

# Perturbing Inputs for Fragile Interpretations in Deep Natural Language Processing

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, Yanjun Qi

Department of Computer Science

University of Virginia

Charlottesville, VA, USA

{ss7mu, hc9mx, as5cu, yangfeng, yq2h}@virginia.edu

## Abstract

Interpretability methods like INTEGRATED GRADIENT and LIME are popular choices for explaining natural language model predictions with relative word importance scores. These interpretations need to be robust for trustworthy NLP applications in high-stake areas like medicine or finance. Our paper demonstrates how interpretations can be manipulated by making simple word perturbations on an input text. Via a small portion of word-level swaps, these adversarial perturbations aim to make the resulting text semantically and spatially similar to its seed input (therefore sharing similar interpretations). Simultaneously, the generated examples achieve the same prediction label as the seed yet are given a substantially different explanation by the interpretation methods. Our experiments generate fragile interpretations to attack two SOTA interpretation methods, across three popular Transformer models and on three different NLP datasets. We observe that the rank order correlation drops by over 20% when less than 10% of words are perturbed on average. Further, rank-order correlation keeps decreasing as more words get perturbed. Furthermore, we demonstrate that candidates generated from our method have good quality metrics. Our code is available at: [github.com/QData/TextAttack-Fragile-Interpretations](https://github.com/QData/TextAttack-Fragile-Interpretations).

## 1 Introduction

Recently, the use of natural language processing (NLP) has gained popularity in many security-relevant tasks like fake news identification (Zhou et al., 2019), authorship identification (Okuno et al., 2014), toxic content detection (Jigsaw, 2017), and for text-based automated privacy policy understanding (Harkous et al., 2018). Since interpretations of NLP predictions have become necessary building blocks of the SOTA deep NLP workflow, explanations have the potential to mislead human users into trusting a problematic interpretation. How-

ever, there has been little analysis of the reliability and robustness of the explanation techniques, especially in high-stake settings, making their utility for critical applications unclear.

Research has shown that it is possible to disrupt and even manipulate interpretations in deep neural networks (Ghorbani et al., 2019; Dombrowski et al., 2019). The core idea in this literature centers around “fragile interpretations”. (Ghorbani et al., 2019) defined that an interpretation is *fragile* if, for a given input, it is possible to generate perturbed input that achieves the same prediction label as the seed, yet is given a substantially different interpretation. Fragility limits how much we can trust and learn from specific interpretations. An adversary for “fragile interpretations” could manipulate the input to draw attention away from relevant words or onto desired features. Such input manipulation might be especially hard to detect because the actual labels have not changed.

The literature includes two relevant groups: (1) to conduct model manipulations (Slack et al., 2019; Wang et al., 2020) (details in Sec. 2), and (2) to manipulate input samples (Ghorbani et al., 2019). There has been little attention studying fragile interpretations via input manipulation in deep NLP.

In this paper, we propose a simple algorithm “ExplainFooler” that can make small adversarial perturbations on text inputs and demonstrate fragility of interpretations. We focus on optimizing two objective metrics - “L2 Norm” or a proposed “Delta LOM”, searching for small word-swap-based input manipulation to produce misleading interpretations and using semantic-oriented constraints to constrain the manipulations. Figure 3 provides one example perturbation process. In summary, this paper provides the following contributions:

- Our input perturbation optimizes to increase the objective metric (“L2 Norm” or “Delta LOM”) that measures difference between the original and generated interpretations. The LOM score

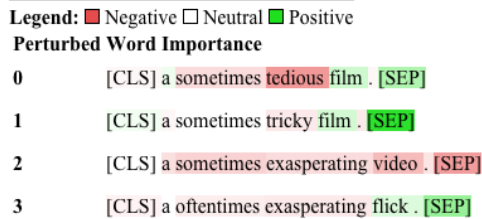


Figure 1: The figure demonstrates the input perturbation process for an increasing number (levels) of word perturbations. The red color depicts negative attribution, and the green shows positive attribution. The saturation of the colors signifies the magnitude of the said attributions. Note: the interpretations gradually become more and more different from the original, although the semantic meaning of the sentence does not change drastically. The model still predicts the correct original output, but the interpretations become senseless as more words get perturbed. [Example taken from the SST-2 dataset. Interpretations calculated using Integrated Gradients on DistilBERT model. Best viewed in color]

captures the approximate center “position” of an interpretation and summarizes it to a scalar.

- We propose an effective algorithm “Explain-Fooler” to optimize the objective metric via an iterative procedure. Our algorithm generates a series of increasingly perturbed text inputs such that their explanations are significantly different from the original but preserving predictions.
- Empirically, we show that it is possible to find perturbed text examples to fool interpretations by INTEGRATED GRADIENT and LIME, even on NLP models that are relatively more robust.

The approximate process and results of word perturbation using our approach is detailed in Figure 1.

## 2 Related Work

**Interpretation Methods:** Several interpretation methods have been proposed (Shrikumar et al., 2017; Li et al., 2015; Bach et al., 2015; Shrikumar et al., 2017) to calculate feature importance scores. Two well-known methods in this area are Integrated Gradients (IG) (Sundararajan et al., 2017) and Local Interpretable Model Explanations (LIME) (Ribeiro et al., 2016b). IG computes the scores by summing up the gradients along a path from the baseline to the input in a fixed number of steps and subsequently multiplied by the input itself. IG overcomes the saturation problem discussed in (Shrikumar et al., 2017; Sundararajan et al., 2017). On the other hand, LIME is a completely black-box approach which explains the predictions of any classifier in an interpretable and

faithful manner, by learning an interpretable model locally around the prediction by training the model on perturbations generated around the input.

**Fragile Interpretations** More recently, several works have focused on discussing the robustness of the said interpretations. Studies have demonstrated that the interpretations generated are not robust and can be easily manipulated due to high dimensionality of networks. (Ghorbani et al., 2019; Dombrowski et al., 2019; Slack et al., 2019; Wang et al., 2020). Multiple other works have tried to fix the problem by making interpretations robust (Lakkaraju et al., 2020; Rieger and Hansen, 2020).

(Wang et al., 2020) demonstrated that it is possible to introduce a new model over the original and alter gradients, to fool gradient-based interpretation methods. Similarly, (Slack et al., 2019) showed that black-box interpretation methods can also be fooled by allowing an adversarial classifier component. More recently, (Zafar et al., 2021) demonstrated empirically. that interpretability methods produce varying results on the same models but differently initialized.

**Adversarial Examples that fool NLP Predictions:** Adversarial examples are inputs to a predictive machine learning model that are maliciously designed to fool the model predictions (Goodfellow et al., 2014). Multiple recent works have focused on applying the concept of adversarial examples on language inputs, including (1) attacks by Character Substitution (Ebrahimi et al., 2017; Gao et al., 2018; Li et al., 2018); (2) attacks by Paraphrase (Ribeiro et al., 2018; Iyyer et al., 2018); (c) attacks by Synonym Substitution (Alzantot et al., 2018; Jin et al., 2020; Kuleshov et al., 2018; Papernot et al., 2016); (d) attacks by Word Insertion or Removal (Liang et al., 2017; Samanta and Mehta, 2017); (e) attacks by limiting  $L_p$  distance in a latent embedding space (Zhao et al., 2017). Our proposed algorithm is closely connected to the TextFooler algorithm (Jin et al., 2020) that searches for input perturbations to achieve mis-classification. Differently, we optimize the “L2 Norm” and “Location of Mass (LOM)” objective directly on the input space for fragile explanations.

## 3 Proposed Method

In this section, we present our algorithm to generate perturbed sentences that demonstrate fragile interpretations. First, we propose the metric “Location of Mass (LOM)” and L2 Norm, followed

by a discussion on the search strategy to optimize the objective metrics. Subsequently, we discuss the interpretation method choices and end with the final candidates’ selection procedure and pseudo-code for our algorithm (Algorithm 1). We denote a text input as  $\mathbf{x}$  and its word importance score vector (from a specific interpretation strategy on a particular NLP model) using notation  $I$ .

### 3.1 Difference Metrics on Interpretation

To quantify the difference between two interpretations, we propose two objective metrics - “Delta LOM” and “L2 Norm”. These metrics are divergent - that is higher the metric, the more different the interpretations.

#### 3.1.1 “Location of Mass (LOM)” Score

First, we propose a metric inspired by (Ghorbani et al., 2019) which provides a quantifiable “position” of the interpretations of a sentence. First, we define the “Location of Mass (LOM)” score as:

$$LOM(I) = \frac{\sum_{t=0}^{t=n-1} (i_t * t)}{\sum_{t=0}^{t=n-1} i_t} \quad (1)$$

Here  $n$  is the length of the sentence (along with starting/end special tokens). And  $i_t$  is the interpretability score assigned to the token at index ‘t’. We then propose to calculate the “Delta LOM” metric as: the difference between the LOM scores on the two interpretations  $I_1$  and  $I_2$ :

$$\Delta LOM(I_1, I_2) = |LOM(I_1) - LOM(I_2)| \quad (2)$$

The intuition behind this metric comes from the fact that changing the approximate position of the “center” of interpretations changes the relative position and magnitudes of interpretations. This observation is demonstrated in Figure 3.

#### 3.1.2 L2 Norm Metric

We also propose to use a standard L2 Norm to measure difference between two interpretations. Mathematically it is computed as follows:

$$L2Norm(I_1, I_2) = \|I_1 - I_2\|_2 \quad (3)$$

L2 Norm quantifies the extent of difference, higher the L2 Norm- higher the difference in pattern of two interpretations.

### 3.2 Searching for Word-level Perturbations

Our objective is to perturb a seed input  $\mathbf{x}$ , into a slightly-modified text  $\mathbf{x}_{adv}$ , so that  $\Delta LOM$  or  $L2Norm$  is maximized under a set of constraints.

First, we rank each word of an input sentence in the order of their importance to a model’s predictions. This is done by the Leave-one-out approach (Li et al., 2016), which removes each word from the sentence one at a time and measures the change in prediction values, ranking the words which produce the greatest change as most important. Subsequently, we start our search in decreasing order of word importance and substituting each word with their  $k$  closest nearest-neighbors according to their counter-fitting synonym embeddings (Mrksic et al., 2016). For every subsequent word replacement, interpretation is calculated according to victim interpretation strategy we try to attack.

### 3.3 Ensuring Constraints

We enforce the following four constraints for each perturbed candidate to ensure candidates do not lose their linguistic structure and approximate semantic meaning of the seed input.

- Repeat Modification: Stops the same word from getting perturbed more than once.
- Stop Word Modification: This excludes pre-defined stop words from getting perturbed.
- Word Embedding Distance: Swaps the original word with words that have less than a particular embedding distance using Counter-Fitting Embeddings.
- Part of Speech: Replaces the original word with only words from the same part of speech.
- Sentence Embedding: Ensure the difference in the Universal Sentence Embedding is less than a pre-defined threshold (Cer et al., 2018).

### 3.4 Victim Interpretation Choices

**Integrated Gradient:** We calculate INTEGRATED GRADIENT (Sundararajan et al., 2017) interpretations of NLP models using the open-source package Captum (Kokhlikyan et al., 2020) that provides accurate implementations of various interpretation methods. We use the popular INTEGRATED GRADIENT algorithm to calculate the importance scores on the embedding space of the models. Once the interpretations are calculated, they are summed up along the dimension axis to derive the word importance scores. Subsequently, the  $\Delta LOM$  and L2 Norm scores of each candidate perturbation are calculated against the original input’s interpretation.

**LIME:** The LIME interpretations are calculated using the official LIME code provided by (Ribeiro et al., 2016a). We normalize the LIME scores by

dividing the vector with its  $L_2$ -norm. Subsequently, the  $\Delta$ LOM and L2 Norm scores of each candidate perturbation are calculated against the original input’s interpretation.

### 3.5 Finding the ideal candidate

Once we obtain all the candidates and their metric scores on every candidate achieving the same prediction label as the original, we store those ideal candidates with each ‘ $m$ ’ number of words perturbed. This gives us a list of candidates for each level of word perturbation and the associated change in objective metric scores. Next, for each level, the candidate with the highest metric score against the original is chosen. Finally, we convert the number of perturbed words into a ratio with respect to the input’s length. This is done to take into account the varying sentence lengths and get a normalized measure. The ratio is limited to 50% because once more than half the words are perturbed, the sentence starts losing its semantic meaning. The complete selection process is schematically detailed in Figure 2

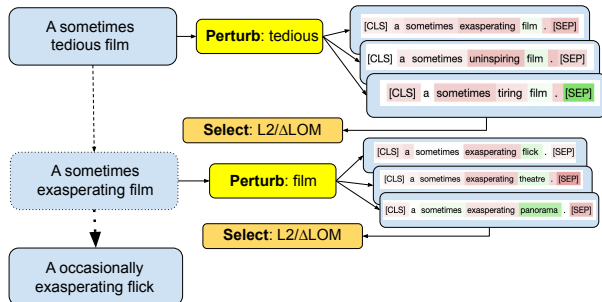


Figure 2: A schematic diagram of the proposed “ExplainFooler” algorithm. In the figure, the “Perturb” step generates a list of all possible perturbations according to the constraints as discussed in Sec. 3.3. The interpretation are generated as discussed in Sec. 3.4. The selection process uses objective metrics explained by Sec. 3.1.

### 3.6 Algorithm

Algorithm 1 “ExplainFooler” provides pseudocode to compute and select a list of candidates that can induce fragile explanations. Our implementation adapts and builds on top of the open-source package TextAttack (Morris et al., 2020).

## 4 Experiments

### 4.1 Data Summary

The experiments are conducted on three different datasets for text classification task. Experiments are conducted on the validation set for SST-

```

Result: A - list of candidate sentences
            ordered by number of words
            perturbed from original

For each sentence in dataset
A ← empty
S ← original sentence
I0 ← InterpretMethod(S)
P ← ordered list of important words (LOO)
while ≤50% of words perturbed from P
do
    w ← P[0]
    C ← empty
    while Possible perturbations exist do
        c ← Perturb S and get candidate
        if constraints pass and prediction
            label is same as S then
            I ← InterpretMethod(c)
            Δdiff ← diff(I0, I),
            C ← C ∪ (Δdiff, c)
        else
            continue
    A ← A ∪ c where max(diff)
    P ← remove P[0]

```

Algorithm 1: The “ExplainFooler” algorithm

2(Socher et al., 2013), test set for AG News(Zhang et al., 2015) and test set for IMDB dataset(Maas et al., 2011). We select the first 500 sentences from the SST-2 and AGNews datasets and 100 sentences from the test set from IMDB dataset to run our experiments. We discard sentences with just 2 words or less.

- **SST-2:** The Stanford Sentiment Treebank-2 dataset for movie review classification. It has two classes: positive and negative. Experiments are conducted on the first 500 sentences of the validation set.
- **AG News:** A collection of raw news articles belonging to 4 different classes including World, Sports, Science/Technology and Business. Experiments are conducted on the first 500 sentences of test set.
- **IMDB:** IMDB website dataset for binary sentiment classification containing a set of highly polar movie reviews. Experiments are conducted on first 500 sentences of test set except for LIME where only 100 sentences are used due to very high computation time due to very long average sentence length.



True Label	Predicted Label	Confidence	Predicted Class	Words Perturbed	LOM Score	Word Importance
0	0	0.98	Negative	0	0.00	[CLS] a sometimes tedious film . [SEP]
0	0	0.97	Negative	1	0.82	[CLS] a sometimes exasperating film . [SEP]
0	0	0.86	Negative	2	0.92	[CLS] a sometimes exasperating flick . [SEP]
0	0	0.57	Negative	3	0.95	[CLS] a occasionally exasperating flick . [SEP]

Figure 3: The figure demonstrates the  $\Delta$ LOM score for an increasing number of word perturbations. The interpretations gradually become more and more different from the original although the semantic meaning of the sentence does not change drastically. We can see that the model still predicts the original output but the interpretations become senseless as the  $\Delta$ LOM score increases.[Best viewed in color]

## 4.2 Interpretability Parameters

**IG:** As integrated gradients is a gradient based approach and requires a reference baseline, we compute the attributions on the embedding space and set the reference baseline to the special token <PAD> which is reserved in transformers as a special character. The step size for Integrated Gradients were chosen as 50 i.e. from reference to baseline, the gradients were summed up in 50 continuous steps.

**LIME:** The number of perturbations for LIME were chosen as 500 and the maximum number of top-k words were chosen as 512 words - the truncation limit for all the models.

## 4.3 Perturbation Parameters

We choose the number of nearest neighbours as 50 for swapping the words to limit the number of candidates. The maximum embedding cosine similarity between sentences was set as 0.5 to ensure sentences do not lose their semantic meaning.

## 4.4 Under the Hood

**Pre-processing:** All sentences with less than 2 words in all datasets are removed due to word perturbations not existing in some cases. In other cases, the smaller sentence have a very big difference in rank correlation which can spuriously decrease evaluation metrics. Each sentence from all datasets is also converted to lower-case.

**Fixing Tokenizations:** As pre-trained tokenizers for transformer models contain a ML matching based lookup vocabulary, many words in candidate sentences are tokenized in an unexpected manner. This results in the change of length of the token list which in turn changes the length of interpretations. To alleviate this problem, we test 2 distinct approaches to combine the unnaturally tokenized words into their original form.

- **Average:** The first approach combines all the tokens prefixed by a set character (## in case of DistilBERT) into one single word and assigns the average value of the tokens to the combined tokens
- **Max:** The second approach combines all the tokens prefixed by a set character (## in case of DistilBERT) into one single word and assigns the absolute maximum value with sign to the combined word.

Upon careful review, we utilize the second approach for our experiments. This is because, in uncommon cases where tokens hold opposite polarity to the ones in the word result in ‘diluted’ value of the original token. An example of the effectiveness of the ‘Max’ approach is given in Figure 4.

[CLS] unflinchingly bleak and desperate [SEP]
[CLS] un ##fl ##in ##ching ##ly bleak and desperate [SEP]
[CLS] unflinchingly bleak and desperate [SEP]
[CLS] un ##fl ##in ##ching ##ly bleak and desperate [SEP]

Figure 4: The figure demonstrates the combining of tokens of a sentence tokenized using DistilBERT’s pre-trained tokenizer. The top group of sentences demonstrates averaging approach and the bottom group of sentences are combined using Abs-Max approach detailed in 4.4. [Best viewed in color]

## 4.5 Evaluation Metrics

### 4.5.1 Rank Correlation

To compare the correlation between interpretations of 2 sentences, we use the Spearman rank correlation metric. The more the ranks of the interpretations agree with each other, the higher the rank correlations. Importantly, we clip the negative values of the metric to 0. This is done because

a negative correlation does not make sense when only comparing the difference in ranks and can spuriously bring down the average scores.

$$R - \text{Correlation} = \max(0, \text{Spearman}(I_1, I_2)) \quad (4)$$

We report results in Tables 1,3,8,10 and corresponding violin graphs Figures 7,10 of average Spearman rank order correlations and standard deviations versus ratio of words perturbed for 3 datasets (SST-2, AG News and IMDB) across both models (DistilBERT-uncased and RoBERTa-base) using 2 interpretability methods - INTEGRATED GRADIENT and LIME.

#### 4.5.2 Top-50% Intersection

To compare the extent to which the words with highest attributions are correctly predicted by both the interpretation methods, we use the Top-k% intersection metric. To compute the intersection, we first find the words with the maximum absolute value of attributions (most important for prediction). We calculate the intersection of the top 50% highest attribution words.

$$\text{Intersection} = \frac{\bigcap(\text{argsort}(I_1), \text{argsort}(I_2))}{0.5 * \text{length}(I_1)} \quad (5)$$

where argsort returns the indices of the top-50% of the words in a sentence with highest attributions.

#### 4.5.3 Candidate Quality

To judge the quality of the candidates generated using “ExplainFooler”, we calculate two different commonly used quality metrics from adversarial attack literature - Perplexity and absolute number of grammar errors similar to (Li et al., 2020).

**Perplexity** We first use perplexity to estimate the fluency of candidates generated using “ExplainFooler”. The lower the value, the more fluent the candidates, measured using a small size GPT-2 model (50k vocabulary) (Radford et al., 2019).

**Grammatical Errors** Estimates the average number of absolute difference in grammatical errors between the original and the candidate sentences. We use the Language Tool (Naber et al., 2003) to compute the errors.

### 4.6 Model Choices

The robustness concern of interpretation strategies challenges their use in critical applications, raising concerns like lack of trust. However, it is unclear what causes the “fragile explanations”, the model

or the interpretation? We therefore select three different transformer models namely, DistilBERT-uncased (Sanh et al., 2019), RoBERTa-base (Liu et al., 2019) and BERT-base (Devlin et al., 2018) to conduct our experiments. More importantly, we re-train the BERT-base to obtain the BERT-base-adv model that is an adversarially trained version of the BERT-base model. The rationale behind the choices is to investigate the impact of model’s robustness on the robustness of the interpretations. (1) First, a generic transformer model like DistilBERT is relatively smaller and faster but less robust than the other two. (2) Next RoBERTa is extensively better pre-trained and has a far more robust performance. (3) Lastly, BERT-base-adv model is trained from adversarial training. We use the popular TextFooler(Jin et al., 2020) algorithm to generate adversarial examples via the open-source package Textattack. DistilBERT and RoBERTa models were from pre-trained models, fine-tuned on the respective datasets and we take them from the Huggingface’s transformer model hub(Wolf et al., 2020) without change. Differently, BERT-base-adv model is adversarially trained by attacking 10000 training examples for the IMDB and AG datasets and attacking all training samples for SST-2 dataset.

## 5 Empirical Results

### 5.1 Rank Order and Top-50% Intersection

The results are reported in a tabular manner across 3 datasets (SST-2, AG News and IMDB), 3 models (DistilBERT, RoBERTa and BERT-adv (Section A(Appendix)) and 2 interpretability methods covering both metrics - L2 Norm, “Delta LOM” and compared against random candidate selection independent of both metrics. The first set of tables (Tables 1 and 3) report the average rank-order correlation between interpretations from the perturbed and the original, across different perturbation ratios in buckets of 10%. The second set of tables (Tables 2 and 4) report the average top-50% intersection. The rank correlation results for the IMDB datasets are reported only on IG due to excessive computational constraints. Due to space constraints, the results for both AGNews and IMDB datasets are reported in (Tables 8-11 and Tables 12-13 respectively, Section A.2 (Appendix)) along with a more detailed representation of the intra-bucket distribution in the form of Violin Graphs (Section A.3 (Appendix)).

SST-2									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.65	0.78	0.8	0.64	0.76	0.81	0.53	0.6	0.73
0.1-0.2	0.53	0.65	0.64	0.57	0.61	0.69	0.43	0.43	0.52
0.2-0.3	0.42	0.55	0.59	0.51	0.59	0.6	0.3	0.33	0.42
0.3-0.4	0.36	0.48	0.48	0.47	0.47	0.55	0.35	0.3	0.43
0.4-0.5	0.31	0.42	0.47	0.42	0.43	0.48	0.14	0.24	0.36

Table 1: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- SST-2 over 3 models - DistilBERT, RoBERTa and BERT-adv.

SST-2									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.77	0.78	0.81	0.75	0.76	0.81	0.75	0.76	0.79
0.1-0.2	0.71	0.71	0.73	0.71	0.71	0.74	0.68	0.68	0.7
0.2-0.3	0.67	0.68	0.68	0.68	0.69	0.7	0.63	0.64	0.65
0.3-0.4	0.65	0.65	0.65	0.66	0.67	0.67	0.61	0.61	0.64
0.4-0.5	0.6	0.62	0.62	0.63	0.63	0.65	0.59	0.56	0.63

Table 2: Change in average Top-50% intersection using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- SST-2 over 3 models - DistilBERT, RoBERTa and BERT-adv.

SST-2									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.64	0.7	0.79	0.59	0.66	0.76	0.57	0.68	0.72
0.1-0.2	0.52	0.58	0.65	0.58	0.63	0.7	0.37	0.52	0.59
0.2-0.3	0.46	0.51	0.56	0.52	0.58	0.62	0.34	0.47	0.54
0.3-0.4	0.39	0.43	0.46	0.48	0.54	0.58	0.31	0.36	0.36
0.4-0.5	0.23	0.29	0.46	0.55	0.55	0.54	0.28	0.2	0.24

Table 3: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: LIME, for dataset- SST-2 over 3 models - DistilBERT, RoBERTa and BERT-adv.

SST-2									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.64	0.7	0.79	0.59	0.66	0.76	0.57	0.68	0.72
0.1-0.2	0.52	0.58	0.65	0.58	0.63	0.7	0.37	0.52	0.59
0.2-0.3	0.46	0.51	0.56	0.52	0.58	0.62	0.34	0.47	0.54
0.3-0.4	0.39	0.43	0.46	0.48	0.54	0.58	0.31	0.36	0.36
0.4-0.5	0.23	0.29	0.46	0.55	0.55	0.54	0.28	0.2	0.24

Table 4: Change in average Top-50% intersection using metrics - L2 Norm, LOM and random selection computed using the interpretability method: LIME, for dataset- SST-2 over 3 models - DistilBERT, RoBERTa and BERT-adv.

Perplexity (lower is better)									
Dataset	DistilBERT			RoBERTa			BERT-adv		
	C-avg	$\Delta$ LOM	L2	C-avg	$\Delta$ LOM	L2	C-avg	$\Delta$ LOM	L2
SST-2 (130.85)	352.55	<b>285.88</b>	286.62	272.67	<b>245.55</b>	248.86	388.99	<b>237.76</b>	238.14
AGNews (76.18)	359.13	<b>239.95</b>	241.33	275.31	<b>194.89</b>	195.29	352.05	230.44	<b>229.38</b>
IMDB (39.12)	101.71	<b>65.04</b>	65.1	101.3	<b>62.41</b>	63.65	84.51	<b>63.21</b>	64.58

Table 5: Average values of perplexity calculated using a small GPT-2 model over all candidates generated by “ExplainFooler” (C-avg). The values in columns LOM and L2 denote the perplexity values calculated on the selected sentences using the proposed metrics. The average value of perplexity of original sentences in dataset are given in parentheses. Selection using metrics give more fluent sentences.

Dataset	Model	Number of Words Perturbed				
		0	1	2	3	4
SST-2	DistilBERT	0.97	0.95	0.92	0.88	0.82
	RoBERTa	0.98	0.98	0.98	0.98	0.98
	BERT-adv	0.97	0.96	0.94	0.92	0.91
AGNews	DistilBERT	0.98	0.97	0.93	0.86	0.82
	RoBERTa	0.98	0.98	0.98	0.98	0.98
	BERT-adv	0.97	0.95	0.95	0.94	0.91
IMDB	DistilBERT	0.99	0.97	0.94	0.89	0.84
	RoBERTa	0.98	0.98	0.98	0.98	0.98
	BERT-adv	0.97	0.96	0.94	0.94	0.91

Table 6: Average model confidence for correct prediction values for increasing number of words perturbed over models - DistilBERT, RoBERTa and BERT-adv on datasets - SST-2, AGNews and IMDB

A bucket represents all instances of perturbed candidates in the ratio between that lower and higher range. For example, bucket between “0.1-0.2” contains all rank-order correlations from sentences with a percentage of words perturbed between 10% and 20%. We also provide violin plots in appendix showcasing intra-bucket distribution for the dataset SST-2 (Figures 7-10). We observe that both average rank-order correlation and top-50% intersection scores decrease as the ratio of words being perturbed increases. Observations imply that interpretations of sentences become increasingly dissimilar to the original sentence as more words are perturbed even though the prediction robustness of the models remains high (see Table 6, Figure 12). Similar trends are observed across all models, datasets, and covering both victim interpretability methods. These empirical observations demonstrate interpretations generated by INTEGRATED GRADIENT and LIME are fragile for all models - even models that are adversarially more robust (BERT-adv). To further demonstrate effectiveness of proposed metrics, we plot violin plots on SST-2 dataset for avg. rank correlation versus selection using metrics and random. (Figure 5 - Appendix)

## 5.2 Quality of candidates

**Perplexity** The average perplexity values over all models and datasets are reported in Table 5. For

Grammatical Errors (lower is better)			
Model	C-avg	L2	$\Delta$ LOM
DistilBERT	0.59	0.59	<b>0.58</b>
RoBERTa	0.79	0.76	<b>0.75</b>
BERT-adv	0.60	<b>0.51</b>	0.52

Table 7: Average number of grammatical errors on candidates generated using “ExplainFooler” on the SST-2 dataset (C-avg). The accompanying values in columns  $\Delta$ LOM and L2 denote the grammar errors calculated on the sentences selected using the proposed metrics.

each dataset, model pair values corresponding to proposed metrics and random selection are reported. It can be observed that perplexity of candidates selected using proposed metrics have lower perplexity score (implying better fluency) than average of all candidates generated by “ExplainFooler”.

**Grammatical Errors** Estimates average number of absolute difference in grammatical errors between the original and candidate sentences. We use Language Tool (Naber et al., 2003) to compute the errors. The results for SST-2 dataset are reported in Table 7.

## 6 Conclusions

Literature sees a growing emphasis on interpretation techniques for explaining NLP model predictions. Our work demonstrates a novel algorithm that generates perturbed inputs that provide evidence of fragile interpretations. We demonstrate the effectiveness of our approach across three different models, with one of them adversarially trained. Our results show that it is possible to attack interpretations using simple input-level word swaps under certain constraints. We also demonstrate that both black and white-box interpretability approaches (LIME and INTEGRATED GRADIENT) show fragility in their derived interpretations. We hope our findings can pave lights for future studies on defending against problem of fragile interpretations in NLP.



## References

- 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*.
- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*, 10(7):e0130140.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. *Universal sentence encoder*. *CoRR*, abs/1803.11175.
- 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*.
- Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. *Explanations can be manipulated and geometry is to blame*. In *Advances in Neural Information Processing Systems*, volume 32, pages 13589–13600. Curran Associates, Inc.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.
- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.
- Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. *Interpretation of neural networks is fragile*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *Proceedings of the 27th USENIX Conference on Security Symposium, SEC'18*, page 531–548, USA. USENIX Association.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. *Adversarial example generation with syntactically controlled paraphrase networks*. *CoRR*, abs/1804.06059.
- Google Jigsaw. 2017. Perspective api. <https://www.perspectiveapi.com/>.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. *Is bert really robust? a strong baseline for natural language attack on text classification and entailment*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.
- Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. *Captum: A unified and generic model interpretability library for pytorch*.
- Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.
- Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. *Robust and stable black box explanations*. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5628–5638. PMLR.
- Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.
- Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.
- Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar S. Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- 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-Volume 1*, pages 142–150. Association for Computational Linguistics.

- John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. [Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp](#).
- Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. In *HLT-NAACL*.
- Daniel Naber et al. 2003. A rule-based style and grammar checker.
- Syunya Okuno, Hiroki Asai, and Hayato Yamana. 2014. A challenge of authorship identification for ten-thousand-scale microblog users. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 52–54. IEEE.
- Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pages 1135–1144. ACM.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.
- Laura Rieger and Lars Kai Hansen. 2020. A simple defense against adversarial attacks on heatmap explanations. *arXiv preprint arXiv:2007.06381*.
- Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.
- Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2019. How can we fool lime and shap? adversarial attacks on post hoc explanation methods.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.
- Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. [Gradient-based analysis of NLP models is manipulable](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–258, Online. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Muhammad Bilal Zafar, Michele Donini, Dylan Slack, Cédric Archambeau, Sanjiv Das, and Krishnamurthy Kenthapadi. 2021. On the lack of robust interpretability of neural text classifiers. *arXiv preprint arXiv:2106.04631*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.
- Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.
- Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 836–837.

## A Appendix

### A.1 Compare with Baseline

Figures 5,6 show the decrease in average rank correlation when considering random candidates as opposed to selection using the LOM metric.

### A.2 Additional Results

In this section we report the average rank order correlation and the average top-50% intersection scores for AGNews and IMDB datasets. The Tables 8,9 correspond to AGNews’ rank correlation and top-50% scores using INTEGRATED GRADIENT whereas Tables 10,11 show same values using LIME. Tables 12 and 13 show similar values but for IMDB dataset.

### A.3 Violin Plots for intra-bucket distribution analysis

The Violin plots convey more information about the relative distribution of average rank correlations and Top-50% values for various bucket ratios. The following figures are only reported on the SST-2 dataset for each combination of evaluation metric and interpretability methods.

### A.4 Visual Results

A few visual results demonstrating the gradual change in interpretations of candidate adversaries is shown in Figure 12. It can be observed that  $\Delta$ LOM score gradually increases with word perturbations. The examples demonstrate the same 3 sentences from the dataset perturbed under DistilBERT and RoBERTa respectively.

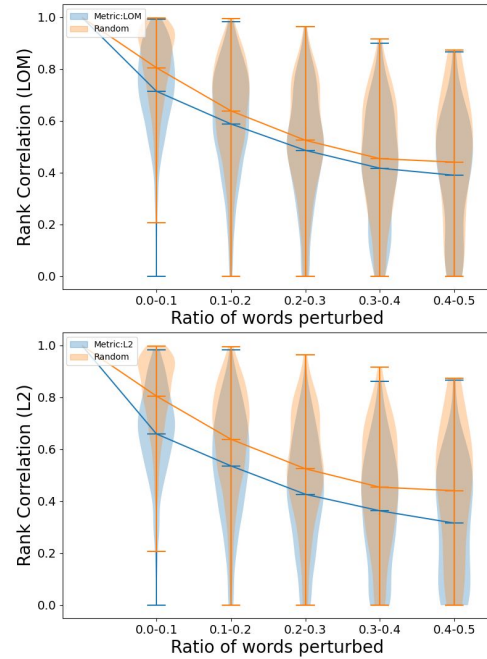


Figure 5: The violin graphs demonstrate the effectiveness of candidate selection based on the proposed metrics LOM and L2 Norm over random selection for SST-2 dataset. As it can be seen that the selection based on the proposed metrics disrupts rank correlation more as compared to randomly selecting candidates.

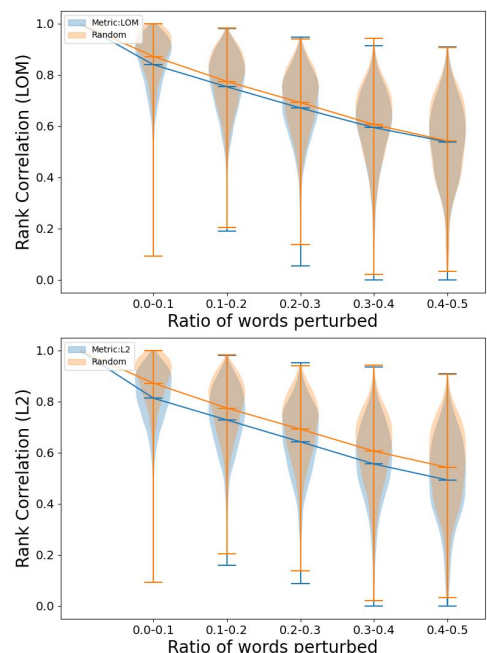


Figure 6: The violin graphs demonstrate the effectiveness of candidate selection based on the proposed metrics LOM and L2 Norm over random selection for AG-News dataset. As it can be seen that the selection based on the proposed metrics disrupts rank correlation more as compared to randomly selecting candidates.

AGNews									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.81	0.84	0.86	0.73	0.68	0.82	0.38	0.56	0.63
0.1-0.2	0.72	0.75	0.78	0.65	0.57	0.72	0.32	0.42	0.46
0.2-0.3	0.64	0.66	0.69	0.62	0.52	0.66	0.28	0.32	0.29
0.3-0.4	0.55	0.58	0.58	0.58	0.48	0.62	0.25	0.25	0.26
0.4-0.5	0.49	0.52	0.56	0.52	0.42	0.56	0.18	0.23	0.24

Table 8: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- AGNews over 3 models - DistilBERT, RoBERTa and BERT-adv.

AGNews									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.64	0.65	0.71	0.7	0.74	0.85	0.48	0.51	0.79
0.1-0.2	0.57	0.58	0.69	0.61	0.64	0.8	0.37	0.4	0.69
0.2-0.3	0.57	0.58	0.62	0.55	0.59	0.77	0.24	0.27	0.64
0.3-0.4	0.53	0.53	0.58	0.52	0.55	0.74	0.22	0.24	0.6
0.4-0.5	0.51	0.52	0.56	0.45	0.5	0.71	0.19	0.24	0.58

Table 9: Change in average Top-50% intersection using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- AGNews over 3 models - DistilBERT, RoBERTa and BERT-adv.

AGNews									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.65	0.69	0.71	0.58	0.57	0.61	0.7	0.61	0.72
0.1-0.2	0.59	0.6	0.62	0.55	0.54	0.56	0.69	0.45	0.7
0.2-0.3	0.53	0.53	0.58	0.54	0.53	0.48	0.65	0.35	0.66
0.3-0.4	0.48	0.52	0.55	0.51	0.51	0.36	0.65	0.28	0.65
0.4-0.5	0.44	0.38	0.46	0.43	0.42	0.43	0.59	0.26	0.61

Table 10: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: LIME, for dataset- AGNews over 3 models - DistilBERT, RoBERTa and BERT-adv.

AGNews									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.62	0.64	0.66	0.6	0.62	0.61	0.56	0.56	0.55
0.1-0.2	0.58	0.59	0.63	0.58	0.58	0.58	0.53	0.54	0.53
0.2-0.3	0.57	0.57	0.58	0.55	0.57	0.57	0.51	0.51	0.52
0.3-0.4	0.55	0.56	0.58	0.55	0.55	0.57	0.5	0.5	0.52
0.4-0.5	0.53	0.55	0.57	0.54	0.54	0.56	0.51	0.5	0.52

Table 11: Change in average Top-50% intersection using metrics - L2 Norm, LOM and random selection computed using the interpretability method: LIME, for dataset- AGNews over 3 models - DistilBERT, RoBERTa and BERT-adv.



IMDB									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.69	0.69	0.71	0.75	0.74	0.8	0.45	0.44	0.55
0.1-0.2	0.53	0.55	0.61	0.64	0.59	0.69	0.32	0.32	0.41
0.2-0.3	0.41	0.44	0.5	0.51	0.48	0.58	0.28	0.29	0.39
0.3-0.4	0.42	0.39	0.49	0.45	0.41	0.51	0.27	0.28	0.34
0.4-0.5	0.33	0.31	0.41	0.34	0.37	0.49	0.12	0.17	0.28

Table 12: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- IMDB over 3 models - DistilBERT, RoBERTa and BERT-adv.

IMDB									
Ratio	DistilBERT			RoBERTa			BERT-adv		
	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random	L2	$\Delta$ LOM	Random
0-0.1	0.7	0.71	0.74	0.73	0.75	0.76	0.61	0.63	0.66
0.1-0.2	0.6	0.63	0.66	0.64	0.66	0.69	0.58	0.61	0.63
0.2-0.3	0.59	0.6	0.63	0.57	0.63	0.65	0.57	0.6	0.61
0.3-0.4	0.56	0.57	0.57	0.57	0.58	0.6	0.55	0.57	0.58
0.4-0.5	0.52	0.52	0.52	0.52	0.55	0.57	0.54	0.54	0.54

Table 13: Change in average rank-order correlation using metrics - L2 Norm, LOM and random selection computed using the interpretability method: INTEGRATED GRADIENT, for dataset- IMDB over 3 models - DistilBERT, RoBERTa and BERT-adv.

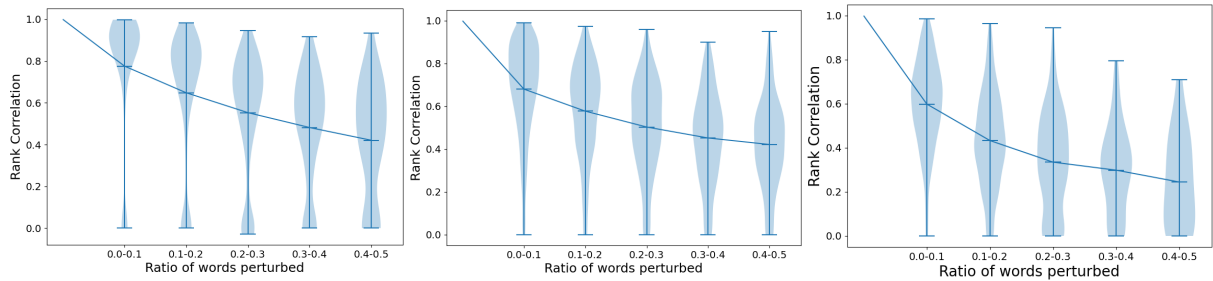


Figure 7: Average Rank-correlation for the dataset: SST-2, using metric: LOM on models DistilBERT, RoBERTa and BERT-adv using interpretability method -INTEGRATED GRADIENT

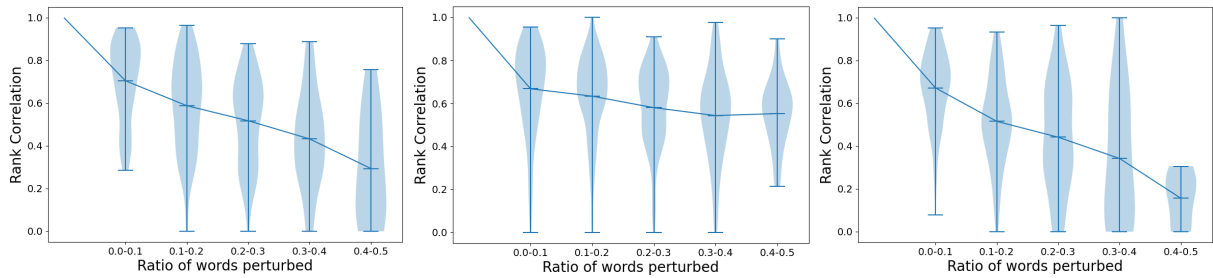


Figure 8: Average Rank-correlation for the dataset: SST-2, using metric: LOM on models DistilBERT, RoBERTa and BERT-adv using interpretability method -LIME

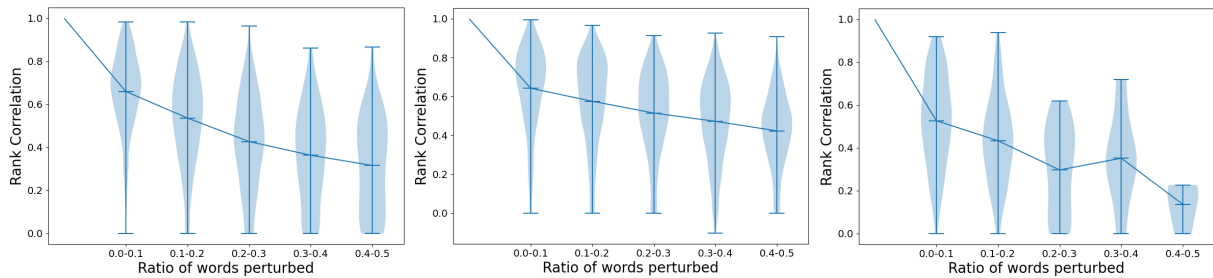


Figure 9: Average Rank-correlation for the dataset: SST-2, using metric: L2 Norm on models DistilBERT, RoBERTa and BERT-adv using interpretability method -INTEGRATED GRADIENT

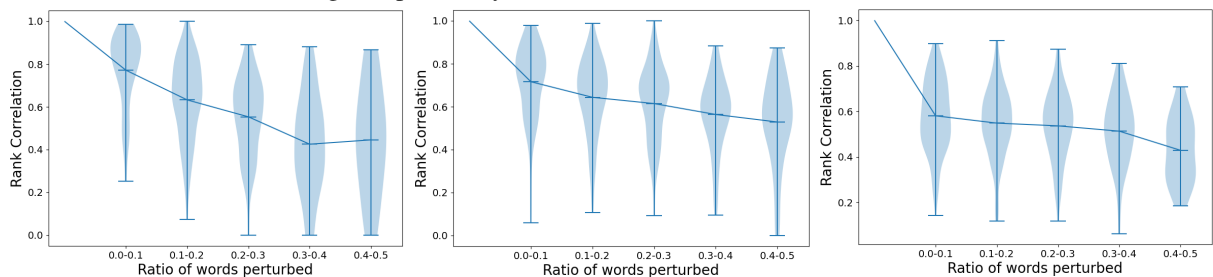


Figure 10: Average Rank-correlation for the dataset: SST-2, using metric: L2 Norm on models DistilBERT, RoBERTa and BERT-adv using interpretability method -LIME

True Label	Predicted Label	Confidence	Predicted Class	Words Perturbed	LOM Score	Word Importance
1	1	1.00	Positive	0	0.00	[CLS] it ' s a charming and often affecting journey . [SEP]
1	1	1.00	Positive	1	0.31	[CLS] it ' s a alluring and often affecting journey . [SEP]
1	1	0.98	Positive	2	0.80	[CLS] it ' s a beautiful and often afflicted journey . [SEP]
1	1	0.72	Positive	3	1.22	[CLS] it ' s a beautiful and often afflicted rook . [SEP]
0	0	0.98	Negative	0	0.00	[CLS] unflinchingly bleak and desperate [SEP]
0	0	0.89	Negative	1	0.72	[CLS] unflinchingly baleful and desperate [SEP]
0	0	0.66	Negative	2	0.86	[CLS] unflinchingly dusky and depressive [SEP]
0	0	0.99	Negative	0	0.00	[CLS] it ' s slow - - very , very slow . [SEP]
0	0	0.99	Negative	1	0.25	[CLS] it ' s slow - - crucially , very slow . [SEP]
0	0	0.99	Negative	2	0.91	[CLS] it ' s slow - - crucially , vitally slow . [SEP]
0	0	0.97	Negative	3	1.44	[CLS] it ' s slow - - crucially , highly lent . [SEP]
0	0	0.91	Negative	4	1.61	[CLS] it ' s sluggish - - crucially , highly lent . [SEP]

Figure 11: A few random sentence explanations from the SST-2 dataset calculated on DistilBERT-uncased using INTEGRATED GRADIENT. [Best viewed in color].

True Label	Predicted Label	Confidence	Predicted Class	Words Perturbed	LOM Score	Word Importance
1	1	1.00	Positive	0	0.00	it ' s a charming and often affecting journey . #/s
1	1	1.00	Positive	1	0.73	it ' s a mignon and often affecting journey . #/s
1	1	0.99	Positive	2	1.19	it ' s a dreamy and often plaguing journey . #/s
1	1	0.98	Positive	3	1.12	it ' s a dreamy and often effect nomad . #/s
0	0	1.00	Negative	0	0.00	unflinchingly bleak and desperate #/s
0	0	1.00	Negative	1	0.45	unflinchingly dreary and desperate #/s
0	0	0.90	Negative	2	0.76	unflinchingly sombre and frenetic #/s
0	0	1.00	Negative	0	0.00	it ' s slow -- very , very slow . #/s
0	0	1.00	Negative	1	0.78	it ' s slow -- very , very lent . #/s
0	0	0.97	Negative	2	0.73	it ' s slow -- perfectly , very slowness . #/s
0	0	1.00	Negative	3	0.31	it ' s slow -- immeasurably , immeasurably slowness . #/s
0	0	1.00	Negative	4	0.39	it ' s slowest -- immeasurably , immeasurably slowness . #/s

Figure 12: The same sentence visualizations calculated on RoBERTa-base. It is clear RoBERTa is much more robust in making predictions but both DistilBERT and RoBERTa are susceptible to such attacks on their interpretations. [Best viewed in color]