On the Sensitivity and Stability of Model Interpretations

Anonymous ACL submission

Abstract

Recent years have witnessed the emergence of a variety of post-hoc interpretations that aim to uncover how natural language processing (NLP) models make predictions. Despite the surge of new interpretation methods, it remains an open problem how to define and quantitatively measure the *faithfulness* of interpretations, i.e., to what extent interpretations reflect the reasoning process by a model. We propose two new criteria, sensitivity and stability, that provide complementary notions of faithfulness to the existed removal-based criteria. Our results show that the conclusion for how faithful interpretations are could vary substantially based on different notions. Motivated by the desiderata of sensitivity and stability, we introduce a new class of interpretation methods that adopt techniques from adversarial robustness. Empirical results show that our proposed methods are effective under the new criteria and overcome limitations of gradient-based methods on removal-based criteria. Besides text classification, we also apply interpretation methods and metrics to dependency parsing. Our results shed light on understanding the diverse set of interpretations.

1 Introduction

As complex NLP models are widely deployed in real-world applications, there is an increasing interest in understanding how these models come to certain decisions. As a result, the line of research on interpretation techniques grows rapidly, facilitating a broad range of model analysis, from building user trust on models (Ribeiro et al., 2016; Hase and Bansal, 2020) to exposing subtle biases (Zhao et al., 2017; Doshi-Velez and Kim, 2017).

In this paper, we focus on *post-hoc interpretations* in NLP. Given a trained model and a specific input text, post-hoc interpretations assign an importance score to each token in the input to indicate its contribution to the model output. Current methods in this direction can be roughly divided into three categories: gradient-based methods (Simonyan et al., 2014; Li et al., 2016); reference-based methods (Sundararajan et al., 2017; Shrikumar et al., 2017); and perturbation-based methods (Zeiler and Fergus, 2014; Ribeiro et al., 2016).

Despite the emergence of new techniques, one critical issue is that there is little consensus on how to define and evaluate the faithfulness of these techniques, i.e., whether they reflect the true reasoning process by a model. A widely employed criterion, especially in NLP, is the *removal-based* criterion (DeYoung et al., 2020), which removes or only preserves a set of tokens given by interpretations and measures how much the model prediction would change. However, as pointed out in prior work (Bastings and Filippova, 2020; Ancona et al., 2018), the corrupted version of an input produced during evaluations falls out of the distribution that models are trained on, and thus results in an inaccurate measurement of faithfulness. This limitation prevents removal-based metrics from being used as the golden standard for evaluating interpretations. To remedy this, we complement the removal-based criterion with two other criteria, *sensitivity* and *stability*, which are overlooked in prior works.

Sensitivity is based on the notion that models should be more sensitive to perturbations on tokens identified by a faithful explanation. In contrast to the removal-based criterion, which completely removes important tokens, the sensitivity criterion adds small but adversarial perturbations in a local region of the token embedding, and thus preserves the structure of input sentences as well as interactions between context words. This criterion is recently discussed in Hsieh et al. (2020) in computer vision, while we provide comprehensive analyses on NLP models. Note that while the removal-based criterion asks the question: *if some important tokens did not 'exist', what would happen*, the sensitivity criterion asks: *if some important tokens were* *'changed' adversarially, what would happen.*

Stability assumes that a faithful interpretation should not produce substantially different explanations for two inputs that the model finds similar. There are several attempts to generate such a pair of inputs. The most relevant one is Ghorbani et al. (2019). However, their method is only applicable to differentiable interpretations. Our work proposes a new paradigm that employs a black-box algorithm to generate a semantically related neighbor of the original input, which is specially designed for NLP and applicable to all interpretations techniques.

Experiments show that interpretations which perform well on the removal-based criterion might not do well on the new criteria. Motivated by the limitations of existing interpretations and the desiderata of the two criteria, we propose *robustness-based methods*, based on projected gradient descent (PGD) attacks (Madry et al., 2018) and certifying robustness (Jia et al., 2019; Huang et al., 2019; Shi et al., 2020; Xu et al., 2020). We demonstrate that the new methods achieve top performance under sensitivity and stability. Moreover, as a simple improvement to gradient-based methods, our methods avoid the gradient saturation issues of gradient-based methods under the removal-based criterion.

Existing works in model interpretations often conduct experiments only on text classification due to the limitation of removal-based criteria – when input tokens are removed, the tree structure is drastically changed and a model might not be able to produce a meaningful parse tree. In this paper, we propose a new paradigm to interpret dependency parsers leveraging prepositional phrase (PP) attachment ambiguity examples. We demonstrate that sensitivity does not have the above restriction when evaluating dependency parsing explanations and conduct experiments to evaluation interpretation methods with our proposed paradigm and metrics.

Our contributions can be summarized as follows.

1. We discuss two overlooked notions of faithfulness in NLP interpretations. We propose quantitative criteria and systematically evaluate interpretations under these notions, including an existed removal-base one.
2. We propose new interpretation methods, which draw the connection between the adversarial robustness domain and the interpretation domain. We demonstrate the effectiveness of these new methods.
3. We propose a new paradigm to evaluate interpretations on the dependency parsing task.

2 Faithfulness Evaluation Criteria

A faithful post-hoc interpretation identifies the important parts of the input a model prediction relies on. Let $x = [x_1; x_2; \ldots ; x_n]$ be a sequence of tokens. $e(\cdot)$ denotes the token embedding function. An NLP model f takes the embedding matrix $e(x) \in \mathcal{R}^{n\times d}$ as input and provides its prediction $f(e(x)) = y$. Let $s_y(e(x))$ denote the output score of $f(e(x))$ on y. The exact form of $s_y(e(x))$ is defined in Appendix D. An interpretation assigns an importance score to each token to indicate its contribution to the model decision.

We first review the well-established *removal-based criterion* and emphasize its relation to the two criteria defined in this paper 1) *sensitivity*, and 2) *stability*, for which we propose novel paradigms to adapt them to various NLP tasks.

Removal-based Criterion A well-established notion of interpretation faithfulness is that the presence of important tokens should have more meaningful influence on the model's decision than random tokens, quantified by the removal-based criterion. We adopt the *comprehensiveness* and the *sufficiency* score in DeYoung et al. (2020). The comprehensiveness score measures how much the model performance would drop after the set of "relevant" tokens identified by an interpretation is removed. A higher comprehensiveness score suggests the tokens are more influential to the model output, and thus a more faithful explanation. The sufficiency score measures to what extent the original model performance is maintained when we solely preserve relevant tokens. A lower sufficiency score means less change in the model prediction, and thus a more faithful explanation. See DeYoung et al. (2020) for detailed definitions. Note that completely removing input tokens produces incomplete texts. Large perturbation of this kind lead to several issues as pointed out by prior studies (Feng et al., 2018; Bastings and Filippova, 2020).

Ours: Sensitivity Instead of removing important tokens, the sensitivity criterion adds *local* but adversarial noise to embedding vectors of the important tokens and measures the magnitude of the noise needed to change the model prediction. This is inspired by the notion that models should be more sensitive to perturbations being added to relevant tokens compared to random or irrelevant tokens. From the adversarial robustness perspective (Hsieh et al., 2020), this notion implies that by perturbing the most relevant tokens, we can reach the local decision boundary of a model with the minimum perturbation magnitude.

Given the sequence of relevant tokens r_k, sensitivity adds perturbation to its embedding $e\left(r_k\right)$ but keeps the remaining token embeddings unchanged. Then, it measures the minimal perturbation norm, denoted as ϵ_{r_k}, that changes the model prediction for this instance:

$$\epsilon_{r_k} = \min \|\boldsymbol{\delta}_{\boldsymbol{r_k}}\|_F \quad \text{s.t.} \quad f\left(e\left(x\right) + \boldsymbol{\delta}_{\boldsymbol{r_k}}\right) \neq y,$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, and $\boldsymbol{\delta}_{\boldsymbol{r_k}} \in \mathcal{R}^{n \times d}$ denotes the perturbation matrix where only the columns for tokens in r_k have non-zero elements. Since the exact computation of ϵ_{r_k} is intractable, we use the PGD attack (Madry et al., 2018) with a binary search to approximate ϵ_{r_k}. A lower ϵ_{r_k} suggests a more faithful interpretation. In practice, we vary the size of r_k, compute multiple ϵ_{r_k}, and summarize them with the area under the curve (AUC) score.

Ours: Stability Another desired property of faithfulness is that a faithful interpretation should not give substantially different importance orders for two input points that the model finds similar. To construct a pair of similar inputs, we propose to generate contrast examples to the original one by synonym substitutions. A contrast example of x, $\tilde{x}$, satisfies (1) has at most k different but synonymous tokens with x; (2) the prediction score at $\tilde{x}$ changes less than τ compared to the score at x. The goal of these two conditions is to generate (almost) natural examples where the changes of model outputs are smaller than a threshold τ. Given all contrast examples, we search for the one that leads to the largest rank difference $\mathcal{D}$ between the importance order for x, $m\left(x\right)$ and the alternated order $m\left(\tilde{x}\right)$:

$$\begin{gathered} \arg\max_{\tilde{x}} \mathcal{D}\left(m\left(x\right), m\left(\tilde{x}\right)\right), \\ \text{s.t. } \left|s_y\left(e\left(x\right)\right) - s_y\left(e\left(\tilde{x}\right)\right)\right| \leq \tau,\ \|x - \tilde{x}\|_0 \leq k. \end{gathered}$$

Specifically, we first extract synonyms for each token x_i following Alzantot et al. (2018). Then, in the decreasing order of $m\left(x\right)$, we greedily search for a substitution of each token that induces the largest change in $m\left(x\right)$ and repeat this process until the model output score changes by more than τ or the pre-defined constraint k is reached. Finally, we measure the difference $\mathcal{D}$ between two importance ranks using Spearman's rank order correlation (Spearman, 1961). We call this criterion *stability*. A higher score indicates that the ranks between this input pair are more similar, and thus a more faithful interpretation.

Note that instead of using the gradient information of interpretation methods to perturb importance ranks like Ghorbani et al. (2019), our algorithm treats interpretations as black-boxes, which makes it applicable to non-differentiable ones. Also, compared to Ding and Koehn (2021), who manually construct similar input pairs, our method is a fully automatic one as suggested by their paper.

3 Interpretations via Adversarial Robustness Techniques

Experiments indicate that existing methods do not work well with the sensitivity and stability metrics (Sec. 4.2). In this section, we define a new class of interpretation methods by adopting techniques in adversarial robustness to remedy this. We first give a brief review of existing interpretation approaches and then introduce our new methods.

3.1 Existing Interpretation Methods

We roughly divide the existing methods into three categories: *gradient-based methods*, *reference-based methods*, and *perturbation-based methods*, and discuss the representatives of them.

Gradient-based methods The first class of methods leverage the gradient at each input token. To aggregate the gradient vector at each token into a single importance score, we consider two methods: 1) using the L_2 norm, $\left\|\frac{\partial s_y(e(x))}{\partial e(x_i)}\right\|_2$, referred to as **Vanilla Gradient** (VaGrad) (Simonyan et al., 2014), and 2) using the dot product of gradient and input, $\left(\frac{\partial s_y(e(x))}{\partial e(x_i)}\right)^\top \cdot e\left(x_i\right)$, referred to as **Gradient · Input** (GradInp) (Li et al., 2016).

Reference-based methods These methods distribute the difference between model outputs on a reference point and on the input as the importance score for each token. We consider **Integrated Gradient** (IngGrad) (Sundararajan et al., 2017) and **DeepLIFT** (Shrikumar et al., 2017). IngGrad computes the linear intergral of the gradients from the reference point to the input. DeepLIFT decomposes the difference between each neuron activation and its 'reference activation' and back-propagates it to each input token. We use DeepLIFT with the Rescale rule. Note DeepLIFT diverges from IngGrad when multiplicative interactions among tokens exist (Ancona et al., 2018).

Perturbation-based methods Methods in this class query model outputs on perturbed inputs. We choose **Occlusion** (Zeiler and Fergus, 2014) and **LIME** (Ribeiro et al., 2016). Occlusion replaces one token at a time by a reference value and uses the corresponding drop on model performance to represent the importance of each token. LIME uses a linear model to fit model outputs on the neighborhood of input x and represents token importance by the weights in the trained linear model.

3.2 Proposed Robustness-based Methods

We propose two methods inspired from the PGD attack (Madry et al., 2018) and the certifying robustness algorithms (Xu et al., 2020) in adversarial robustness.

VaPGD and PGDInp The PGD attack in adversarial robustness considers a small vicinity of the input and takes several "mini-steps" within the vicinity to search for an adversarial example. Consider the token embeddings for the input x, we perform t iterations of the standard PGD procedure starting from $e^{(0)} = e\left(x\right)$:

$$e^{(j)} = \mathcal{P}\left(e^{(j-1)} - \alpha \nabla s_y\left(e^{(j-1)}\right)\right), \; j = 1, 2, \ldots, t.$$

$\mathcal{P}$ represents the operation that projects the new instance at each step back to the vicinity of $e\left(x\right)$, and α is the step size.

Intuitively, $e^{(t)} - e\left(x\right)$ tells us the descent direction of model confidence. Similar to the gradient-based methods, the importance of each token x_i can be either represented by $\left\|e_i^{(t)} - e\left(x_i\right)\right\|_2$, where $e_i^{(t)}$ is the i-th column in $e^{(t)}$, referred to as **Vanilla PGD** (VaPGD), or by $\left(e\left(x_i\right) - e_i^{(t)}\right)^\top \cdot e\left(x_i\right)$, referred to as **PGD** · **Input** (PGDInp)

Note that different from the PGD attack we use for approximating the sensitivity criterion, we manually decide the magnitude of the vicinity of $e\left(x\right)$ instead of using a binary search. We add perturbations to the whole sentence at the same time. Also, the final $e^{(t)}$ does not necessarily change the model prediction. See Appendix B for details.

Certify Certifying robustness algorithms also consider a vicinity of the original input and aim to provide guaranteed lower and upper bounds of a model output within that region. We use the linear relaxation based perturbation analysis (LiRPA) discussed in (Shi et al., 2020; Xu et al., 2020). LiRPA looks for two linear functions that bound the model. Specifically, LiRPA computes $\overline{\boldsymbol{W}}$, $\underline{\boldsymbol{W}}$, $\overline{\boldsymbol{b}}$, and $\underline{\boldsymbol{b}}$ that satisfy $\sum_i \underline{\boldsymbol{W_i}} e\left(x_i'\right) + \underline{\boldsymbol{b}} \leq s_y\left(e\left(x'\right)\right) \leq \sum_i \overline{\boldsymbol{W_i}} e\left(x_i'\right) + \overline{\boldsymbol{b}}$ for any point $e\left(x'\right)$ that lies within the L_2 ball of $e\left(x\right)$ with size δ. We use the IBP+backward method in Xu et al. (2020). It uses Interval Bound Propagation (Gowal et al., 2018; Mirman et al., 2018) to compute bounds of internal neurons of the model and then constructs the two linear functions with a bound back-propagation process (Zhang et al., 2018; Singh et al., 2019). Finally, the importance score of the i-th token in the input is represented by $\underline{\boldsymbol{W_i}} \cdot e\left(x_i\right)$, where $\underline{\boldsymbol{W_i}}$ is the i-th row of $\underline{\boldsymbol{W}}$. We call this method **Certify**.

Robustness-based vs. Gradient-based Gradient-based methods provide a linear approximation of the model decision boundary at the single input, which is not accurate for non-linear models. Robustness-based methods instead search multiple steps in neighbors and approximate the steepest descent direction better. We also empirically show that robustness-based methods avoid the saturation issue of gradient-based methods, i.e, gradient becomes zero at some inputs. See Appendix H. Note that VaPGD (PGDInp) degrades to VaGrad (GradInp) when the number of iterations is 1.

Robustness-based vs. IngGrad IngGrad leverages the average gradient in a segment between the input and a reference. It is likely to neglect local properties desired by the sensitivity criterion. Robustness-based methods instead search in the vicinity of the input, and thus local properties are better preserved. See results in Sec. 4.2.

4 Experiments on Text Classification

In this section, we present the results on text classification tasks under the three criteria. We find that the correlation between interpretation faithfulness based on different criteria are relatively low in some cases. Results verify the effectiveness of our new methods.

4.1 Experimental Setup

Datasets We conduct experiments on three text classification datasets: SST-2 (Socher et al., 2013), Yelp (Zhang et al., 2015), and AGNews (Zhang et al., 2015) following Jain and Wallace (2019)'s preprocessing approach. All of them are converted to binary classification tasks. SST-2 and Yelp are sentiment classification tasks where models predict whether a review is *negative* (0) or *positive* (1). AGNews is to discriminate between *world* (0) and *business* (1) articles. See Appendix A for statistics of the three datasets. When evaluating interpretation methods, for each dataset, we select 200 random samples (100 samples from class 0 and 100 samples from class 1) from the test set.

Models For text classification, we consider two model architectures: BERT (Devlin et al., 2019) and BiLSTM (Hochreiter and Schmidhuber, 1997).

Interpretation Methods Besides our robustness-based interpretations **PGDInp**, **VaPGD**, and **Certify**, we experiment with 6 others from three existing categories: **VaGrad**, **GradInp** (gradient-based); **IngGrad**, **DeepLIFT** (reference-based); and **Occlusion**, **LIME** (perturbation-based). We also include a random baseline **Random** that randomly assigns importance scores. We use comprehensiveness (**Comp.**), sufficiency (**Suff.**), sensitivity (**Sens.**), and stability (**Stab.**) metrics.

See Appendix A∼C for experimental details.

4.2 Results and Discussion

Overall Results Results of interpretations for BERT and BiLSTM are presented in Table 1 and 2. The interpretations’ performance are averaged over three runs on models trained from different random seeds. Results verify the effectiveness of our proposed robustness-based methods. Specifically, VaPGD achieves the best performance under the sensitivity and the stability criteria for both BERT and BiLSTM. Our methods also outperform their gradient-based counterparts under removal-based criteria. Especially, when interpreting BERT on SST-2 and AGNews, GradInp has near random performance. PGDInp can avoid these unreasonable behaviors. See Appendix H for a qualitative study on this, where we find PGDInp does not suffer from the saturation issue as GradInp.

However, the performance of other methods tend to be inconsistent under different measurements. For example, under the removal-based criterion, IngGrad performs well for BiLSTM, which gives four out of six best numbers. But, IngGrad has very limited performance under the sensitivity metric, especially for BiLSTM on SST-2 and Yelp. Similar issues exist for LIME and Occlusion. Also, one might fail to recognize the faithfulness of VaPGD by solely looking at the removal-based criterion. Thus, when deploying interpretation methods on real tasks, we advocate for a careful selection of the method you use based on the underlying faithfulness notion that aligned with your goal.

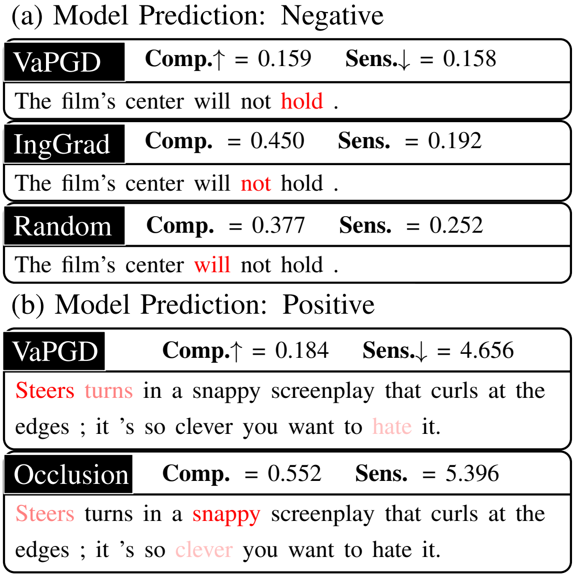

(a) Model Prediction: Negative
VaPGD
Comp.↑ = 0.159
Sens.↓ = 0.158
The film’s center will not hold .
IngGrad
Comp. = 0.450
Sens. = 0.192
The film’s center will not hold .
Random
Comp. = 0.377
Sens. = 0.252
The film’s center will not hold .
(b) Model Prediction: Positive
VaPGD
Comp.↑ = 0.184
Sens.↓ = 4.656
Steers turns in a snappy screenplay that curls at the edges ; it ’s so clever you want to hate it.
Occlusion
Comp. = 0.552
Sens. = 5.396
Steers turns in a snappy screenplay that curls at the edges ; it ’s so clever you want to hate it.

Figure 1: Two examples demonstrating different notions of faithfulness given by Comp. and Sens. A deeper red means the token is identified as more important. Comp. and Sens. scores are also shown.

Performance Curves We plot the curves of interpretations’ performance under the comprehensiveness and the sensitivity metrics when increasing the number of tokens being removed (perturbed). We use the case of interpreting BERT on Yelp as an example. Specifically, we collect two groups of examples from the test set of Yelp based on input lengths, where examples in each group are of 30 ± 5 and 120 ± 5 tokens long. We remove (perturb) the top-k most important tokens given by interpretations. Results are shown in Fig 2.

We observe that Occlusion is able to discover a smaller set of impactful tokens, under both metrics. However, when the size of the relevant set is increased, the performance of IngGrad under the comprehensiveness metric and the performance of VaPGD under the sensitivity metric gradually surpass Occlusion and other methods. This implies that the two methods could be better at identifying a relevant set containing more tokens.

Different Notions of Faithfulness We qualitatively study the two notions of faithfulness given by comprehensiveness (*comp.*) and sensitivity (*sens.*), and discuss two main differences.

First, *comp.* removes important tokens during evaluations, which could possibly break the interaction between removed tokens and context tokens, and underestimate the importance of context tokens. In example (a), the tokens ‘not’ and ‘hold’ together determine the negative sentiment of the sentence. *Sens.* considers both ‘not’ and ‘hold’ as

	SST-2				Yelp				AGNews			
Methods	**Comp.↑**	**Suff.↓**	**Sens.↓**	**Stab.↑**	**Comp.**	**Suff.**	**Sens.**	**Stab.**	**Comp.**	**Suff.**	**Sens.**	**Stab.**
Random	0.202	0.412	0.853	-0.343	0.166	0.383	1.641	-0.254	0.039	0.269	1.790	-0.392
VaGrad	0.371	0.286	0.546	0.850	0.273	0.254	1.034	0.798	0.251	0.113	1.041	0.843
GradInp	0.257	0.371	0.814	0.336	0.240	0.328	1.363	0.559	0.081	0.281	1.379	0.390
Occlusion	0.498	0.208	0.655	0.604	0.480	**0.192**	1.135	0.662	0.233	0.169	1.330	0.609
LIME	**0.562**	**0.208**	0.626	0.458	**0.511**	0.199	1.260	0.002	**0.461**	**0.063**	1.178	0.115
IngGrad	0.420	0.286	0.711	0.729	0.417	0.201	1.350	0.793	0.284	0.153	1.251	0.761
DeepLIFT	0.266	0.367	0.820	0.351	0.265	0.315	1.413	0.569	0.082	0.135	1.326	0.457
PGDInp	0.390	0.284	0.560	0.605	0.275	0.295	1.079	0.628	0.205	0.141	1.028	0.590
VaPGD	0.373	0.277	**0.542**	**0.853**	0.285	0.266	**1.022**	**0.832**	0.256	0.109	**0.995**	**0.869**

Table 1: Results of evaluating interpretations for BERT under three criteria on text classification datasets. ↑ means a higher number under this metric indicates a better performance. ↓ means the opposite. The best performance across all interpretations is **bolded**. *Certify* is missed here since current certifying robustness approaches cannot be scaled to deep Transformer-based models like BERT.

	SST-2				Yelp				AGNews			
Methods	**Comp.↑**	**Suff.↓**	**Sens.↓**	**Stab.↑**	**Comp.**	**Suff.**	**Sens.**	**Stab.**	**Comp.**	**Suff.**	**Sens.**	**Stab.**
Random	0.162	0.291	5.394	-0.316	0.035	0.217	14.242	-0.242	0.062	0.170	13.712	-0.378
VaGrad	0.196	0.256	3.448	0.860	0.139	0.108	9.438	0.887	0.061	0.187	10.485	0.812
GradInp	0.520	0.036	4.327	0.692	0.610	-0.057	11.719	0.810	0.345	0.006	13.286	0.773
Occlusion	0.595	-0.006	4.436	0.756	0.750	**-0.062**	11.725	0.816	0.513	-0.018	12.573	0.753
LIME	**0.609**	-0.001	4.367	0.563	0.378	0.013	12.504	0.137	0.591	-0.021	11.915	0.292
IngGrad	0.606	**-0.007**	4.500	0.767	**0.780**	-0.062	12.394	0.849	**0.657**	**-0.021**	12.608	0.815
DeepLIFT	0.538	0.024	4.404	0.669	0.637	-0.059	11.738	0.816	0.381	-0.014	12.146	0.735
PGDInp	0.548	0.008	4.228	0.713	0.663	-0.058	11.247	0.806	0.430	-0.006	11.302	0.794
VaPGD	0.229	0.214	**3.420**	**0.875**	0.166	0.094	**8.943**	**0.901**	0.113	0.113	**9.740**	**0.815**
Certify	0.524	0.038	4.317	0.692	0.612	-0.056	11.738	0.811	0.367	-0.011	12.143	0.778

Table 2: Results of evaluating different interpretation methods for BiLSTM. Same symbols as above.

important tokens as one expects. However, *comp.* regards 'hold' less important than 'will'.

Second, *sens.* measures token importance by how much model performance would change after 'adversarially perturbing' that token. In this sense, both positive and negative pertinent tokens will be deemed important. In contrast, *comp.* only considers positive pertinent ones. In example (b), which is predicted as positive, removing the negative verb 'hate' would not influence model performance much. However, adversarially perturbing 'hate' (e.g. change 'hate' to a more negative verb) might change the model prediction from positive to negative. Thus, *sens.* prefers interpretations that identify 'hate' as an important token like VaPGD.

The full version of example (b) is in Appendix E. Some contrast examples generated for the stability criterion are presented in Appendix F. We also justify that both *comp.* and *sens.* recognize the contribution of each individual token in a relevant token set instead of leveraging only parts of the relevant set. See Appendix I.

5 Experiments on Structured Prediction

	PTB-SD	
Method	**Comp.**	**Sens.**
Random	0.051	10.928
VaGrad	0.156	3.373
GradInp	0.152	5.257
IngGrad	0.190	4.315
DeepLIFT	0.153	5.252
Occlusion	0.194	4.671
LIME	**0.195**	4.529
PGDInp	0.163	4.704
VaPGD	0.157	**3.358**
Certify	0.155	4.701

Table 3: Evaluating interpretations for DeepBiaffine under the comprehensiveness and the sensitivity metric on the dependency parsing task.

Structured prediction tasks are in the center of NLP applications. However, applying interpretation methods and criteria to these tasks are difficult because 1) the required output is a structure instead of a single score. It is hard to define the contribution of each token to a structured output, and 2) compared to text classification tasks, removing

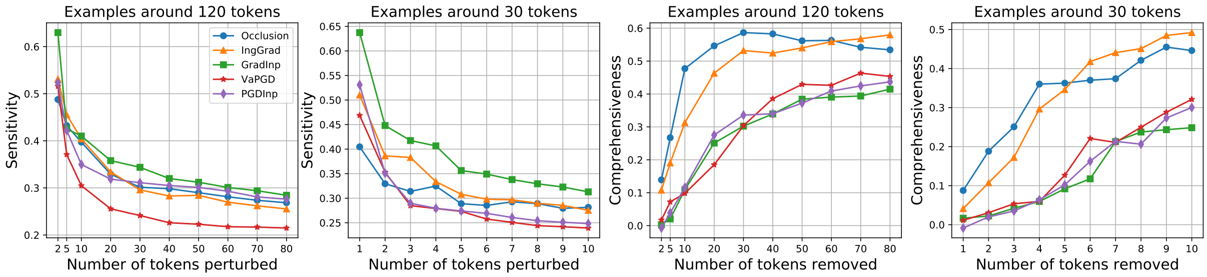


Figure 2: Evaluation curves of five interpretation methods. The title of each figure indicates the group of examples based on input lengths. The X-axis is the number of tokens being perturbed or removed for each instance, which varies in 1, 2, . . ., 10 for 30 tokens and 2, 5, 10, 20, . . ., 80 for 120 tokens. The Y-axis is the performance under the criterion. Results imply that IngGrad and VaPGD could be better at identifying a relevant set with more tokens.

parts of the input like what removal-based criteria do, would cause more drastic changes to model predictions as well as the groundtruth. Therefore, existing works often only conduct experiments on binary or multi-class text classification tasks. To remedy these issues, we investigate interpretations for dependency parsing, especially focus on analyzing how models resolve the PP attachment ambiguity to avoid interpreting the structured output as a whole. Then, we show that our sensitivity metric is more compatible with dependency parsing as it causes negligible changes to model outputs compared to removal-based metrics.

5.1 Evaluation Paradigm

Our paradigm focuses on the PP attachment ambiguity, which involves both syntactic and semantics considerations. A dependency parser needs to determine either the preposition in PP attaches to the preceding noun phrase NP (NP-attachment) or the verb phrase VP (VP-attachment) (Hindle and Rooth, 1993). The basic structure of ambiguity is VP – NP – PP. For example, in the sentence *I saw a cat with a telescope*, a parser uses the semantics of the noun phrase *a telescope* to predict the head of *with*, which is *saw*. If we change *a telescope* to *a tail*, the head of *with* would become the preceding noun *cat*. We will later call nouns in PPs like *telescope* “disambiguating nouns", as they provide semantic information for a parser to disambiguate PP attachment ambiguity. The main advantage of this paradigm is that disambiguating nouns can be viewed as “proxy groundtruths” for faithfulness as parsers must rely on them to make decisions.

Experimental Setup We use DeepBiaffine, a graph-based dependency parser as the target model (Dozat and Manning, 2017). We extract 100 examples that contain the PP attachment ambiguity from the English Penn Treebank converted to Stanford Dependencies 3.5.0 (PTB-SD). We consider the same interpretation methods as before, and they assign an importance score to each token in the sentence to indicate how much it impacts the model prediction on PP attachment arcs. We test the faithfulness of the attributions using comprehensiveness and sensitivity. See Appendix A∼C for details.

	PGD	Occlusion	IngGrad	GradInp
Comp.	0.82	0.81	0.81	0.79
Sens.	0.95	0.96	0.95	0.95

Table 4: Similarity between the parser outputs before and after applying the evaluation metric. We show that sensitivity changes the global model output less.

5.2 Results and Discussion

Results are shown in Table 3. Similar to the results on text classification tasks, we find that perturbation-based methods like LIME and Occlusion perform well on the comprehensiveness metric, while VaPGD performs the best under the sensitivity metric. PGDInp and Certify are slightly better than GradInp under both the two metrics.

Qualitatively, we find that according to interpretation methods, important tokens for a PP-attachment decision are often: the preposition itself, the preceding noun or verb, and the disambiguating noun. This is close to human expectations. An example is shown in Appendix E.

Metric Check Removing even a small piece of inputs breaks the dependency tree. It will be hard to distinguish either the decision process behind the model has changed or the removal of important tokens actually causes the performance drop. Thus, We expect a better metric to have less influence on the tree structure of a sentence. In Table 4, we show

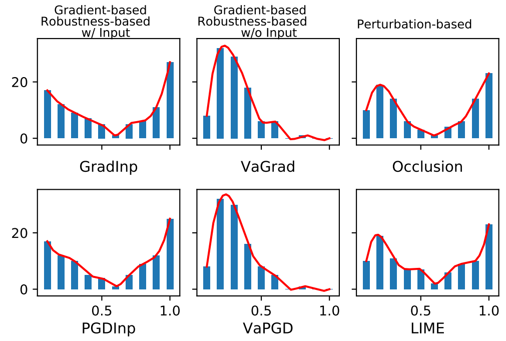


Figure 3: Where do interpretations place the disambiguating nouns. The results demonstrate obvious patterns in different categories. The X-axis is the top-k interval. Scales in {10%, 20%, . . ., 100%}. The Y-axis is the number of examples that an interpretation ranks the disambiguating noun within each top-k interval.

that evaluating interpretations with the sensitivity metric leads to smaller changes in the output dependency tree compared to the comprehensiveness, suggesting sensitivity a more compatible metric for the dependency parsing task interpretations.

Disambiguating Noun Analysis Disambiguating nouns are expected to be identified as important signals by faithful interpretations. We summarize how many times they are actually recognized as the top-k most important words by interpretation methods, where k is the interval varies in 10-20%, . . ., 90-100% of total tokens in an example.

Results in Figure 3 demonstrate that interpretation methods from the same category have high correlations when extracting disambiguating nouns. For example, VaGrad and VaPGD leveraging gradients only, tend to position disambiguating nouns on the top of their importance lists, which is consistent with human judgments. Likewise, the perturbation-based methods, Occlusion and LIME, also put the disambiguation words to very similar positions.

6 Related Work

Interpretation methods Various post-hoc interpretation methods are proposed to explain the behaviors of black-box models. These methods can be roughly categorized into three classes: gradient-based methods (Simonyan et al., 2014; Li et al., 2016), which leverage local gradient information; reference-based methods (Shrikumar et al., 2017; Sundararajan et al., 2017), which consider the model output difference between the original point and a reference point; and perturbation-based methods (Ribeiro et al., 2016; Zeiler and Fergus, 2014; Lundberg and Lee, 2017), which query model outputs on perturbed data. In our work, we propose new interpretation methods called robustness-based methods, which adopt techniques in the adversarial robustness domain and bridge the gap between the gradient-based and the reference-based methods.

Evaluating interpretation methods One line of studies explores approaches to evaluate interpretations. Several studies propose measurements for faithfulness. A large proportion of such them occlude tokens identified as important by interpretations and measure the performance changes of models (DeYoung et al., 2020; Jain and Wallace, 2019; Zaidan and Eisner, 2008; Serrano and Smith, 2019). Some other works propose to evaluate the faithfulness by checking to what extent they satisfy some desired axioms (Ancona et al., 2018; Sundararajan et al., 2017; Shrikumar et al., 2017). Besides, Alvarez-Melis and Jaakkola (2018); Ghorbani et al. (2019); Kindermans et al. (2019) reveal limitations in interpretation faithfulness through testing the robustness of interpretations. Another group of studies measure the plausibility of interpretations, i.e., whether the explanations conform with human judgments (Doshi-Velez and Kim, 2017; Ribeiro et al., 2016), or assist humans or student models to predict model behaviors on new data (Hase and Bansal, 2020; Pruthi et al., 2020). Note that although there exist many hybrid works that evaluate both the faithfulness and the plausibility of interpretations by combining a suite of diagnostic tests (DeYoung et al., 2020; Atanasova et al., 2020; Liu et al., 2020), Jacovi and Goldberg (2020) advocate to explicitly distinguish between the two measurements. In our work, we focus on interpretation faithfulness but consider two new metrics. We apply them to the dependency parsing task. Notice that the stability is an automatic input consistency tests suggested by Ding and Koehn (2021).

7 Conclusion

Our study shed a light on understanding interpretations. We studied interpretations under three criteria of faithfulness. We found that interpretations have inconsistent performance regarding different criteria. We proposed a new class of interpretations, which achieves the best performance under the sensitivity and the stability criteria. We further proposed a novel paradigm to evaluate interpretations on the dependency parsing task, which moves beyond text classification in the literature.

8 Ethical Considerations

This paper does not contain direct social influences. However, we believe the model analysis and interpretation techniques discussed in this paper are critical for deploying deep learning based models to real-world applications. Following previous work in this direction such as Jacovi and Goldberg (2020), we advocate to carefully consider the explanations obtained from interpretation methods as they may not always reflect the true reasoning process behind model predictions.

Besides the three notions of faithfulness discussed in this paper, there are other important aspects for measuring interpretations that could be applied to evaluate interpretations. Also, We are not claiming that the proposed paradigm are perfect as faithfulness measurements. For example, we recognize that it requires further and detailed analysis on either the model itself or the interpretation methods lead to a low performance on the *stability* metric, although we do try to make sure models behaviors do not change substantially between an input pair.

Moreover, experiments in this paper are all based on mainstream English corpora. Although our techniques are not language specific, there could be different conclusions given the varying properties of languages. For example, the discussion for dependency parsing could be easily affected by the language one considers.

References

David Alvarez-Melis and Tommi S Jaakkola. 2018. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *Arxiv*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Amirata Ghorbani, Abubakar Abid, and James Y. Zou. 2019. Interpretation of neural networks is fragile. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3681–3688. AAAI Press.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet

Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. 2020. Evaluations and methods for explanation through robustness analysis. *CoRR*, abs/2006.00442.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4083–4093.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 681–691. The Association for Computational Linguistics.

Ninghao Liu, Yunsong Meng, Xia Hu, Tie Wang, and Bo Long. 2020. Are interpretations fairly evaluated? a definition driven pipeline for post-hoc interpretability. *arXiv preprint arXiv:2009.07494*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Matthew Mirman, Timon Gehr, and Martin Vechev. 2018. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3575–3583.

Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. 2020. Evaluating explanations: How much do explanations from the teacher aid students? *arXiv preprint arXiv:2012.00893*.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. 2019. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41.

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.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 649–657.

Zheng Zhang, Pierre Zweigenbaum, and Ruiqing Yin. 2018. Efficient generation and processing of word co-occurrence networks using corpus2graph. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 7–11.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989.

A Dataset and Model Details

Datasets Statistics of the datasets are presented in Table 5.

Dataset	Train/Dev/Test	Avg Len
SST-2	67.3k/0.8k/1.8k	19.2
Yelp	447.9k/112.0k/1.2k	119.8
AGNews	51.0k/9.0k/3.8k	35.5
PTB-SD	39.8k/1.7k/2.4k	23.5

Table 5: Data Statistics

Models All models are implemented based on the PyTorch [1] library. All experiments are conducted on NVIDIA GeForce GTX 1080 Ti GPUs. For BERT, we use the bert-base-uncased model. We fine-tune BERT model on each dataset, using a unified setup: dropout rate 0.1, Adam (Kingma and Ba, 2015) with an initial learning rate of 1e-4, batch size 128, and no warm-up steps. We set the maximum number of fine-tuning to be 3. The fine-tuned BERT achieves 90.7, 95.4, and 96.9 accuracy on SST-2, Yelp and AGNews, respectively. When explaining BERT predictions, we only consider the contribution of word embeddings to the model output.

For BiLSTM classifier, we use an one-layer BiLSTM encoder with a linear classifier. The embedding is initialized with the 100-dimensional pretrained GloVe word embedding. We use Adam with an initial learning rate of 1e-3, batch size 512, hidden size 100 and dropout rate 0.2 for training. We set the maximum number of epochs to be 20 but perform early stopping when the performance on the development set doesn't improve for three epochs. Our BiLSTM classifier receives 84.2, 93.3, 95.9 accuracy on SST-2, Yelp and AGNews, respectively.

For DeepBiaffine, we simplify the original architecture by using a one-layer BiLSTM encoder and a biaffine classifier. The word embedding is also initialized with the pre-trained 100-dimensional

[1] https://pytorch.org/

GloVe word embedding while the part-of-speech tag embeddings are initialized to all zero. The encoder hidden size is 100. The arc and dependency relation hidden size are both 500. We get an UAS of 95.1 with our model. Note that for DeepBiaffine, each input token is represented by the concatenation of its word embedding and its part-of-speech tag embedding. When applying the interpretation methods and the evaluation metrics, we only modify the word embeddings but keep the part-of-speech tag embeddings unchanged.

B Interpretation Methods Details

For VaGrad, GradInp, VaPGD, PGDInp, and IngGrad, we use the automatic differentiation mechanism of PyTorch. For LIME, we modify the code from the original implementation of Ribeiro et al. (2016) [1]. For DeepLIFT, we use the implementation in Captum [2]. For Certify, we modify the code in auto_LiRPA [3].

For the two reference-based methods IngGrad and DeepLIFT, we use all zero word embeddings as the reference point. To approximate the integral in IngGrad, we sum up 50 points along the linear path from the reference point to the current point. For the perturbation-based methods LIME and Occlusion, we also set the word embedding of a token to an all zero embedding when it is perturbed.

Hyper-parameter tuning For all interpretations that require hyper-parameter tuning, including LIME, PGDInp, VaPGD, we randomly select 50 examples from the development set and choose the best hyperparameters based on the performance on these 50 examples. Specifically, the number of perturbed examples around the original point for LIME to fit a linear regression model is selected from {100, 200, 500, 800}. For PGDInp and VaPGD, we select the best maximum perturbation norm ϵ as for BERT and BiLSTM classifier from {0.1, 0.5, 1.2, 2.2}. We set the number of iterations as 50, and the step size as $\epsilon/5$. Note that we might be able to achieve better performance of VaPGD and PGDInp by also tuning the number of iterations and the step size. However, to keep the computational burden comparable with other interpretations, we do not tune these hyperparameters.

C Evaluation Criteria Details

Sensitivity Details We use PGD with a binary search for the minimal perturbation magnitude. In practice, we set the number of iterations to be 100 and the step size to be 1.0. Then, we conduct a binary search to estimate the smallest vicinity of the original point which contains an adversarial example that changes the model prediction.

Stability Details The synonyms in the stability metrics come from (Alzantot et al., 2018), where they extract nearest neighbors in the GloVe embeddings space and filter out antonyms with a counter-fitting method. We allow at most four tokens replaced by their synonyms for each input and at most 0.1 change in the output probability of the model prediction for BERT and 0.2 for BiLSTM.

Thresholds To compute the removal-based metrics and the AUC of sensitivity for text classification tasks, we vary the number of tokens being removed (preserved) or perturbed to be 10%, 20%, . . ., 50% of the total number of tokens in the input. For the dependency parsing task, the corresponding thresholds are 10%, 20% and 30%.

D Task Details

We evaluate the interpretation methods under both the text classification task and the dependency parsing task. Below, we cover implementation details for each task, respectively, including what is the specific model score interpretation methods explain, and what metrics we use for that task.

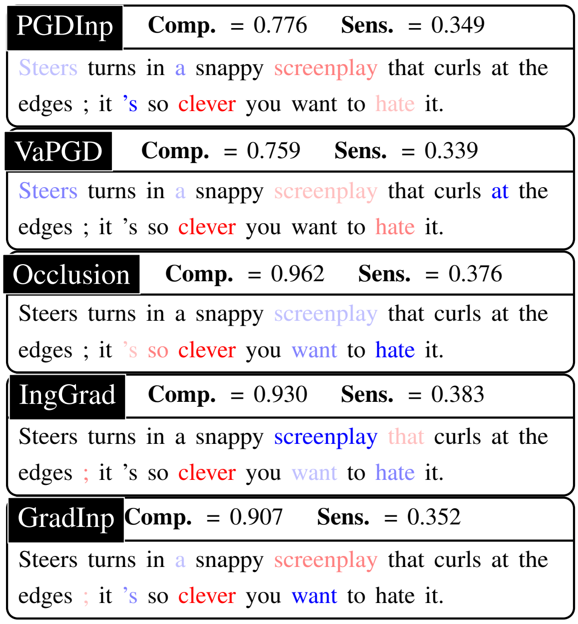
Text Classification Task $s_y(e(x))$ is the probability after the Softmax function corresponding to the original model prediction. We apply all the metrics mentioned in the main paper: removal-based metrics, including comprehensiveness and sufficiency scores, sensitivity score, and stability score. For removal-based metrics, we replace the important tokens with the pad token as a proxy for removing it.

Dependency Parsing Task $s(e(x))$ is the unlabeled arc log probability between the preposition and its head, i.e., unlabeled arc score after log_softmax, in the graph-based dependency parser. We discard the sufficiency score as it is unreasonable to remove a large proportion of tokens on a structured prediction task. We also discard the stability metric as there is little consensus on how to attack a structured model.

[1] https://github.com/marcotcr/lime

[2] https://github.com/pytorch/captum

[3] https://github.com/KaidiXu/auto_LiRPA



PGDInp **Comp.** = 0.776 **Sens.** = 0.349

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

VaPGD **Comp.** = 0.759 **Sens.** = 0.339

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

Occlusion **Comp.** = 0.962 **Sens.** = 0.376

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

IngGrad **Comp.** = 0.930 **Sens.** = 0.383

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

GradInp **Comp.** = 0.907 **Sens.** = 0.352

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

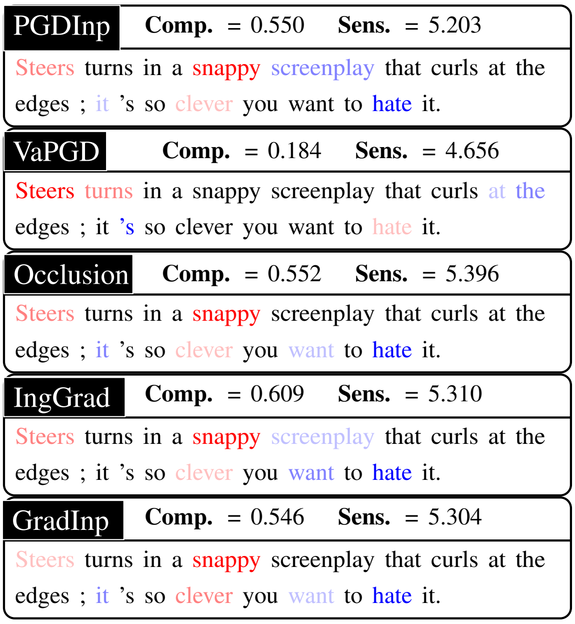
Figure 4: An example of interpreting BERT with five interpretation methods. A deeper red color means the token is identified as more important while a deeper blue color stands for a less important token. Performance under Comp. and Sens. scores are shown.

E An Example of Interpreting BERT and BiLSTM on the Text Classification Task

We showcase an example for interpreting BERT and BiLSTM in Figure 4 and 5. The example comes from the test set of SST-2. A deeper red color means the token is identified as more important to the model output by an interpretation while a deeper blue color stands for a less important token. Both the BiLSTM classifier and BERT classifier assign a positive label to this instance. Qualitatively, given an input, we observe that the most relevant or irrelevant sets of words identified by different interpretations are highly overlapped for BiLSTM, although the exact order of importance scores might be different. Whereas for BERT, different interpretations usually give different important tokens.

F An Example of Interpreting the Dependency Parser

An example of interpreting the PP attachment decision of a DeepBiaffine model. A deeper red color means the token is identified as more important for the model to predict the PP attachment arc.



PGDInp **Comp.** = 0.550 **Sens.** = 5.203

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

VaPGD **Comp.** = 0.184 **Sens.** = 4.656

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

Occlusion **Comp.** = 0.552 **Sens.** = 5.396

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

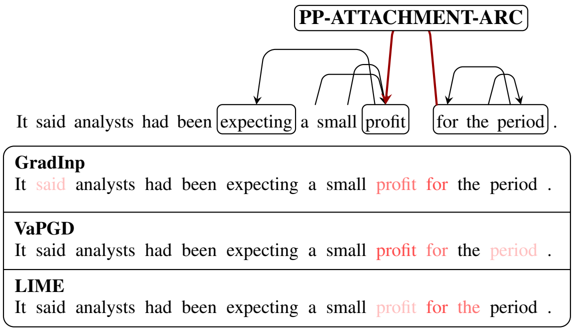
IngGrad **Comp.** = 0.609 **Sens.** = 5.310

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

GradInp **Comp.** = 0.546 **Sens.** = 5.304

Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it.

Figure 5: An example of interpreting BiLSTM using five interpretation methods.


PP-ATTACHMENT-ARC
It said analysts had been expecting a small profit for the period .

GradInp
It said analysts had been expecting a small profit for the period .

VaPGD
It said analysts had been expecting a small profit for the period .

LIME
It said analysts had been expecting a small profit for the period .

Figure 6: An example of interpreting the PP attachment arc in the dependency parsing task. A deeper red color means the token is identified as more important for the model to predict the PP attachment arc.

G Examples for the Stability Criterion

G.1 SST-2 Examples

Table 6 shows some contrast examples constructed for the stability criterion on SST-2.

G.2 AGNews Examples

Table 7 shows some contrast examples constructed for the stability criterion on AGNews

G.3 Yelp Examples

Table 8 shows some contrast examples constructed for the stability criterion on Yelp.

H Case Study on Gradient Saturation

We qualitatively study some cases where PGDInp does well under the removal-based criterion while

GradInp does not. In Figure 9, we show an example from explaining BERT on the SST-2 dataset, with the importance scores given by PGDInp, VaPGD, GradInp, VaGrad and the comprehensiveness score. For PGDInp and GradInp, we show the exponential of importance scores.

As shown in Figure 9, the importance score for each token given by GradInp is close to zero. VaGrad also gives near zero importance scores. At the same time, PGDInp and VaPGD have distinguishable and meaningful importance scores.

Based on the above observations, we suspect that the reason why PGD-based methods could avoid the failure of gradient-based methods is that they do not suffer from the gradient saturation issue. Gradient saturation refers to the cases where gradients are close to zero at some specific inputs and provide no information about the importance of different features of those inputs. Note that PGD-based methods consider not only a single input, but search on the vicinity of that input where the neighbors have none-zero gradients.

However, notice that VaGrad works better than GradInp. We suspect that is because although all elements in the gradient vector are close to zero, the **L-2 norm** of it is still distinguishable. However, GradInp takes the **dot-product** between embeddings and their gradients as the importance score. It is likely that negative and positive dimensions are neutralized, making the importance scores undistinguishable, and thus the behavior of GradInp corrupted. This hypothesis needs further explorations and demonstrations.

I Interpolation for Comprehensiveness and Sensitivity

We conduct interpolation between a relevant set given by a reasonable interpretation and a random set for the comprehensiveness and sensitivity scores. The goal of this experiment is to check whether these two metrics completely recognize the contribution of each token in the relevant set during evaluation, or partially consider some specific tokens. We select the best interpretation under these two metrics (LIME for comprehensiveness and VaPGD for sensitivity), respectively, and gradually replace each token in the relevant set with a random token outside of the set.

Specifically, we select 50 examples from SST-2 and test on BERT. For each example, we extract a relevant set consists of the top four important tokens given by an interpretation and gradually replace each token (from the least to the most important one) in the set with a random token. We denote the relevant set at each step as $S_0, S_1, ..., S_4$, where S_0 is the original relevant set containing the top four tokens and S_4 the set of four random tokens. The performance change at step i is represented by $f(i) = \frac{|M(S_0)-M(S_i)|}{|M(S_0)-M(S_4)|}$, where M is the comprehensiveness or sensitivity score. We expect that a good metric should induce a monotonic increasing function f. Further, f should be strictly convex as that indicates the importance of each token is different.

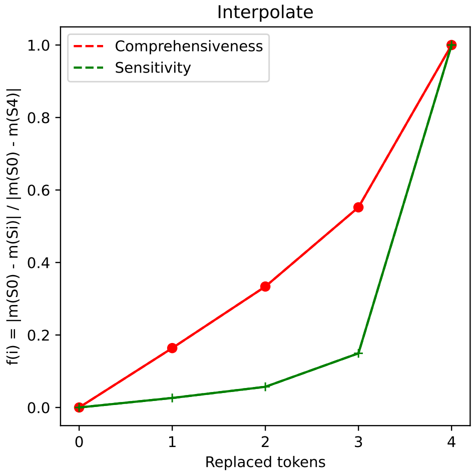


Figure 7: Interpolation between the relevant set and a random set.

We plot the curve in Figure 7. Results show that both the comprehensiveness and sensitivity metrics give out a monotonic increasing function, which indicates that both methods are completely considering each token in the relevant set and are ‘stable’ against interpolation. We also notice that based on the comprehensiveness metric, the influence of each token is more evenly distributed in the relevant set, while for the sensitivity metric, the most important toke has much higher influence than other tokens.

VaPGD, BERT on SST-2	
Rank correlation = 0.346	Model change = 0.00
Original	This is a film well worth seeing , talking and singing heads and all .
Contrast	This is a films well worth staring , talking and singing heads and entirety .

IngGrad, BERT on SST-2	
Rank correlation = 0.645	Model change = 0.15
Original	Ray Liotta and Jason Patric do some of their best work in their underwritten roles , but do n't be fooled : Nobody deserves any prizes here .
Contrast	Ray Liotta and Jason Patric do certain of their best collaborate in their underwritten roles , but do n't be fooled : Nobody deserves any awards here .

LIME, BiLSTM on SST-2	
Rank correlation = 0.425	Model change = 0.05
Original	Nearly surreal , dabbling in French , this is no simple movie , and you 'll be taking a risk if you choose to see it .
Contrast	Almost surreal , dabbling in French , this is no simple cinematography , and you 'll be taking a risk if you choose to seeing it .

Table 6: Generated contrast examples for evaluating the stability criterion on SST-2. Modified words are underlined. Spearman's rank correlation between a pair of examples and the performance difference of a model on the pair of examples are shown above each pair.

Erasure, BERT on AGNew	
Rank correlation = 0.689	Model change = 0.08
Original	Supporters and rivals warn of possible fraud ; government says chavez 's defeat could produce turmoil in world oil market .
Contrast	Supporters and rivals warn of possible fraud ; government says chavez 's defeat could produce disorder in planet oil trade .

DeepLIFT, BERT on AGNews	
Rank correlation = 0.317	Model change = 0.00
Original	Mills corp. agreed to purchase a qqq percent interest in nine malls owned by general motors asset management corp. for just over qqq billion , creating a new joint venture between the groups .
Contrast	Mills corp. agree to purchase a qqq percent interest in nine malls owned by comprehensive motors asset management corp. for just over qqq trillion , creating a new joint venture between the groups .

VaGrad, BERT on AGNews	
Rank correlation = 0.970	Model change = 0.12
Original	London (reuters) - oil prices surged to a new high of qqq a barrel on wednesday after a new threat by rebel militia against iraqi oil facilities and as the united states said inflation had stayed in check despite rising energy costs .
Contrast	london (reuters) - oil prices surged to a new high of qqq a canon on wednesday after a new menace by rebel militia against iraqi oil facilities and as the united states said inflation had stayed in check despite rising energy costs .

Table 7: Generated contrast examples for evaluating the stability criterion on AGNews.

PGD, BiLSTM on Yelp	
Rank correlation = 0.530	Model change = 0.00
Original	Love this beer distributor. They always have what I'm looking for. The workers are extremely nice and always willing to help. Best one I've seen by far.
Contrast	Love this beer distributor. They repeatedly have what I'm seeking for. The workers are extremely nice and always loan to help. Best one I've seen by far.

Certify, BiLSTM on Yelp	
Rank correlation = 0.633	Model change = 0.01
Original	Last summer I had an appointment to get new tires and had to wait a super long time. I also went in this week for them to fix a minor problem with a tire they put on. They "fixed" it for free, and the very next morning I had the same issue. I called to complain, and the "manager" didn't even apologize!!! So frustrated. Never going back. They seem overpriced, too.
Contrast	Last summer I took an appoints to get new tires and had to wait a super long time. I also went in this week for them to fix a minor problem with a tire they put on. They "fixed" it for free, and the very impending morning I had the same issue. I called to complain, and the "manager" didn't even apologize!!! So frustrated. Never going back. They seem overpriced, too.

Table 8: Generated contrast examples for evaluating the stability criterion on Yelp.

Example: A very funny movie .

		Importance Scores				
Method	**Comp.**	A	very	funny	movie	.
PGDInp	0.90	0.996	1.009	1.055	0.999	0.994
GradInp	0.33	1.000	1.000	1.000	1.000	1.000
VaPGD	0.67	0.072	0.124	0.399	0.199	0.079
VaGrad	0.54	0.000	0.001	0.001	0.001	0.000

Table 9: An example showing the gradient saturation issue. We show the importance score for each word given by the four interpretations and the corresponding comprehensiveness score. We find that while gradient-based methods suffer from the saturation issue, PGDInp and VaPGD could avoid the limitation.