can demonstrate the relationship between the input and output prediction and help people understand the model. However, some recent observations reveal the interest and concern in the stability of the explanation methods.

Previous works (Ghorbani, Abid, and Zou 2019; Subramanya, Pillai, and Pirsiavash 2019; Xinyang Zhang et al. 2020; Lakkaraju, Arsov, and Bastani 2020) had observations on the explanation vulnerability caused by the perturbation at model input. Some of them have conflicting opinions on what source caused the explanation to be fragile and whether explanation methods are stable. This thesis explores the potential source that caused the fragile explanation. Is it caused by the neural model itself or the explanation method? I focus on a model agnostic explanation method that generates explanations from the post-hoc manner (e.g., model output probability), which is called the post-hoc explanation method. To investigate the cause of the fragile explanation, I propose a simple *output probability perturbation* method,

which adds a slight noise to the output probability of the neural model. Compared to prior perturbation methods applied at the model's inputs, the *output probability perturbation* method can circumvent the neural model's potential influence on the explanation. The proposed method is evaluated by three post-hoc explanation methods (LIME, Kernel Shapley, and Sample Shapley) and three neural network classifiers (CNN, LSTM, and BERT). The result demonstrates that the neural model is the potential primary source causing fragile explanations. Furthermore, I analyze the kernel calculation inside a post-hoc explanation method (LIME). The result demonstrates the stability of this post-hoc explanation method.

Acknowledgments

The experience in the past two and half years is a fantastic journey. I never thought I would conduct research in the machine learning field. Until Professor Ji provided me with this opportunity, I joined his Information and Language Processing (ILP) lab. Although it was not easy during this process, I noticed some interesting facts in the research work and enjoyed this work from the deep in my heart. Here, I truly appreciate every people who helped me on this journey.

First, I genuinely appreciate my advisor, Professor Yangfeng Ji. He instructed me on how to conduct a research project, how to understand and recognize the slight difference in definition, how to communicate with other researchers on an idea, and how to pursue an academic life. Meanwhile, he is a wise person to communicate with. He is always a proper person who I want to discuss, not only with academic problems but also with problems related to my daily life. He always shares his thoughts generously and supports me. I may not be able to complete the M.S. program smoothly without his assistance and instruction.

Second, I would like to thank other members of my thesis committee, Professor Tom Fletcher and Professor Jundong Li. I truly appreciate them for devoting their time to participating in my defense meeting, providing valuable feedback, and taking care of all procedures to ensure my graduation.

Third, I want to thank the entire ILP Lab group. I have the honor to be part of this group. We are not only researchers who are working in the similar field, but also friends who can offer help in both academic life and daily life. I want to thank Hanjie Chen, who was the mentor of this project, for her patience and carefulness in

assisting me in this project. I also want to thank Wanyu Du for teaching me some valuable tips in the research work.

Finally, I want to appreciate my family and friends truly. I want to thank my parents, who fully support my life and research, and care for my everyday life. I also would like to thank my life partner, Wenting. Without her encouragement, I may not be able to make the progress I make now.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to their remarkable prediction performance, neural network models have become indispensable for natural language processing (NLP) tasks. However, a neural model's evaluation should not be based only on the model's accurate prediction performance but also should be based on the reliability and trustfulness of the model's prediction results. Even though the prediction is accurate, if we cannot understand why the model generates the prediction, we cannot trust the model. One way that can be applied to improve the trustworthiness and reliability of neural models is to generate explanations that can demonstrate the relationship between the input text and the prediction result in a humanly understandable form. However, depending on the complexity of neural models, we cannot directly generate the explanations to understand how variables are combined in the model to generate predictions.

To address this issue, some explanation methods have been developed to provide insights into figuring out the relationship between input text and prediction results (Du, N. Liu, and Hu 2019). Some methods explain a black-box model from the post-hoc manner by attributing an importance score to each feature, which will be discussed detailed in the chapter 2 (Ribeiro, Singh, and Guestrin 2016b; Lundberg and Lee 2017a; Sundararajan, Taly, and Yan 2017).

Some recent works have demonstrated that explanations' instability was observed (Ghorbani, Abid, and Zou 2019; Subramanya, Pillai, and Pirsiavash 2019; Xinyang

Zhang et al. 2020; Lakkaraju, Arsov, and Bastani 2020). Based on this observation, different research groups have two different but intuitive conclusions. One conclusion believes that the observation result can be attributed to the potential instability of post-hoc explanation methods. While the other conclusion thinks that it is premature to claim if post-hoc explanation methods are stable according to this observation. Both conclusions raise the concern about assessing post-hoc explanation methods' stability. Recall that the explanations generated by a post-hoc method (Ribeiro, Singh, and Guestrin 2016a; Lundberg and Lee 2017b; Friedman 2001) depend on a black-box model's prediction probabilities. If perturbations at the input cause model prediction probabilities to change, that perturbation at the input would indirectly lead to explanation changing. It is difficult to identify the source that explanations change by only perturbing at the input of the black-box model because the major unstable component could be either the black-box model itself or the post-hoc explanation method. In other words, both the black-box model and the post-hoc explanation method are possible factors that result in unstable explanations.

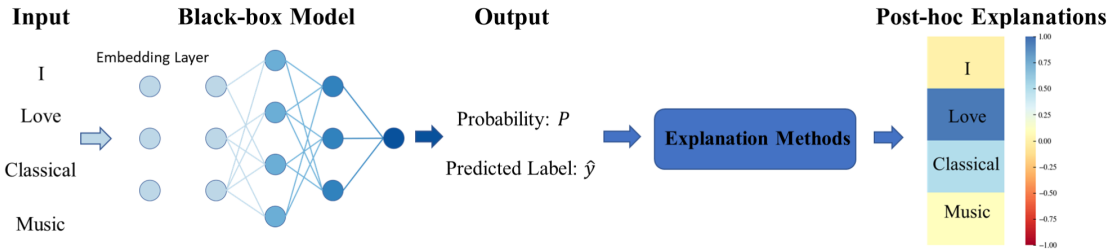

Figure 1.1: The pipeline of a simple example that post-hoc methods generate explanations with no perturbation applied.

In Figure 1.1, I demonstrate a simple example of the process that generates explanations using a post-hoc method. In Figure 1.2, I demonstrate an example of the process that generates explanations with perturbation at the input side using a post-hoc method. In this example, I apply this perturbation to the word embedding layer.

The prediction label Figure 1.1, $\hat{y}$, and the prediction label in Figure 1.2, $\bar{y}$, are equal, which is $\hat{y} = \bar{y}$. In Figure 1.1 and Figure 1.2, it can be noticed that the feature im-

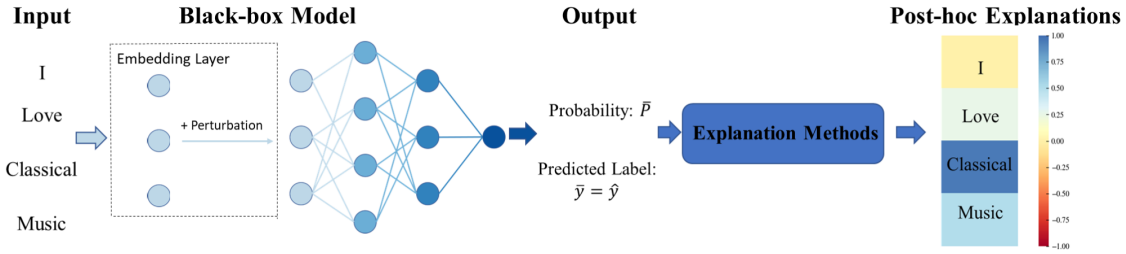

Figure 1.2: The pipeline of a simple example that post-hoc methods generate explanations with perturbation applied at the input side.

portance score of the same feature to the corresponding prediction result is changed. In general, when a feature has a higher score, it can be considered a more important explanation that supports the corresponding prediction result. For example, the word "love", with the highest feature importance score, means that it is the most relevant feature connecting to the prediction result in Figure 1.1. Similarly, the word "classical" is the most relevant feature connecting to the prediction result in Figure 1.2. Normally, the explanation to input is ranked by the features' importance score, which means the first feature is the most relevant explanation word to the result and the last feature is the least relevant explanation word.

In the examples in Figure 1.1 and Figure 1.2, when explanations are inconsistent while $\hat{y} = \bar{y}$, it is difficult to tell whether it is the model or the explanation method that causes the explanation discrepancy. One intuitive claim is that the random perturbation at the input side has to go through both the model and the post-hoc explanation method. Therefore, one possible source is the unstable black-box model. Its prediction behavior can change under the perturbation at the input side, that is, focusing on different features to make predictions. This may not tell from predicted labels if they remain unchanged but can reflect in post-hoc explanations, i.e., expla-

nations discrepancy (H. Chen and Ji 2020). Another possible source is the post-hoc explanation methods if the mechanism is vulnerable to perturbation at the input side.

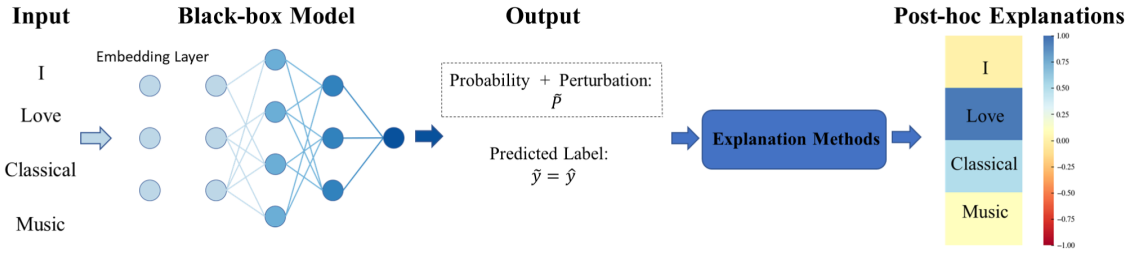

Figure 1.3: The pipeline of a simple example that post-hoc methods generate explanations with perturbation applied at the output probability (proposed method).

In this thesis, I propose a simple strategy to demonstrate the potential primary source causing fragile explanations as observed. To circumvent the potential influence on explanations from the model, instead of applying perturbation at the input side, I design an output probability perturbation method by slightly modifying the prediction probabilities of a black-box model, as shown in Figure 1.3. Since this method directly perturbs the black-box model's output probability, it isolates the potential influence caused by the model's instability to explanations. If similar fragile explanations can be observed when only output probability perturbation is applied, it would suggest that post-hoc explanation methods may be unstable because the potential influence from the black-box model is blocked. Post-hoc explanation methods can be attributed to one of the sources that caused the explanation discrepancy in previous observations. Otherwise, we should not blame post-hoc explanation methods as unstable and contributing to explanation discrepancy in the observation of the prior works.

In chapter 2, I will introduce the proposed method in detail. In chapter 3, I will discuss a comparison experiment on the explanation discrepancy, including experiment setup, results, and discussion on an analysis of the kernel calculation inside the LIME

algorithm. In chapter 4, I will introduce some important previous relevant works related to this thesis.

Chapter 2

Output Probability Perturbation Method

In this chapter, I will demonstrate the proposed method in detail. I will introduce some important related works in chapter 4.

2.1 Background

For a text classification task, $\boldsymbol{x}$ denotes the input text consisting of N words, $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}]$, with each component $\boldsymbol{x}^{(n)} \in R^d$ representing the n-th word embedding. We define a black-box classifier as $f(\cdot)$ and the neural network model output probability of a given $\boldsymbol{x}$ on the corresponding label k is $P(y = k \mid \boldsymbol{x}) = f_k(\boldsymbol{x})$, where $k \in \{1, \ldots, C\}$ and C is the total number of label classes.

To explain a black-box model's prediction $\hat{y} = f(\boldsymbol{x})$, a class of post-hoc explanation methods approximate the model locally via additive feature attributions (Lundberg and Lee 2017a; Ribeiro, Singh, and Guestrin 2016b; Shrikumar, Greenside, and Kundaje 2017). Specifically, these algorithms demonstrate the relationship between the input text and the prediction result by evaluating the contribution of each input feature to the model prediction result. These methods would assign a feature importance score to each input feature to represent its contribution to the prediction. Let's apply

the LIME method as an example.

Example: Post-hoc Explanation Method, LIME. It first subsamples words from the input, $\boldsymbol{x}$, to form a list of pseudo examples $\{\boldsymbol{z}_{j=1}^{L}\}$, and then the contributions of input features are estimated by a linear approximation $f_{\hat{y}}(\boldsymbol{r}) \approx g_{\hat{y}}(\boldsymbol{r'})$, where $\boldsymbol{r} \in \{\boldsymbol{x}, \boldsymbol{z}_{j=1}^{L}\}$, $g_{\hat{y}}(\boldsymbol{r'}) = \boldsymbol{w}_{\hat{y}}^{T}\boldsymbol{r'}$, and $\boldsymbol{r'}$ is a simple representation of $\boldsymbol{r}$, e.g. bag-of-words representation. The weights $\{w_{\hat{y}}^{(n)}\}$ represent importance scores of input features $\{\boldsymbol{x}^{(n)}\}$. Let $I(\boldsymbol{x}, \hat{y}, P)$ denote the explanation for the model prediction on $\boldsymbol{x}$, where $\hat{y}$ is the predicted label and P represents output probabilities. For input perturbations, previous work claimed that explanation methods are fragile if $I(\boldsymbol{x}, \hat{y}, P)$ and $I(\bar{\boldsymbol{x}}, \hat{y}, \bar{P})$ are discrepant, while model predicitons on the original input $\boldsymbol{x}$ and the perturbed input $\bar{\boldsymbol{x}}$ are the same (Ghorbani, Abid, and Zou 2019; Subramanya, Pillai, and Pirsiavash 2019; Sinha et al. 2021). However, with different inputs, the model may behave differently, which would reflect in post-hoc explanation, though it may not be in prediction labels. It is not reasonable to blame explanation vulnerability without excluding the effect of model robustness on explanation discrepancy. To circumvent the model prediction behavior change on different inputs, I propose a method to explore the source of fragile explanation by adding output perturbations.

2.2 Explanation Discrepancy

The basic idea of this thesis is to examine the source that caused the explanation discrepancy due to the perturbation at the model input. Recall the previous section, $I(\boldsymbol{x}, \hat{y}, P)$ and $I(\bar{\boldsymbol{x}}, \hat{y}, \bar{P})$ denote the explanation to the model prediction based on the original input $\boldsymbol{x}$ and the perturbed input $\bar{\boldsymbol{x}}$ respectively, where $\bar{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon$, ε is

the perturbation at input. Similarly, we define $I(\boldsymbol{x}, \hat{y}, \widetilde{P})$ as the explanation to the prediction based on the perturbed output probability $\widetilde{P} = P + \varepsilon'$, ε' is the perturbation on output probability. Note that when ε and ε' are small, the model prediction $\hat{y}$ would not change. Especially, ε' is a slight perturbation, that would not make $\widetilde{P}$ to a value lower than 0.

The explanation discrepancy between $I(\bar{\boldsymbol{x}}, \hat{y}, \bar{P})$ and $I(\boldsymbol{x}, \hat{y}, P)$ is denoted as δ_{input}, and the discrepancy between $I(\boldsymbol{x}, \hat{y}, \widetilde{P})$ and $I(\boldsymbol{x}, \hat{y}, P)$ is denoted as δ_{output}. Here, I use figures in chapter 1 as an example to illustrate the explanation discrepancy in detail.

As introduced, the explanation, $I(\boldsymbol{x}, \hat{y}, P)$, in Figure 1.1 is "Love", "Classical", "I" and "Music", with the order. The explanation, $I(\bar{\boldsymbol{x}}, \hat{y}, \bar{P})$, in Figure 1.2 is "Classical", "Music", "Love", and "I", with the order. The explanation, $I(\boldsymbol{x}, \hat{y}, \widetilde{P})$, in Figure 1.3 is "Love", "Classical", "I" and "Music", with the order. Generally, after perturbed, explanation inconsistency reflects in two aspects. The first aspect is the difference in the same feature's importance score. For example, the importance score for the word "I" may differ in Figure 1.1 and Figure 1.2. The difference is considered δ_{input}. The second aspect is whether the top K important features in the explanation are consistent. For example, if $K = 2$, the first two important words in Figure 1.1 is "Love", and "Classical" and the first two important words in Figure 1.2 is "Classical", and "Music". The difference between these is considered δ_{input} as well. Similarly, δ_{output} can describe the same aspects of explanation discrepancy in Figure 1.1 and Figure 1.3.

2.3 The Proposed Method

Recall the limitation of the method that perturbation is applied at the input side; it is difficult to identify the primary source causing explanation discrepancy. The purpose of the output probability perturbation method is to evaluate the explanation discrepancy without the possible influence of black-box models. For the output probability perturbation, given an example $\boldsymbol{x}$, I add a small perturbation to model output probabilities $\{P(y = k \mid \boldsymbol{x}) + \varepsilon'_{y=k}\}_{k=1}^{C}$, where each $\varepsilon'_{y=k} \sim \mathcal{N}(0, \sigma^2)$, σ^2 is the variance of Gaussian Distribution.

To guarantee the modified $\{P(y = k \mid \boldsymbol{x}) + \varepsilon'_{y=k}\}_{k=1}^{C}$ are still legitimate probabilities, I further normalize them as

$$\widetilde{P}(y = k \mid \boldsymbol{x}) = \frac{P(y = k \mid \boldsymbol{x}) + \varepsilon'_{y=k}}{\sum_{i=1}^{C}\{P(y = i \mid \boldsymbol{x}) + \varepsilon'_{y=i}\}} \tag{2.1}$$

Similar to finding an explanation to the original prediction, the explanation to the prediction with output probability perturbation is computed based on $\widetilde{P}(y = \hat{y} \mid \boldsymbol{x})$. The output probability perturbation idea well suits the motivation of investigating the primary source causing explanation changes. The main reason is that, unlike perturbation applied at the input side, the proposed method avoids the potential effects of the model vulnerability to explanations. I apply the LIME algorithm as an example to demonstrate the proposed method.

Example: Output probability perturbation in LIME. To give an example of the proposed perturbation idea, I would like to represent $\boldsymbol{r'}$ as the bag-of-words representations of the original input texts. A simplified version [1] of LIME is equivalent

[1] Without the example weight computed from a kernel function and the regularization term of explanation complexity.

to finding a solution of the following linear equation:

$$\boldsymbol{w}_{\hat{y}}^T \boldsymbol{r'} = \widetilde{\boldsymbol{p}}_{\hat{y}} \tag{2.2}$$

where $\widetilde{\boldsymbol{p}}_{\hat{y}} = [\widetilde{P}(y = \hat{y} \mid \boldsymbol{x}), \widetilde{P}(y = \hat{y} \mid \boldsymbol{z}_1), \ldots, \widetilde{P}(y = \hat{y} \mid \boldsymbol{z}_L)]^T$ are the perturbed probabilities on the label $\hat{y}$, and $\boldsymbol{w}_{\hat{y}}^T$ is the weight vector, where each element measures the contribution of an input word to the prediction $\hat{y}$. A typical explanation from LIME consists of top important words according to $\boldsymbol{w}_{\hat{y}}$. Essentially, the proposed output perturbation is similar to the perturbation analysis in linear systems (Golub and Van Loan 2013), which aims to identify the stability of these systems. Despite the simple formulation in Equation 2.2, a similar linear system can also be used to explain the Shapley-based explanation methods (e.g., Sample Shapley (Strumbelj and Kononenko 2010)).

Chapter 3

Experiments and Discussion

In this chapter, I will introduce the experiment setup, comparison experiment, and evaluation result. Furthermore, according to the result observed in the comparison experiment, I will further analyze the kernel calculation inside the LIME algorithm, including results from a simulation experiment in section 3.4.

3.1 Experiment Setup

Datasets. I adopt four text classification datasets: IMDB movie reviews dataset (Maas et al. 2011, IMDB), AG's news dataset (Xiang Zhang, J. Zhao, and LeCun 2015, AG's News), Stanford Sentiment Treebank dataset with binary labels (Socher et al. 2013, SST-2), and 6-class questions classification dataset TREC (X. Li and Roth 2002, TREC). The summary statistics of datasets are shown in Table 3.1.

Dataset	C	L	*#train*	*#dev*	*#test*	vocab	threshold	length
IMDB	2	268	20K	5K	25K	29571	5	250
SST-2	2	19	6920	872	1821	16190	0	50
AG's News	4	32	114K	6K	7.6K	21838	5	50
TREC	6	10	5000	452	500	8026	0	15

Table 3.1: Summary statistics for the datasets where C is the number of classes, L is the average sentence length, # counts the number of examples in train/dev/test sets, vocab is the vocab size, and the threshold is the low-frequency threshold, and length is mini-batch sentence length.

Models. I apply three neural network models, Convolutional Neural Network (Kim 2014, CNN), Long Short Term Memory network (Hochreiter and Schmidhuber 1997, LSTM), and Bidirectional Encoder Representations from Transformers (Devlin et al. 2018, BERT).

Convolutional Neural Network, CNN, is one of the most widely used deep learning architectures. Although it has a supremacy position in the field of vision, the use of CNN in other fields (e.g., natural language processing) is increasing rapidly. In simple, the principle of CNN is based on information processing in the visual system of humans. The core characteristics are that it can efficiently decrease the dimension of input, and it can efficiently retain important features of the input.

Long Short Term Memory Neural Network, LSTM, is another widely used deep learning algorithm, which is one kind of the advanced, recurrent neural network, RNN. Unlike the architecture of a standard feedforward deep learning neural network, it has feedback connections in the architecture, which helps to process sequential data (e.g., language and speech). LSTM can reduce long-term dependencies problems and vanishing gradient problems, that are caused by standard RNN.

Bidirectional Encoder Representations from Transformers, BERT, is a Language Model (LM), which is released by the Google company in 2018 (Devlin et al. 2018, BERT). After it was released, because of its superior performance in many NLP tasks, BERT became the most popular deep learning architecture. In the Natural Language Processing field, the main tasks of the BERT model are (1) Sentence pairs classification tasks and (2) Single sentence classification tasks. In this thesis, I apply the BERT model focusing on the second task.

Post-hoc Explanation Methods. I adopt three post-hoc explanation methods, Local Interpretable Model-Agnostic Explanations (Ribeiro, Singh, and Guestrin 2016b, LIME), Kernel Shapley (Lundberg and Lee 2017b), and Sample Shapley (Strumbelj and Kononenko 2010). LIME, Kernel Shapley, and Sample Shapley are additive feature attribution methods. The additive feature method provides a feature importance score on every feature for each text input based on model prediction.

LIME and Kernel Shapley are two post-hoc methods adopting a similar intuitive strategy. The first step is to generate a new dataset by sub-sampling based on the original dataset and the corresponding prediction result based on the black-box model. The second step is to train an explainable machine learning model (eg: linear regression, LASSO) with the new sub-sampling dataset. The difference between the LIME algorithm and the Kernel Shapley algorithm is in the way to calculate the weight of sub-sampling data in the explainable model. LIME algorithm relies on the distance between the original data and the sub-sampling data. Kernel Shapley algorithm relies on the Sharpley value estimation.

Sample Shapley is a post-hoc method based on another intuitive strategy. It evaluates how much contribution of each input feature to the prediction result (based on the Sharpley value estimation), which is another way to explain the model's prediction.

Evaluation Metrics. In the experiment, I apply two evaluation metrics, Kendall's Tau order rank correlation score, and the Top-K important words overlap score.

As illustrated in section 2.2, explanation discrepancy reflects in two aspects intuitively. I use Kendall's Tau order rank correlation score to evaluate the explanation discrepancy on the same feature. Kendall's Tau order rank correlation score can be applied to evaluate the correlation between two values in the same position. If the

score in the same position is close to 1, then the two values do not differ significantly. If the score in the same position is close to -1, then the two values differ significantly. I use Top-K important words overlap score to evaluate the discrepancy on the first K number features in the ordered explanation on the same input. The counting method is to evaluate the overlap rate of the first K number features in the ordered explanation. In this thesis, I set $K = 5$.

3.2 Explanation Discrepancy Comparison Experiment

To explore the primary source causing explanation vulnerability, I conduct a comparison experiment to evaluate and compare between explanation discrepancy δ_1, and explanation discrepancy δ_2. The definition of δ_1 and δ_2 are introduced in chapter 2. δ_1 represents discrepancy of the explanation generated by the black-box model with no perturbation, $I(\boldsymbol{x}, \hat{y}, P)$, and the explanation generated by the black-box model with perturbation at the input, $I(\bar{\boldsymbol{x}}, \hat{y}, \bar{P})$. While δ_2 represents the discrepancy of $I(\boldsymbol{x}, \hat{y}, P)$ and the explanation generated by the black-box model with perturbation at the output probability, $I(\bar{\boldsymbol{x}}, \hat{y}, \widetilde{P})$.

To ensure the model performance is acceptable, the prediction accuracy for all models and datasets applied is recorded in Table 3.2.

In this experiment, I directly add random noise to the word embedding layer, as the way to add perturbation at the input (X. Liu et al. 2020). In comparison, for adding perturbation at the output, I directly add random noise to the output probabilities.

I apply the Gaussian Distribution to generate perturbation values, which has two

Dataset	CNN	LSTM	BERT
IMDB	86.30%	86.60%	90.80%
SST-2	82.48%	80.83%	91.82%
AG's News	89.90%	88.90%	95.10%
TREC	92.41%	90.80%	97.00%

Table 3.2: Prediction accuracy(%) four datasets (IMDB, SST-2, AG's News and TREC) on three neural network models (CNN, LSTM and BERT).

preponderance. First, I can control the perturbation level by modifying the variance of Gaussian Distribution σ^2, which is intuitive and simple. Second, directly applying the Gaussian Distribution generated perturbation at the word embedding layer, can not only perturb at the input, but also it would not cause potential changes on input features. To generate a generalized evaluation result, I applied five different perturbation levels. "0" means the slightest perturbation level, zero perturbation, while "4" represents the strongest perturbation level.

Detailed values of perturbation levels, which is the variance of Gaussian Distribution, at both input side and output probability are shown in Table 3.3, 3.4, 3.5, and 3.6.

Dataset	Perturbation	Level	Model		
			CNN	LSTM	BERT
IMDB	Input Perturbation (σ^2_{input})	0	0	0	0
		1	0.05	0.05	0.05
		2	0.1	0.1	0.1
		3	0.15	0.15	0.15
		4	0.2	0.2	0.2
	Output Perturbation (σ^2_{output})	0	0	0	0
		1	0.25	0.25	0.25
		2	0.5	0.5	0.5
		3	0.75	0.75	0.75
		4	1	1	1

Table 3.3: Perturbation levels applied on IMDB dataset

Dataset	**Perturbation**	**Level**	**Model**		
			CNN	LSTM	BERT
AG's News	Input Perturbation (σ^2_{input})	0	0	0	0
		1	0.05	0.05	0.05
		2	0.1	0.1	0.1
		3	0.15	0.15	0.15
		4	0.2	0.2	0.2
	Output Perturbation (σ^2_{output})	0	0	0	0
		1	0.25	0.25	0.25
		2	0.5	0.5	0.5
		3	0.75	0.75	0.75
		4	1	1	1

Table 3.4: Perturbation levels applied on AG's News dataset

Dataset	**Perturbation**	**Level**	**Model**		
			CNN	LSTM	BERT
SST-2	Input Perturbation (σ^2_{input})	0	0	0	0
		1	0.05	0.05	0.05
		2	0.1	0.1	0.1
		3	0.15	0.15	0.15
		4	0.2	0.2	0.2
	Output Perturbation (σ^2_{output})	0	0	0	0
		1	0.25	0.25	0.25
		2	0.5	0.5	0.5
		3	0.75	0.75	0.75
		4	1	1	1

Table 3.5: Perturbation levels applied on SST-2 dataset

Dataset	Perturbation	Level	Model		
			CNN	LSTM	BERT
TREC	Input Perturbation (σ^2_{input})	0	0	0	0
		1	0.05	0.05	0.05
		2	0.1	0.1	0.1
		3	0.15	0.15	0.15
		4	0.2	0.2	0.2
	Output Perturbation (σ^2_{output})	0	0	0	0
		1	0.25	0.25	0.25
		2	0.5	0.5	0.5
		3	0.75	0.75	0.75
		4	1	1	1

Table 3.6: Perturbation levels applied on TREC dataset

3.3 Results and Discussion

Plots of results of the comparison experiment on four datasets are shown in Figure 3.1, 3.2, 3.3 and 3.4.

I apply analysis on the IMDB dataset result, shown in Figure 3.1, as an example. Results shown in other plots have a similar tendency. Kendall's Tau order rank correlation score plots are shown in Figure 3.1(a), (c) and (e). Top-K important words overlap score plots are shown in Figure 3.1 (b), (d) and (f). Figure 3.1 (a) and (b) reflects the results on the LIME method. Figure 3.1 (c) and (d) reflects the results on the Kernel Shapley method. Figure 3.1 (e) and (f) reflects the results on the Sample Shapley method.

Kendall's Tau order rank correlation score results. Kendall's Tau order rank correlation score results indirectly illustrate the stability of post-hoc explanation methods. Furthermore, previous observation on the explanation difference can be attributed to the potential influence caused by the black-box model. In Figure 3.1

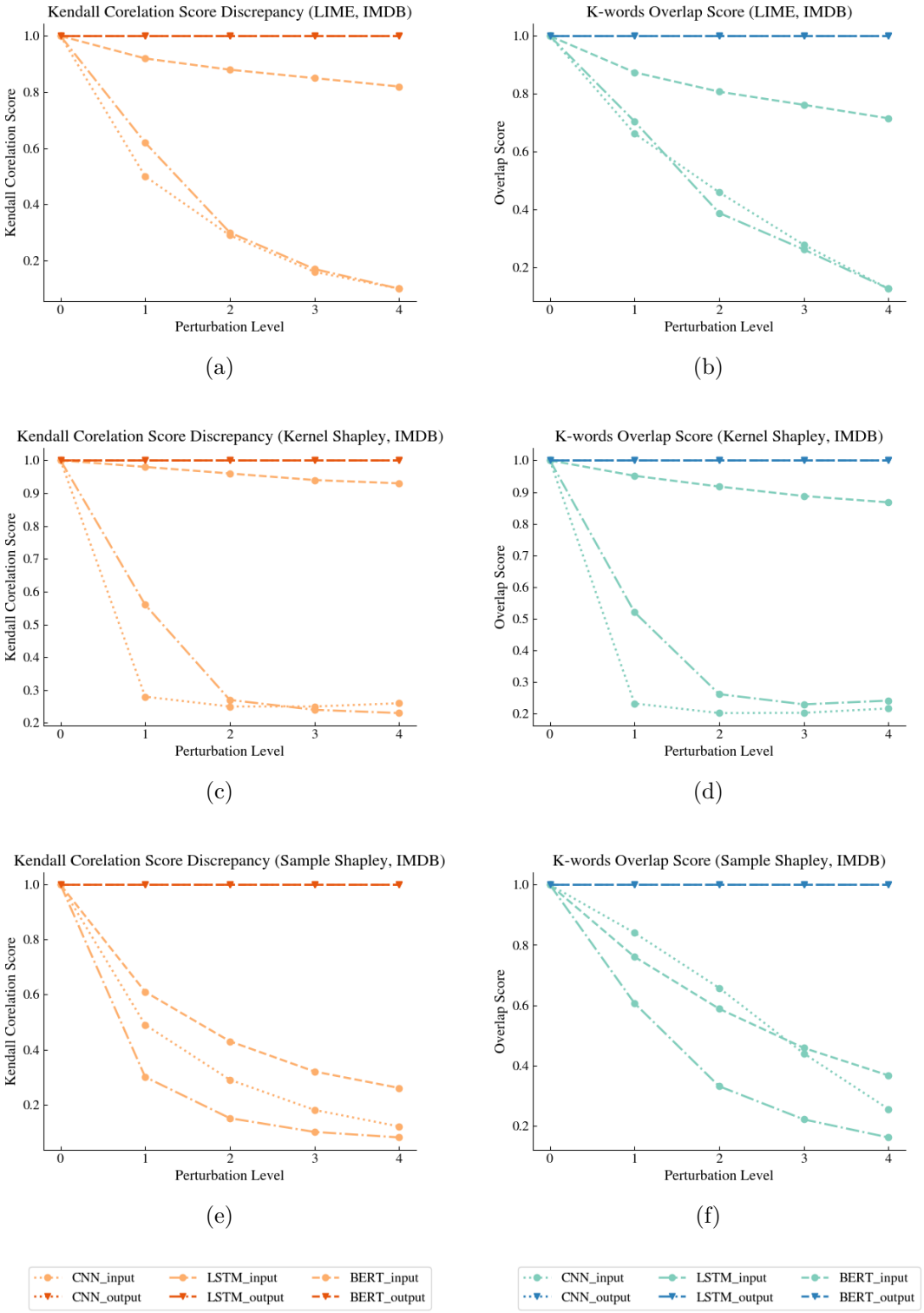

Figure 3.1: Comparison experiment results on IMDB dataset

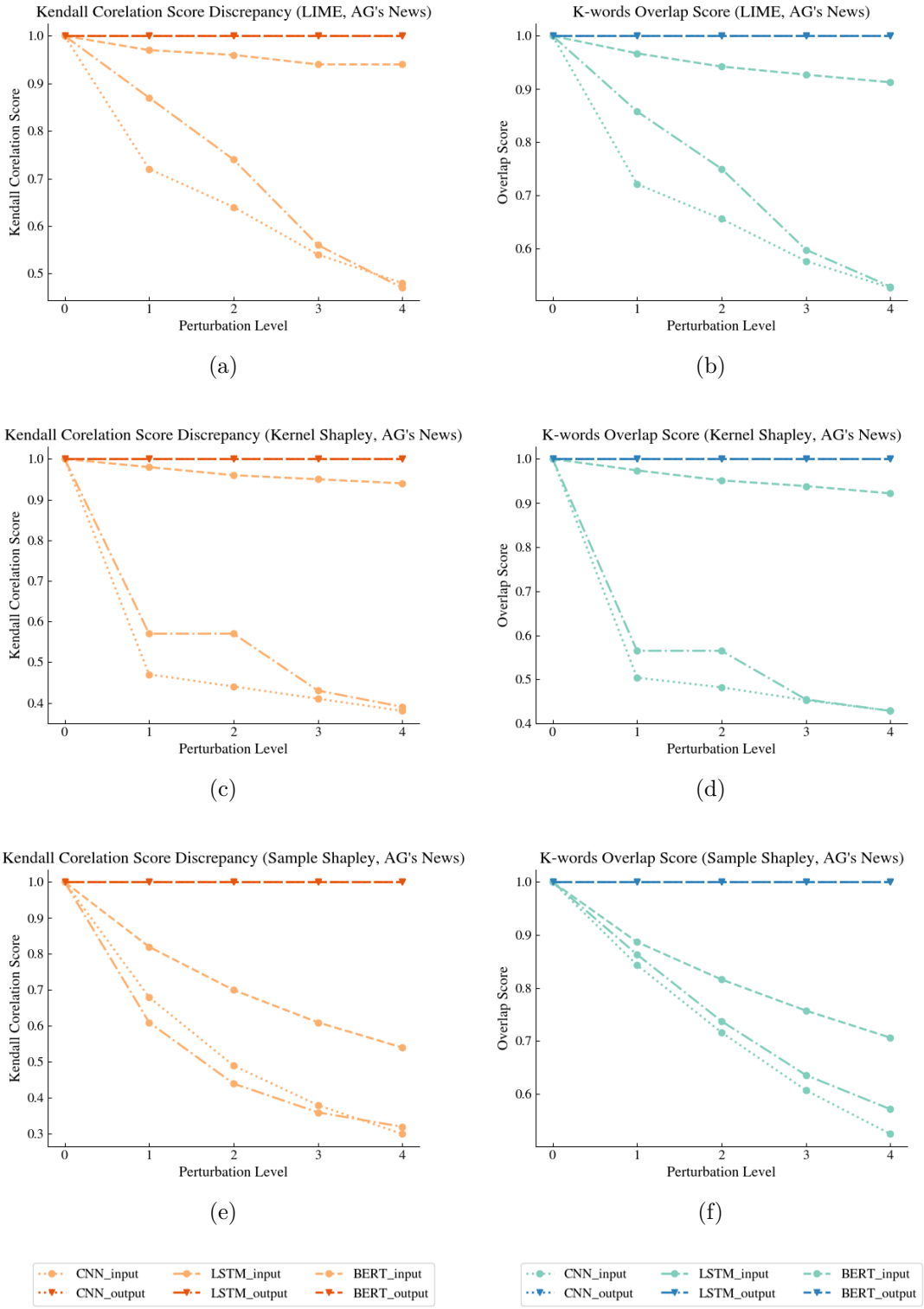

Figure 3.2: Comparison experiment results on AG's News dataset

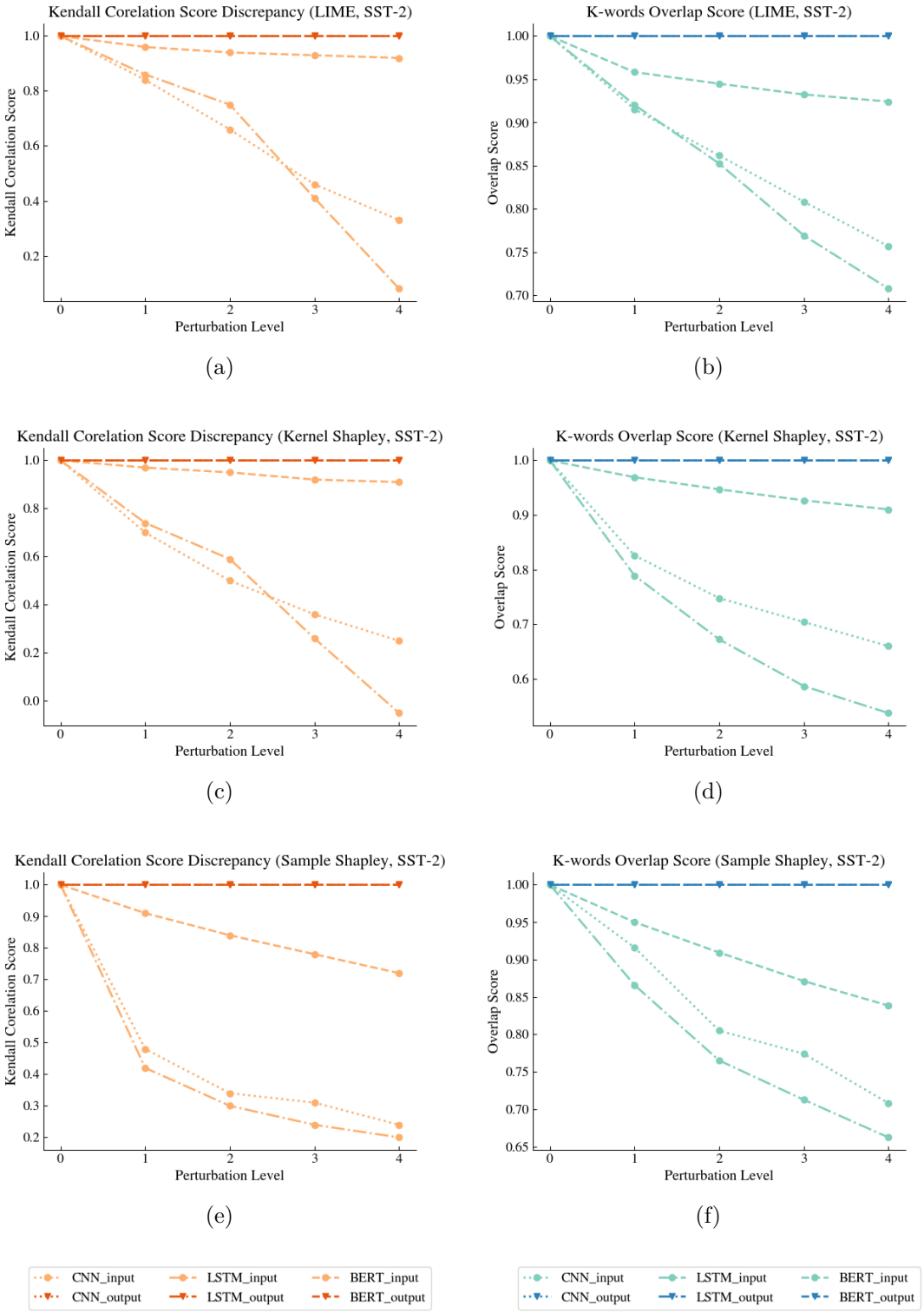

Figure 3.3: Comparison experiment results on SST-2 dataset

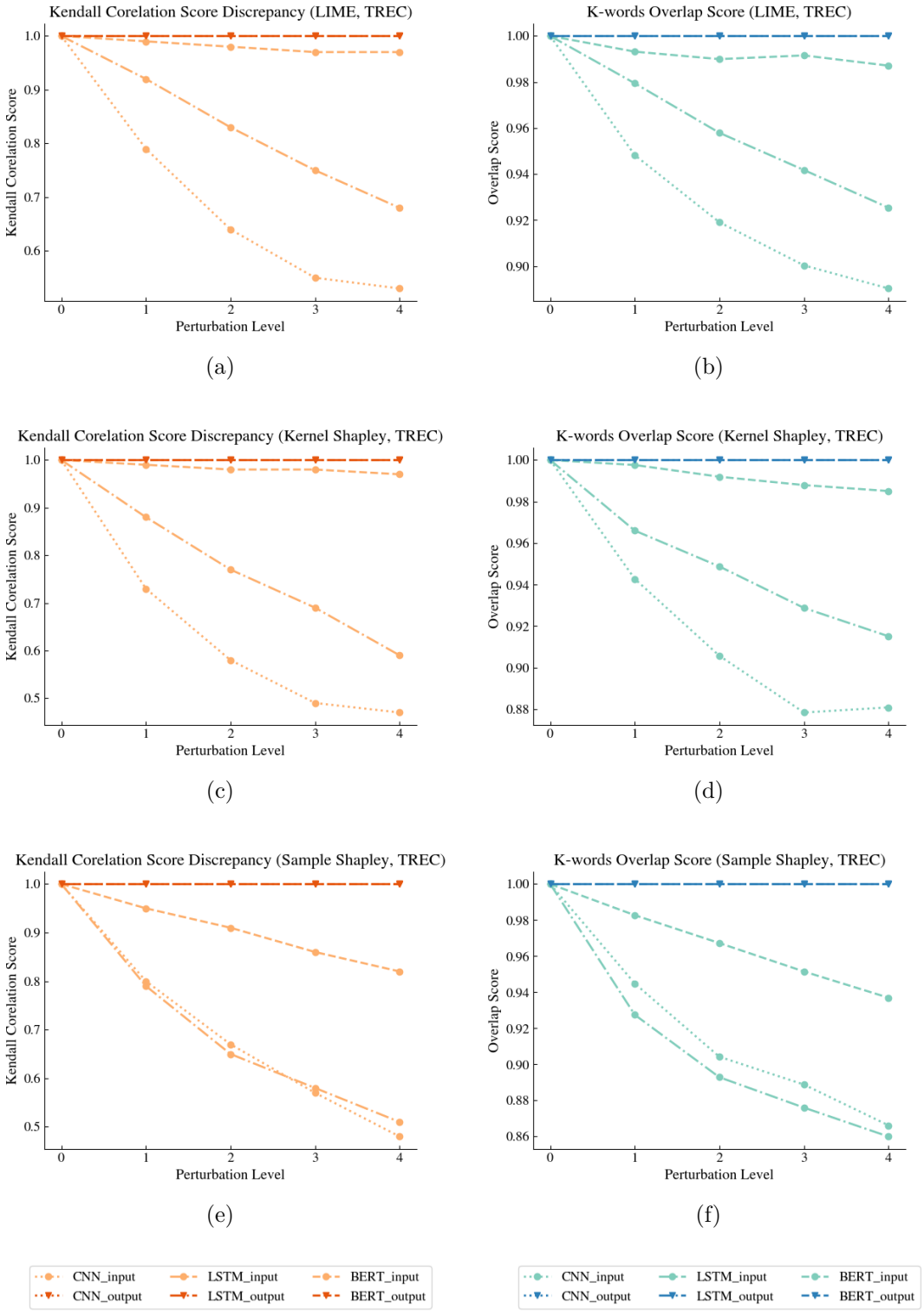

Figure 3.4: Comparison experiment results on TREC dataset

(a), (c) and (e), existing apparent discrepancies between δ_1 against δ_2 in all three post-hoc explanation methods, LIME, Kernel Shapley and Sample Shapley. For output probability perturbation method results, it is noticeable that the values of Kendall's Tau order rank correlation scores remain the same with the increased perturbation levels from "0" to "4". It indicates that the feature importance score remains the same for every input word, although output probability perturbation is applied. Furthermore, the result suggests that for a given input, if $\boldsymbol{x}$ and $\hat{y}$ stay unchanged, the only perturbation ε' at output probability is unlikely to influence explanations generated by post-hoc methods. In other words, the explanation of vulnerability observed in the previous study is unlikely caused by post-hoc methods.

Meanwhile, for the baseline results (perturbation applied at the input), it is notifiable that the values of Kendall's Tau order rank correlation scores decrease obviously with the increase of input perturbation intensity levels. Except for some datasets using the BERT model, it decreased slightly. It means that the black-box model becomes more vulnerable to the perturbation at the input, which causes fragile explanations. Compared to the previous result, the black-box model is more likely to be the primary source causing fragile explanation as prior works observed.

Top-K word importance score results. Top-K word importance score plots reflect the same result: the black-box model is the primary source causing explanation difference. In Figure 3.1 (b), (d) and (f), δ_1 against δ_2 displays an obvious discrepancy in three post-hoc explanation methods as well. For output probability method results, δ_2 reveals no change in the overlap score of the K most important words. It means that, for the top five important features to each corresponding prediction result, output probability perturbations cannot cause the difference. It is a corresponding

result due to the previous evaluation metrics result. Because the feature importance score does not change for each input feature, the overlap rate would remain the same as well. Moreover, the result in this metric indicates that the black-box model is more likely to cause explanation vulnerability than explanation methods.

3.4 Further Analysis on the LIME Algorithm

3.4.1 Analysis

According to the results of comparison experiments, one indirect conclusion is that the post-hoc explanation methods are likely stable. Although we can conclude the potential primary source that caused explanations fragile in the previous work is the black-box model, it may be premature to claim the post-hoc methods are stable. After reviewing the work of LIME (Ribeiro, Singh, and Guestrin 2016a) and Kernel Shapley (Lundberg and Lee 2017b), although they apply different kernel functions to calculate the feature importance score, both methods apply the same processing strategy, which I have introduced in section 3.1. I will use LIME as an example to explore the stability of post-hoc methods.

$$L(f, g, \pi) = \sum_{z, z' \in Z} \pi(f(z) - g(z')) \tag{3.1}$$

Equation 3.1 is definition of the loss function (Ribeiro, Singh, and Guestrin 2016a). According to the loss function equation, I define the kernel calculation of the LIME algorithm, $\pi g(z')$. $g(z')$ represents the explainable model on the sub-sampling dataset, z', in the post-hoc algorithm. Here, I use a linear model to represent this explainable

model, $g(z') = Wz'$. W is the weight function of the linear model and the importance feature score function as well. Equation 3.2 is the kernel calculation equation of the explanation method after expanding.

$$G = \pi W z' = \pi z' W \tag{3.2}$$

The form of the kernel calculation equation can be interpreted as a general linear function. Recall a general form of a linear equation, $Ax = b$. According to (I. Goodfellow, Bengio, and Courville 2016), if the condition number, (κ), which is the largest eigenvalue in the matrix A divided by the smallest eigenvalue in the matrix A, is large, the solution x would change rapidly by a slight difference in b, which would cause sensitivity of the solution to the slight error in the input. In Equation 3.2, $\pi z'$ represents the matrix A, and the feature importance score function W represents the solution x. If $\pi z'$ is a stable linear system with a minor condition number, the feature importance score function W would be unlikely sensitive to the minor error, and the corresponding post-hoc explanation method is stable.

In the next section, I apply the LIME algorithm as an example to evaluate its condition number in the linear system $\pi z'$. Since the kernel calculation part is a pure numerical step without semantics involved. I conduct a simulation experiment to explore the stability of the post-hoc explanation method.

3.4.2 Simulation Experiment

For the simulation experiment, I simulate z' as a matrix with the size of sub-sampling size, m, multiple the length, l, of input text. For the sample size, I select $m = 200$,

which is the actual value I applied in the comparison experiment, in section 3.2. For the sentence length, I firstly simulated the case when sentence length, $l = 20$. Then I compare condition number distribution when sentence lengths are different. I apply two other cases, $l = 30$, and $l = 40$. For each length, I simulated it for 500 iterations.

For π, it is the kernel function in LIME, which is the distance between the original input to the sub-sampling based on the original input. In the simulation experiment, I apply cosine distance to represent the value of π.

Total iteration number	$\kappa \in [5, 6)$	$\kappa \in [6, 7)$
500	392	108

Table 3.7: Condition number (κ) distribution when $l = 20$

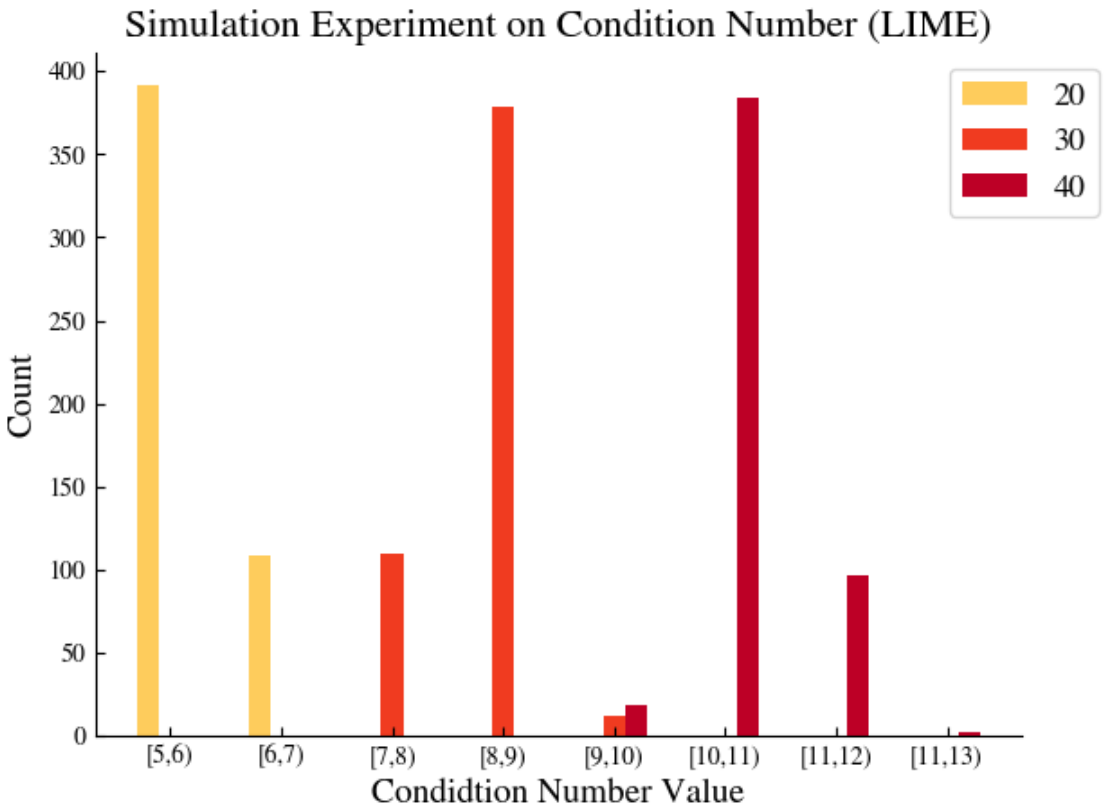

Figure 3.5: Simulation Experiment Result.

In Table 3.7, the result of the simulation experiment when the sentence length is 20, demonstrates that the majority of condition number of the matrix $\pi z'$ is lower than 7. In Goldstein's work, it suggests that the condition number of a stable or a well-conditioning matrix should be lower than 30 (Goldstein 1993). It means that the feature importance score function, W, is less likely influenced by a small perturbation involved, which reflects in the results in the comparison experiment in section 3.2. In Figure 3.5, the result of the further simulation experiment shows that when the length of the sentence increases, the condition number of the matrix $\pi z'$ increases with a tiny amplitude. The majority of the condition number of the matrix is lower than 13 when the length is from 20 to 40. Although the result demonstrates that the condition number would increase with sentence length increasing, the increasing ampitude is small and the majority of the condition number is lower than the threshhold number. The result suggests that the matrix $\pi z'$ in the LIME algorithm can remain a small condition number, which makes the system relatively stable. In other words, the LIME algorithm is a relatively stable post-hoc explanation method.

Chapter 4

Previous Relevant Works

4.1 Post-hoc Explanation Methods

Most work focuses on explaining neural network models in a post-hoc manner, especially generating a local explanation for each model prediction. The white-box explanation methods, such as gradient-based explanations (Hechtlinger 2016) and attention-based explanations (Ghaeini, Fern, and Tadepalli 2018), either require additional information (e.g. gradients) from the model or incur much debates regarding their faithfulness to model predictions (Jain and B. C. Wallace 2019).

Another line of work focuses on explaining black-box models in a model-agnostic way. Jiwei Li, Monroe, and Jurafsky (2016) proposed a perturbation-based explanation method, Leave-one-out, that attributes feature importance to model predictions by erasing input features one by one. Ribeiro, Singh, and Guestrin (2016b) proposed to estimate feature contributions locally via linear approximation based on pseudo examples. Some other works proposed the variants of the Shapley value (Shapley 1953) to measure feature importance, such as the Sample Shapley method (Strumbelj and Kononenko 2010), the Kernel Shapley method (Lundberg and Lee 2017b), and the L/C-Shapley method(J. Chen et al. 2018).

In this project, I focus on two well-adopted post-hoc explanation methods on the

black-box model, the LIME method(Ribeiro, Singh, and Guestrin 2016b) and the Sample Shapley method (Strumbelj and Kononenko 2010).

4.2 Model Robustness

Recent works have shown the vulnerability of model due to adversarial attacks (Szegedy et al. 2013; I. J. Goodfellow, Shlens, and Szegedy 2014; Z. Zhao, Dua, and Singh 2017). Some adversarial examples are similar to original examples but can quickly flip model predictions, which causes concern on model robustness (Jia et al. 2019). In the text domain, a common way to generate adversarial examples is by heuristically manipulating the input text, such as replacing words with their synonyms (Alzantot et al. 2018; Ren et al. 2019; Jin et al. 2020), misspelling words (Jinfeng Li et al. 2018; Gao et al. 2018), inserting/removing words (Liang et al. 2017), or concatenating triggers (E. Wallace et al. 2019). Unlike these works that modify original input texts, I add noise at model outputs and disentangle explanation robustness from model robustness.

4.3 Explanations Robustness

Previous work explored explanation robustness by either perturbing the inputs (Ghorbani, Abid, and Zou 2019; Subramanya, Pillai, and Pirsiavash 2019; Xinyang Zhang et al. 2020; Heo, Joo, and Moon 2019) or manipulating the model (Wang et al. 2020; Slack et al. 2020; Zafar et al. 2021). For example, Slack's group fooled post-hoc explanation methods by hiding the bias for black-box models based on the proposed novel scaffolding technique (Slack et al. 2020). However, all of these works cannot disen-

tangle the sources that cause fragile explanation. Differently, the proposed method mitigates the influence of model to the explanation by perturbing model outputs.

Chapter 5

Conclusion

In this thesis, my main contribution is to identify the primary source of fragile explanation, where I propose an output probability perturbation method that slightly modifies the prediction probability of black-box models. With the help of this proposed method, some observation results based on the method can illustrate a conclusion that the primary potential source that caused fragile explanations in the previous studies is the black-box model itself.

In chapter 2, I demonstrate this proposed method in details and its advantages compared to the previous methods. In chapter 3, I explore this method with a comparison experiment. Furthermore, in section 3.4, I analyze the kernel calculation inside the LIME algorithm. I apply the condition number of the matrix and a simulation experiment to demonstrate that the kernel calculation matrix inside LIME has a low condition number. This result further suggests the stability of the LIME algorithm.

The final destination of research in artificial intelligence is to apply this technology in real life and industry to help people of the next generation. However, the trustworthiness and reliability problems in machine learning have become a restriction. I hope the contributions in this thesis can assist future research that can verify the stability of the post-hoc explanation method, which is the key to the trustworthiness and reliability of machine learning.

Bibliography

Shapley, LS (1953). "QUOTA SOLUTIONS OP n-PERSON GAMES1". In: *Edited by Emil Artin and Marston Morse*, p. 343.

Goldstein, Richard (1993). *Conditioning diagnostics: Collinearity and weak data in regression*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Li, Xin and Dan Roth (2002). "Learning question classifiers". In: *COLING 2002: The 19th International Conference on Computational Linguistics*.

Strumbelj, Erik and Igor Kononenko (2010). "An efficient explanation of individual classifications using game theory". In: *The Journal of Machine Learning Research* 11, pp. 1–18.

Maas, Andrew et al. (2011). "Learning word vectors for sentiment analysis". In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150.

Golub, Gene H and Charles F Van Loan (2013). *Matrix computations*. Vol. 3. JHU press.

Socher, Richard et al. (2013). "Recursive deep models for semantic compositionality over a sentiment treebank". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Szegedy, Christian et al. (2013). "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199*.

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy (2014). "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572*.

Kim, Yoon (Oct. 2014). "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: https://aclanthology.org/D14-1181.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). "Character-level convolutional networks for text classification". In: *arXiv preprint arXiv:1509.01626*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Hechtlinger, Yotam (2016). "Interpretation of prediction models using the input gradient". In: *arXiv preprint arXiv:1611.07634*.

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

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016a). "" 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*, pp. 1135–1144.

– (2016b). ""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, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144.

Liang, Bin et al. (2017). "Deep text classification can be fooled". In: *arXiv preprint arXiv:1704.08006*.

Lundberg, Scott M and Su-In Lee (2017a). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

– (2017b). "A unified approach to interpreting model predictions". In: *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). "Learning important features through propagating activation differences". In: *International Conference on Machine Learning*. PMLR, pp. 3145–3153.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR, pp. 3319–3328.

Zhao, Zhengli, Dheeru Dua, and Sameer Singh (2017). "Generating natural adversarial examples". In: *arXiv preprint arXiv:1710.11342*.

Alzantot, Moustafa et al. (2018). "Generating natural language adversarial examples". In: *arXiv preprint arXiv:1804.07998*.

Chen, Jianbo et al. (2018). "L-shapley and c-shapley: Efficient model interpretation for structured data". In: *arXiv preprint arXiv:1808.02610*.

Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Gao, Ji et al. (2018). “Black-box generation of adversarial text sequences to evade deep learning classifiers”. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 50–56.

Ghaeini, Reza, Xiaoli Z Fern, and Prasad Tadepalli (2018). “Interpreting recurrent and attention-based neural models: a case study on natural language inference”. In: *arXiv preprint arXiv:1808.03894*.

Li, Jinfeng et al. (2018). “Textbugger: Generating adversarial text against real-world applications”. In: *arXiv preprint arXiv:1812.05271*.

Du, Mengnan, Ninghao Liu, and Xia Hu (2019). “Techniques for interpretable machine learning”. In: *Communications of the ACM* 63.1, pp. 68–77.

Ghorbani, Amirata, Abubakar Abid, and James Zou (2019). “Interpretation of neural networks is fragile”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 3681–3688.

Heo, Juyeon, Sunghwan Joo, and Taesup Moon (2019). “Fooling neural network interpretations via adversarial model manipulation”. In: *arXiv preprint arXiv:1902.02041*.

Jain, Sarthak and Byron C Wallace (2019). “Attention is not explanation”. In: *arXiv preprint arXiv:1902.10186*.

Jia, Robin et al. (2019). “Certified robustness to adversarial word substitutions”. In: *arXiv preprint arXiv:1909.00986*.

Ren, Shuhuai et al. (2019). “Generating natural language adversarial examples through probability weighted word saliency”. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097.

Subramanya, Akshayvarun, Vipin Pillai, and Hamed Pirsiavash (2019). “Fooling network interpretation in image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2020–2029.

Wallace, Eric et al. (2019). "Universal adversarial triggers for attacking and analyzing NLP". In: *arXiv preprint arXiv:1908.07125*.

Chen, Hanjie and Yangfeng Ji (Nov. 2020). "Learning Variational Word Masks to Improve the Interpretability of Neural Text Classifiers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4236–4251. DOI: 10.18653/v1/2020.emnlp-main.347. URL: https://aclanthology.org/2020.emnlp-main.347.

Jin, Di et al. (2020). "Is bert really robust? a strong baseline for natural language attack on text classification and entailment". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05, pp. 8018–8025.

Lakkaraju, Himabindu, Nino Arsov, and Osbert Bastani (2020). "Robust and stable black box explanations". In: *International Conference on Machine Learning*. PMLR, pp. 5628–5638.

Liu, Xiaodong et al. (2020). "Adversarial training for large neural language models". In: *arXiv preprint arXiv:2004.08994*.

Slack, Dylan et al. (2020). "Fooling lime and shap: Adversarial attacks on post hoc explanation methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.

Wang, Junlin et al. (2020). "Gradient-based Analysis of NLP Models is Manipulable". In: *arXiv preprint arXiv:2010.05419*.

Zhang, Xinyang et al. (2020). "Interpretable deep learning under fire". In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*.

Sinha, Sanchit et al. (2021). "Perturbing inputs for fragile interpretations in deep natural language processing". In: *arXiv preprint arXiv:2108.04990*.

Zafar, Muhammad Bilal et al. (2021). "On the Lack of Robust Interpretability of Neural Text Classifiers". In: *arXiv preprint arXiv:2106.04631*.