Double Perturbation: On the Robustness of Robustness and Counterfactual Bias Evaluation

Chong Zhang   Jieyu Zhao   Huan Zhang   Kai-Wei Chang   Cho-Jui Hsieh
Department of Computer Science, UCLA
{chongz, jyzhao, kwchang, chohsieh}@cs.ucla.edu,   huan@huan-zhang.com

Abstract

Robustness and counterfactual bias are usually evaluated on a test dataset. However, are these evaluations robust? If the test dataset is perturbed slightly, will the evaluation results keep the same? In this paper, we propose a "double perturbation" framework to uncover model weaknesses beyond the test dataset. The framework first perturbs the test dataset to construct abundant natural sentences similar to the test data, and then diagnoses the prediction change regarding a single-word substitution. We apply this framework to study two perturbation-based approaches that are used to analyze models' robustness and counterfactual bias in English. (1) For robustness, we focus on synonym substitutions and identify vulnerable examples where prediction can be altered. Our proposed attack attains high success rates (96.0%–99.8%) in finding vulnerable examples on both original and robustly trained CNNs and Transformers. (2) For counterfactual bias, we focus on substituting demographic tokens (e.g., gender, race) and measure the shift of the *expected* prediction among constructed sentences. Our method is able to reveal the hidden model biases not directly shown in the test dataset. Our code is available at https://github.com/chong-z/nlp-second-order-attack.

1 Introduction

Recent studies show that NLP models are vulnerable to adversarial perturbations. A seemingly "invariance transformation" (a.k.a. adversarial perturbation) such as synonym substitutions (Alzantot et al., 2018; Zang et al., 2020) or syntax-guided paraphrasing (Iyyer et al., 2018; Huang and Chang, 2021) can alter the prediction. To mitigate the model vulnerability, robust training methods have been proposed and shown effective (Miyato et al., 2017; Jia et al., 2019; Huang et al., 2019; Zhou et al., 2020).
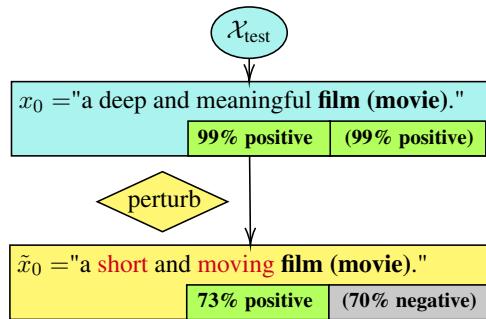

Figure 1: A vulnerable example beyond the test dataset. Numbers on the bottom right are the sentiment predictions for `film` and `movie`. Blue x_0 comes from the test dataset and its prediction cannot be altered by the substitution `film` → `movie` (robust). Yellow example $\tilde{x}_0$ is slightly perturbed but remains natural. Its prediction can be altered by the substitution (vulnerable).

In most studies, model robustness is evaluated based on a given test dataset or synthetic sentences constructed from templates (Ribeiro et al., 2020). Specifically, the robustness of a model is often evaluated by the ratio of test examples where the model prediction cannot be altered by semantic-invariant perturbation. We refer to this type of evaluations as the *first-order* robustness evaluation. However, even if a model is first-order robust on an input sentence x_0, it is possible that the model is not robust on a natural sentence $\tilde{x}_0$ that is slightly modified from x_0. In that case, adversarial examples still exist even if first-order attacks cannot find any of them from the given test dataset. Throughout this paper, we call $\tilde{x}_0$ a *vulnerable example*. The existence of such examples exposes weaknesses in models' understanding and presents challenges for model deployment. Fig. 1 illustrates an example.

In this paper, we propose the *double perturbation framework* for evaluating a stronger notion of *second-order robustness*. Given a test dataset, we consider a model to be *second-order* robust if there is no vulnerable example that can be identified in the *neighborhood* of given test instances

3899

(§2.2). In particular, our framework first perturbs the test set to construct the neighborhood, and then diagnoses the robustness regarding a single-word synonym substitution. Taking Fig. 2 as an example, the model is first-order robust on the input sentence x_0 (the prediction cannot be altered), but it is not second-order robust due to the existence of the vulnerable example $\tilde{x}_0$. Our framework is designed to identify $\tilde{x}_0$.

We apply the proposed framework and quantify second-order robustness through two *second-order attacks* (§3). We experiment with English sentiment classification on the SST-2 dataset (Socher et al., 2013) across various model architectures. Surprisingly, although robustly trained CNN (Jia et al., 2019) and Transformer (Xu et al., 2020) can achieve high robustness under strong attacks (Alzantot et al., 2018; Garg and Ramakrishnan, 2020) (23.0%–71.6% success rates), for around 96.0% of the test examples our attacks can find a vulnerable example by perturbing 1.3 words on average. This finding indicates that these robustly trained models, despite being first-order robust, are not second-order robust.

Furthermore, we extend the double perturbation framework to evaluate counterfactual biases (Kusner et al., 2017) (§4) in English. When the test dataset is small, our framework can help improve the evaluation robustness by revealing the hidden biases not directly shown in the test dataset. Intuitively, a fair model should make the same prediction for nearly identical examples referencing different groups (Garg et al., 2019) with different protected attributes (e.g., gender, race). In our evaluation, we consider a model *biased* if substituting tokens associated with protected attributes changes the *expected* prediction, which is the average prediction among all examples within the neighborhood. For instance, a toxicity classifier is biased if it tends to increase the toxicity if we substitute straight → gay in an input sentence (Dixon et al., 2018). In the experiments, we evaluate the expected sentiment predictions on pairs of protected tokens (e.g., (he, she), (gay, straight)), and demonstrate that our method is able to reveal the hidden model biases.

Our main contributions are: (1) We propose the double perturbation framework to diagnose the robustness of existing robustness and fairness evaluation methods. (2) We propose two second-order attacks to quantify the stronger notion of second-
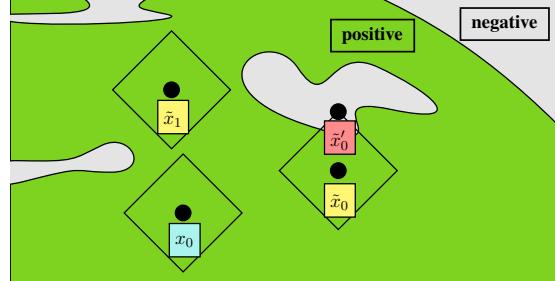

Figure 2: An illustration of the decision boundary. Diamond area denotes invariance transformations. Blue x_0 is a robust input example (the entire diamond is green). Yellow $\tilde{x}_0$ is a *vulnerable* example in the neighborhood of x_0. Red $\tilde{x}_0'$ is an *adversarial* example to $\tilde{x}_0$. Note: $\tilde{x}_0'$ is *not* an adversarial example to x_0 since they have different meanings to human (outside the diamond).

order robustness and reveal the models' vulnerabilities that cannot be identified by previous attacks. (3) We propose a counterfactual bias evaluation method to reveal the hidden model bias based on our double perturbation framework.

2 The Double Perturbation Framework

In this section, we describe the double perturbation framework which focuses on identifying vulnerable examples within a small neighborhood of the test dataset. The framework consists of a neighborhood perturbation and a word substitution. We start with defining word substitutions.

2.1 Existing Word Substitution Strategy

We focus our study on word-level substitution, where existing works evaluate robustness and counterfactual bias by directly perturbing the test dataset. For instance, adversarial attacks alter the prediction by making synonym substitutions, and the fairness literature evaluates counterfactual fairness by substituting protected tokens. We integrate the word substitution strategy into our framework as the component for evaluating robustness and fairness.

For simplicity, we consider a single-word substitution and denote it with the operator $\oplus$. Let $\mathcal{X} \subseteq \mathcal{V}^l$ be the input space where $\mathcal{V}$ is the vocabulary and l is the sentence length, $\boldsymbol{p} = (p^{(1)}, p^{(2)}) \in \mathcal{V}^2$ be a pair of synonyms (called *patch words*), $\mathcal{X}_{\boldsymbol{p}} \subseteq \mathcal{X}$ denotes sentences with a single occurrence of $p^{(1)}$ (for simplicity we skip other sentences), $x_0 \in \mathcal{X}_{\boldsymbol{p}}$ be an input sentence, then $x_0 \oplus \boldsymbol{p}$ means "substitute $p^{(1)} \rightarrow p^{(2)}$ in x_0". The result after substitution is:

$$x_0' = x_0 \oplus \boldsymbol{p}.$$

Taking Fig. 1 as an example, where $\boldsymbol{p} = $ (`film`, `movie`) and $x_0 = $ `a deep and meaningful film`, the perturbed sentence is $x_0' = $ `a deep and meaningful movie`. Now we introduce other components in our framework.

2.2 Proposed Neighborhood Perturbation

Instead of applying the aforementioned word substitutions directly to the original test dataset, our framework perturbs the test dataset within a small neighborhood to construct similar natural sentences. This is to identify vulnerable examples with respect to the model. Note that examples in the neighborhood are not required to have the same meaning as the original example, since we only study the prediction difference caused by applying synonym substitution $\boldsymbol{p}$ (§2.1).

Constraints on the neighborhood. We limit the neighborhood sentences within a small ℓ_0 norm ball (regarding the test instance) to ensure syntactic similarity, and empirically ensure the naturalness through a language model. The neighborhood of an input sentence $x_0 \in \mathcal{X}$ is:

$$\text{Neighbor}_k(x_0) \subseteq \text{Ball}_k(x_0) \cap \mathcal{X}_{\text{natural}}, \quad (1)$$

where $\text{Ball}_k(x_0) = \{x \mid \|x - x_0\|_0 \leq k, x \in \mathcal{X}\}$ is the ℓ_0 norm ball around x_0 (i.e., at most k different tokens), and $\mathcal{X}_{\text{natural}}$ denotes natural sentences that satisfy a certain language model score which will be discussed next.

Construction with masked language model. We construct neighborhood sentences from x_0 by substituting at most k tokens. As shown in Algorithm 1, the construction employs a recursive approach and replaces one token at a time. For each recursion, the algorithm first masks each token of the input sentence (may be the original x_0 or the $\tilde{x}$ from last recursion) separately and predicts likely replacements with a masked language model (e.g., DistilBERT, Sanh et al. 2019). To ensure the naturalness, we keep the top 20 tokens for each mask with the largest logit (subject to a threshold, Line 9). Then, the algorithm constructs neighborhood sentences by replacing the mask with found tokens. We use the notation $\tilde{x}$ in the following sections to denote the constructed sentences within the neighborhood.

Algorithm 1: Neighborhood construction

Data: Input sentence x_0, masked language model LM, max distance k.

1 **Function** Neighbor$_k$ (x_0):
2   **if** $k = 0$ **then return** $\{x_0\}$;
3   **if** $k \geq 2$ **then**
4   **return** $\bigcup_{\tilde{x} \in \text{Neighbor}_1(x_0)} \text{Neighbor}_{k-1}(\tilde{x})$;
5    $\mathcal{X}_{\text{neighbor}} \leftarrow \varnothing$;
6   **for** $i \leftarrow 0, \ldots, \text{len}(x_0) - 1$ **do**
7      $T, L \leftarrow \text{LM.fillmask}(x_0, i)$;
      ▷ Mask i_{th} token and return candidate tokens and corresponding logits.
8      $L \leftarrow \text{SortDecreasing}(L)$;
9      $l_{\min} \leftarrow \max\{L^{(\kappa)}, L^{(0)} - \delta\}$;
      ▷ $L^{(i)}$ denotes the i_{th} element. We empirically set $\kappa \leftarrow 20$ and $\delta \leftarrow 3$.
10      $T_{\text{new}} \leftarrow \{t \mid l > l_{\min}, (t, l) \in T \times L\}$;
11      $\mathcal{X}_{\text{new}} \leftarrow \{x_0 \mid x_0^{(i)} \leftarrow t, t \in T_{\text{new}}\}$;
      ▷ Construct new sentences by replacing the i_{th} token.
12      $\mathcal{X}_{\text{neighbor}} \leftarrow \mathcal{X}_{\text{neighbor}} \cup \mathcal{X}_{\text{new}}$;
13   **return** $\mathcal{X}_{\text{neighbor}}$;

---

3 Evaluating Second-Order Robustness

With the proposed double perturbation framework, we design two black-box attacks[1] to identify vulnerable examples within the neighborhood of the test set. We aim at evaluating the robustness for inputs beyond the test set.

3.1 Previous First-Order Attacks

Adversarial attacks search for small and invariant perturbations on the model input that can alter the prediction. To simplify the discussion, in the following, we take a binary classifier $f(x) : \mathcal{X} \rightarrow \{0, 1\}$ as an example to describe our framework. Let x_0 be the sentence from the test set with label y_0, then the smallest perturbation δ^* under ℓ_0 norm distance is:[2]

$$\delta^* := \underset{\delta}{\arg\min} \|\delta\|_0 \text{ s.t. } f(x_0 \oplus \delta) \neq y_0.$$

Here $\delta = \boldsymbol{p}_1 \oplus \cdots \oplus \boldsymbol{p}_l$ denotes a series of substitutions. In contrast, our second-order attacks fix $\delta = \boldsymbol{p}$ and search for the vulnerable x_0.

3.2 Proposed Second-Order Attacks

Second-order attacks study the prediction difference caused by applying $\boldsymbol{p}$. For notation convenience we define the prediction difference $F(x; \boldsymbol{p})$:

---

[1]Black-box attacks only observe the model outputs and do not know the model parameters or the gradient.
[2]For simplicity, we use ℓ_0 norm distance to measure the similarity, but other distance metrics can be applied.
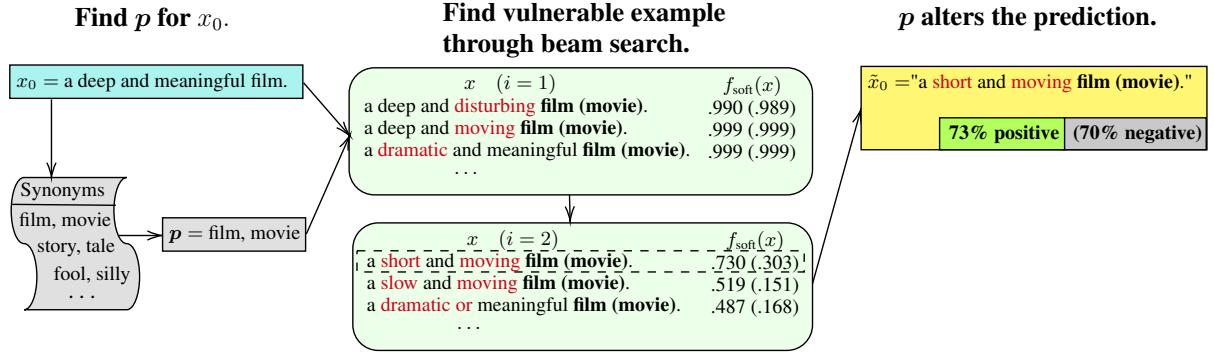
Figure 3: The attack flow for SO-Beam (Algorithm 2). Blue x_0 is the input sentence and yellow $\tilde{x}_0$ is our constructed vulnerable example (the prediction can be altered by substituting `film` → `movie`). Green boxes in the middle show intermediate sentences, and $f_{\text{soft}}(x)$ denotes the probability outputs for `film` and `movie`.

$\mathcal{X} \times \mathcal{V}^2 \to \{-1, 0, 1\}$ by:[3]

$$F(x; \boldsymbol{p}) := f(x \oplus \boldsymbol{p}) - f(x). \qquad (2)$$

Taking Fig. 1 as an example, the prediction difference for $\tilde{x}_0$ on $\boldsymbol{p}$ is $F(\tilde{x}_0; \boldsymbol{p}) = f(\texttt{...moving}$ $\textbf{movie.}) - f(\texttt{...moving } \textbf{film.}) = -1$.

Given an input sentence x_0, we want to find patch words $\boldsymbol{p}$ and a vulnerable example $\tilde{x}_0$ such that $f(\tilde{x}_0 \oplus \boldsymbol{p}) \neq f(\tilde{x}_0)$. Follow Alzantot et al. (2018), we choose $\boldsymbol{p}$ from a predefined list of counter-fitted synonyms (Mrkšić et al., 2016) that maximizes $|f_{\text{soft}}(p^{(2)}) - f_{\text{soft}}(p^{(1)})|$. Here $f_{\text{soft}}(x)$: $\mathcal{X} \to [0, 1]$ denotes probability output (e.g., after the softmax layer but before the final argmax), $f_{\text{soft}}(p^{(1)})$ and $f_{\text{soft}}(p^{(2)})$ denote the predictions for the single word, and we enumerate through all possible $\boldsymbol{p}$ for x_0. Let k be the neighborhood distance, then the attack is equivalent to solving:

$$\tilde{x}_0 = \underset{x \in \text{Neighbor}_k(x_0)}{\operatorname{argmax}} |F(x; \boldsymbol{p})|. \qquad (3)$$

Brute-force attack (*SO-Enum*). A naive approach for solving Eq. (3) is to enumerate through $\text{Neighbor}_k(x_0)$. The enumeration finds the smallest perturbation, but is only applicable for small k (e.g., $k \leq 2$) given the exponential complexity.
Beam-search attack (*SO-Beam*). The efficiency can be improved by utilizing the probability output, where we solve Eq. (3) by minimizing the cross-entropy loss with regard to $x \in \text{Neighbor}_k(x_0)$:

$$\mathcal{L}(x; \boldsymbol{p}) := -\log(1 - f_{\min}) - \log(f_{\max}), \qquad (4)$$

where $f_{\min}$ and $f_{\max}$ are the smaller and the larger output probability between $f_{\text{soft}}(x)$ and $f_{\text{soft}}(x \oplus$

---

[3] We assume a binary classification task, but our framework is general and can be extended to multi-class classification.

$\boldsymbol{p}$), respectively. Minimizing Eq. (4) effectively leads to $f_{\min} \to 0$ and $f_{\max} \to 1$, and we use a beam search to find the best x. At each iteration, we construct sentences through $\text{Neighbor}_1(x)$ and only keep the top 20 sentences with the smallest $\mathcal{L}(x; \boldsymbol{p})$. We run at most k iterations, and stop earlier if we find a vulnerable example. We provide the detailed implementation in Algorithm 2 and a flowchart in Fig. 3.

Algorithm 2: Beam-search attack (SO-Beam)

Data: Input sentence x_0, synonyms $\mathcal{P}$, model functions F and f_{soft}, loss $\mathcal{L}$, max distance k.

1 **Function** SO-Beam$_k$ (x_0):
2    $\boldsymbol{p} \leftarrow \underset{\boldsymbol{p} \in \mathcal{P} \text{ s.t. } x_0 \in \mathcal{X}_{\boldsymbol{p}}}{\operatorname{argmax}} |f_{\text{soft}}(p^{(2)}) - f_{\text{soft}}(p^{(1)})|$;
3    $\mathcal{X}_{\text{beam}} \leftarrow \{x_0\}$;
4   **for** $i \leftarrow 1, \ldots, k$ **do**
5      $\mathcal{X}_{\text{new}} \leftarrow \bigcup_{\tilde{x} \in \mathcal{X}_{\text{beam}}} \text{Neighbor}_1(\tilde{x})$;
6      $\tilde{x}_0 \leftarrow \operatorname{argmax}_{x \in \mathcal{X}_{\text{new}}} |F(x; \boldsymbol{p})|$;
7   **if** $F(\tilde{x}_0; \boldsymbol{p}) \neq 0$ **then return** $\tilde{x}_0$;
8      $\mathcal{X}_{\text{new}} \leftarrow \text{SortIncreasing}(\mathcal{X}_{\text{new}}, \mathcal{L})$;
9      $\mathcal{X}_{\text{beam}} \leftarrow \{\mathcal{X}_{\text{new}}^{(0)}, \ldots, \mathcal{X}_{\text{new}}^{(\beta-1)}\}$;
     ▷ Keep the best beam. We set $\beta \leftarrow 20$.
10   **return** None;

3.3 Experimental Results

In this section, we evaluate the second-order robustness of existing models and show the quality of our constructed vulnerable examples.

3.3.1 Setup

We follow the setup from the robust training literature (Jia et al., 2019; Xu et al., 2020) and experiment with both the base (non-robust) and robustly trained models. We train the binary sentiment classifiers on the SST-2 dataset with bag-of-words (BoW), CNN, LSTM, and attention-based

Original: 70% Negative	
Input Example:	in its best moments , resembles a bad high school production of grease , without benefit of song .
Genetic: 56% Positive	
Adversarial Example:	in its best moment , recalling a naughty high school production of lubrication , unless benefit of song .
BAE: 56% Positive	
Adversarial Example:	in its best moments , resembles a great high school production of grease , without benefit of song .
SO-Enum and SO-Beam (ours): 60% Negative (67% Positive)	
Vulnerable Example:	in its best moments , resembles a bad (unhealthy) high school production of musicals , without benefit of song .

Table 1: Sampled attack results on the robust BoW. For Genetic and BAE the goal is to find an *adversarial* example that alters the original prediction, whereas for SO-Enum and SO-Beam the goal is to find a *vulnerable* example beyond the test set such that the prediction can be altered by substituting `bad → unhealthy`.

models.

Base models. For BoW, CNN, and LSTM, all models use pre-trained GloVe embeddings (Pennington et al., 2014), and have one hidden layer of the corresponding type with 100 hidden size. Similar to the baseline performance reported in GLUE (Wang et al., 2019), our trained models have an evaluation accuracy of 81.4%, 82.5%, and 81.7%, respectively. For attention-based models, we train a 3-layer Transformer (the largest size in Shi et al. 2020) and fine-tune a pre-trained `bert-base-uncased` from HuggingFace (Wolf et al., 2020). The Transformer uses 4 attention heads and 64 hidden size, and obtains 82.1% accuracy. The BERT-base uses the default configuration and obtains 92.7% accuracy.

Robust models (first-order). With the same setup as base models, we apply robust training methods to improve the resistance to word substitution attacks. Jia et al. (2019) provide a provably robust training method through Interval Bound Propagation (IBP, Dvijotham et al. 2018) for all word substitutions on BoW, CNN and LSTM. Xu et al. (2020) provide a provably robust training method on general computational graphs through a combination of forward and backward linear bound propagation, and the resulting 3-layer Transformer is robust to up to 6 word substitutions. For both works we use the same set of counter-fitted synonyms provided in Jia et al. (2019). We skip BERT-base due to the lack of an effective robust training method.

Attack success rate (first-order). We quantify first-order robustness through attack success rate, which measures the ratio of test examples that an *adversarial* example can be found. We use first-order attacks as a reference due to the lack of a direct baseline. We experiment with two black-box attacks: (1) The Genetic attack (Alzantot et al., 2018; Jia et al., 2019) uses a population-based optimization algorithm that generates both syntactically and semantically similar adversarial examples, by replacing words within the list of counter-fitted synonyms. (2) The BAE attack (Garg and Ramakrishnan, 2020) generates coherent adversarial examples by masking and replacing words using BERT. For both methods we use the implementation provided by TextAttack (Morris et al., 2020).

Attack success rate (second-order). We also quantify second-order robustness through attack success rate, which measures the ratio of test examples that a *vulnerable* example can be found. To evaluate the impact of neighborhood size, we experiment with two configurations: (1) For the small neighborhood ($k = 2$), we use SO-Enum that finds the most similar vulnerable example. (2) For the large neighborhood ($k = 6$), SO-Enum is not applicable and we use SO-Beam to find vulnerable examples. We consider the most challenging setup and use patch words p from the same set of counter-fitted synonyms as robust models (they are provably robust to these synonyms on the test set). We also provide a random baseline to validate the effectiveness of minimizing Eq. (4) (Appendix A.1).

Quality metrics (perplexity and similarity). We quantify the quality of our constructed vulnerable examples through two metrics: (1) GPT-2 (Radford et al., 2019) perplexity quantifies the naturalness of a sentence (smaller is better). We report the perplexity for both the original input examples and the constructed vulnerable examples. (2) ℓ_0 norm distance quantifies the disparity between two sentences (smaller is better). We report the distance between the input and the vulnerable example. Note that first-order attacks have different objectives and thus cannot be compared directly.

	Attack Success Rate (%)			
	Genetic	BAE	SO-Enum	SO-Beam
Base Models:				
BoW	57.0	69.7	95.3	99.7
CNN	62.0	71.0	95.3	99.8
LSTM	60.0	68.3	95.8	99.5
Transformer	73.0	74.3	95.4	98.0
BERT-base	41.0	61.5	94.3	98.7
Robust Models:				
BoW	28.0	63.1	81.5	88.4
CNN	23.0	64.4	91.0	96.0
LSTM	24.0	61.0	62.9	77.5
Transformer	56.0	71.6	91.2	96.2

Table 2: The average rates over 872 examples (100 for Genetic due to long running time). Second-order attacks achieve higher successful rate since they are able to search beyond the test set.

	SO-Enum			**SO-Beam**		
	Original PPL	Perturb PPL	ℓ_0	Original PPL	Perturb PPL	ℓ_0
Base Models:						
BoW	168	202	1.1	166	202	1.2
CNN	170	204	1.1	166	201	1.2
LSTM	168	204	1.1	166	204	1.2
Transformer	165	193	1.0	165	195	1.1
BERT-base	170	229	1.3	168	222	1.4
Robust Models:						
BoW	170	212	1.2	171	222	1.4
CNN	166	209	1.2	168	210	1.3
LSTM	194	251	1.3	185	260	1.8
Transformer	170	213	1.2	165	208	1.3

Table 3: The quality metrics for second-order methods. We report the median perplexity (PPL) and average ℓ_0 norm distance. The original PPL may differ across models since we only count successful attacks.

3.3.2 Results

We experiment with the validation split (872 examples) on a single RTX 3090. The average running time per example (in seconds) on base LSTM is 31.9 for Genetic, 1.1 for BAE, 7.0 for SO-Enum ($k = 2$), and 1.9 for SO-Beam ($k = 6$). We provide additional running time results in Appendix A.3. Table 1 provides an example of the attack result where all attacks are successful (additional examples in Appendix A.5). As shown, our second-order attacks find a vulnerable example by replacing grease $\rightarrow$ musicals, and the vulnerable example has different predictions for bad and unhealthy. Note that, Genetic and BAE have different objectives from second-order attacks and focus on finding the adversarial example. Next we discuss the results from two perspectives.

Second-order robustness. We observe that existing robustly trained models are not second-order robust. As shown in Table 2, our second-order attacks attain high success rates not only on the base models but also on the robustly trained models. For instance, on the robustly trained CNN and Transformer, SO-Beam finds vulnerable examples within a small neighborhood for around 96.0% of the test examples, even though these models have improved resistance to strong first-order attacks (success rates drop from 62.0%–74.3% to 23.0%–71.6% for Genetic and BAE).[4] This phenomenon can be explained by the fact that both first-order attacks and robust training methods focus on synonym substitutions on the test set, whereas our attacks, due to their second-order nature, find vul-

nerable examples beyond the test set, and the search is not required to maintain semantic similarity. Our methods provide a way to further investigate the robustness (or find vulnerable and adversarial examples) even when the model is robust to the test set.

Quality of constructed vulnerable examples. As shown in Table 3, second-order attacks are able to construct vulnerable examples by perturbing 1.3 words on average, with a slightly increased perplexity. For instance, on the robustly trained CNN and Transformer, SO-Beam constructs vulnerable examples by perturbing 1.3 words on average, with the median[5] perplexity increased from around 165 to around 210. We provide metrics for first-order attacks in Appendix A.5 as they have different objectives and are not directly comparable.

Furthermore, applying existing attacks on the vulnerable examples constructed by our method will lead to much smaller perturbations. As a reference, on the robustly trained CNN, Genetic attack constructs adversarial examples by perturbing 2.7 words on average (starting from the input examples). However, if Genetic starts from our vulnerable examples, it would only need to perturb a single word (i.e., the patch words p) to alter the prediction. These results demonstrate the weakness of the models (even robustly trained) for those inputs beyond the test set.

3.3.3 Human Evaluation

We perform human evaluation on the examples constructed by SO-Beam. Specifically, we randomly

---

[4]BAE is more effective on robust models as it may use replacement words outside the counter-fitted synonyms.

[5]We report median due to the unreasonably large perplexity on certain sentences. e.g., 395 for that's a cheat. but 6740 for that proves perfect cheat.

Naturalness (1-5)		Semantic Similarity (%)	
Original	Perturb	Original	Perturb
3.87	3.63	85	71

Table 4: The quality metrics from human evaluation.

select 100 successful attacks and evaluate both the original examples and the vulnerable examples. To evaluate the naturalness of the constructed examples, we ask the annotators to score the likelihood (on a Likert scale of 1-5, 5 to be the most likely) of being an original example based on the grammar correctness. To evaluate the semantic similarity after applying the synonym substitution $\boldsymbol{p}$, we ask the annotators to predict the sentiment of each example, and calculate the ratio of examples that maintain the same sentiment prediction after the synonym substitution. For both metrics, we take the median from 3 independent annotations. We use US-based annotators on Amazon's Mechanical Turk[6] and pay $0.03 per annotation, and expect each annotation to take 10 seconds on average (effectively, the hourly rate is about $11). See Appendix A.2 for more details.

As shown in Table 4, the naturalness score only drop slightly after the perturbation, indicating that our constructed vulnerable examples have similar naturalness as the original examples. As for the semantic similarity, we observe that 85% of the original examples maintain the same meaning after the synonym substitution, and the corresponding ratio is 71% for vulnerable examples. This indicates that the synonym substitution is an invariance transformation for most examples.

4 Evaluating Counterfactual Bias

In addition to evaluating second-order robustness, we further extend the double perturbation framework (§2) to evaluate counterfactual biases by setting $\boldsymbol{p}$ to pairs of protected tokens. We show that our method can reveal the hidden model bias.

4.1 Counterfactual Bias

In contrast to second-order robustness, where we consider the model vulnerable as long as there exists *one* vulnerable example, counterfactual bias focuses on the *expected* prediction, which is the average prediction among all examples within the neighborhood. We consider a model biased if the
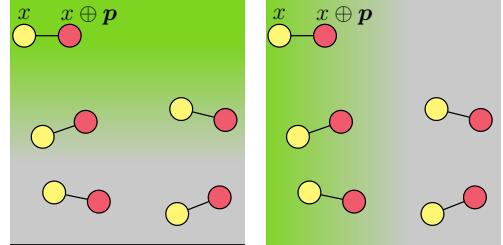

Figure 4: An illustration of an unbiased model vs. a biased model. Green and gray indicate the probability of positive and negative predictions, respectively. **Left**: An unbiased model where the $(x, x \oplus \boldsymbol{p})$ pair (yellow-red dots) is relatively parallel to the decision boundary. **Right**: A biased model where the predictions for $x \oplus \boldsymbol{p}$ (red) are usually more negative (gray) than x (yellow).

expected predictions for protected groups are different (assuming the model is not intended to discriminate between these groups). For instance, a sentiment classifier is biased if the expected prediction for inputs containing woman is more positive (or negative) than inputs containing man. Such bias is harmful as they may make unfair decisions based on protected attributes, for example in situations such as hiring and college admission.

Counterfactual token bias. We study a narrow case of counterfactual bias, where counterfactual examples are constructed by substituting protected tokens in the input. A naive approach of measuring this bias is to construct counterfactual examples directly from the test set, however such evaluation may not be robust since test examples are only a small subset of natural sentences. Formally, let $\boldsymbol{p}$ be a pair of protected tokens such as (he, she) or (Asian, American), $\mathcal{X}_{\text{test}} \subseteq \mathcal{X}_{\boldsymbol{p}}$ be a test set (as in §2.1), we define *counterfactual token bias* by:

$$B_{\boldsymbol{p},k} := \mathop{\mathbb{E}}_{x \in \text{Neighbor}_k(\mathcal{X}_{\text{test}})} F_{\text{soft}}(x; \boldsymbol{p}). \quad (5)$$

We calculate Eq. (5) through an enumeration across all natural sentences within the neighborhood.[7] Here $\text{Neighbor}_k(\mathcal{X}_{\text{test}}) = \bigcup_{x \in \mathcal{X}_{\text{test}}} \text{Neighbor}_k(x)$ denotes the union of neighborhood examples (of distance k) around the test set, and $F_{\text{soft}}(x; \boldsymbol{p}) : \mathcal{X} \times \mathcal{V}^2 \to [-1, 1]$ denotes the difference between probability outputs f_{soft} (similar to Eq. (2)):

$$F_{\text{soft}}(x; \boldsymbol{p}) := f_{\text{soft}}(x \oplus \boldsymbol{p}) - f_{\text{soft}}(x). \quad (6)$$

---

[7]For gender bias, we employ a blacklist to avoid adding gendered tokens during the neighborhood construction. This is to avoid semantic shift when, for example, $\boldsymbol{p} = $ (he, she) such that it may refer to different tokens after the substitution.

Patch Words	# Original	# Perturbed
he,she	5	325,401
his,her	4	255,245
him,her	4	233,803
men,women	3	192,504
man,woman	3	222,981
actor,actress	2	141,780
. . .		
Total	34	2,317,635

Table 5: The number of original examples ($k = 0$) and the number of perturbed examples ($k = 3$) in $\mathcal{X}_{\text{filter}}$.

The model is unbiased on $\boldsymbol{p}$ if $B_{\boldsymbol{p},k} \approx 0$, whereas a positive or negative $B_{\boldsymbol{p},k}$ indicates that the model shows preference or against to $p^{(2)}$, respectively. Fig. 4 illustrates the distribution of $(x, x \oplus \boldsymbol{p})$ for both an unbiased model and a biased model.

The aforementioned neighborhood construction does not introduce additional bias. For instance, let x_0 be a sentence containing he, even though it is possible for $\text{Neighbor}_1(x_0)$ to contain many stereotyping sentences (e.g., contains tokens such as doctor and driving) that affect the distribution of $f_{\text{soft}}(x)$, but it does not bias Eq. (6) as we only care about the prediction difference of replacing he $\rightarrow$ she. The construction has no information about the model objective, thus it would be difficult to bias $f_{\text{soft}}(x)$ and $f_{\text{soft}}(x \oplus \boldsymbol{p})$ differently.

4.2 Experimental Results

In this section, we use gender bias as a running example, and demonstrate the effectiveness of our method by revealing the hidden model bias. We provide additional results in Appendix A.4.

4.2.1 Setup

We evaluate counterfactual token bias on the SST-2 dataset with both the base and debiased models. We focus on binary gender bias and set $\boldsymbol{p}$ to pairs of gendered pronouns from Zhao et al. (2018a).
Base Model. We train a single layer LSTM with pre-trained GloVe embeddings and 75 hidden size (from TextAttack, Morris et al. 2020). The model has 82.9% accuracy similar to the baseline performance reported in GLUE.
Debiased Model. Data-augmentation with gender swapping has been shown effective in mitigating gender bias (Zhao et al., 2018a, 2019). We augment the training split by swapping all male entities with the corresponding female entities and vice-versa. We use the same setup as the base LSTM and attain 82.45% accuracy.
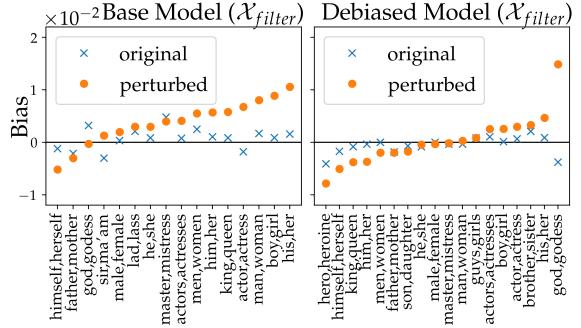

Figure 5: Our proposed $B_{\boldsymbol{p},k}$ measured on $\mathcal{X}_{\text{filter}}$. Here "original" is equivalent to $k = 0$, "perturbed" is equivalent to $k = 3$, $\boldsymbol{p}$ is in the form of (male, female).

Metrics. We evaluate model bias through the proposed $B_{\boldsymbol{p},k}$ for $k = 0, \ldots, 3$. Here the bias for $k = 0$ is effectively measured on the original test set, and the bias for $k \geq 1$ is measured on our constructed neighborhood. We randomly sample a subset of constructed examples when $k = 3$ due to the exponential complexity.
Filtered test set. To investigate whether our method is able to reveal model bias that was hidden in the test set, we construct a filtered test set on which the bias cannot be observed directly. Let $\mathcal{X}_{\text{test}}$ be the original validation split, we construct $\mathcal{X}_{\text{filter}}$ by the equation below and empirically set $\epsilon = 0.005$. We provide statistics in Table 5.

$$\mathcal{X}_{\text{filter}} := \{x \mid |F_{\text{soft}}(x; \boldsymbol{p})| < \epsilon, \ x \in \mathcal{X}_{\text{test}}\}.$$

4.2.2 Results

Our method is able to reveal the hidden model bias on $\mathcal{X}_{\text{filter}}$, which is not visible with naive measurements. In Fig. 5, the naive approach ($k = 0$) observes very small biases on most tokens (as constructed). In contrast, when evaluated by our double perturbation framework ($k = 3$), we are able to observe noticeable bias, where most $\boldsymbol{p}$ has a positive bias on the base model. This observed bias is in line with the measurements on the original $\mathcal{X}_{\text{test}}$ (Appendix A.4), indicating that we reveal the correct model bias. Furthermore, we observe mitigated biases in the debiased model, which demonstrates the effectiveness of data augmentation.

To demonstrate how our method reveals hidden bias, we conduct a case study with $\boldsymbol{p} = $ (actor, actress) and show the relationship between the bias $B_{\boldsymbol{p},k}$ and the neighborhood distance k. We present the histograms for $F_{\text{soft}}(x; \boldsymbol{p})$ in Fig. 6 and plot the corresponding $B_{\boldsymbol{p},k}$ vs. k in the right-most panel. Surprisingly, for the base model, the bias is
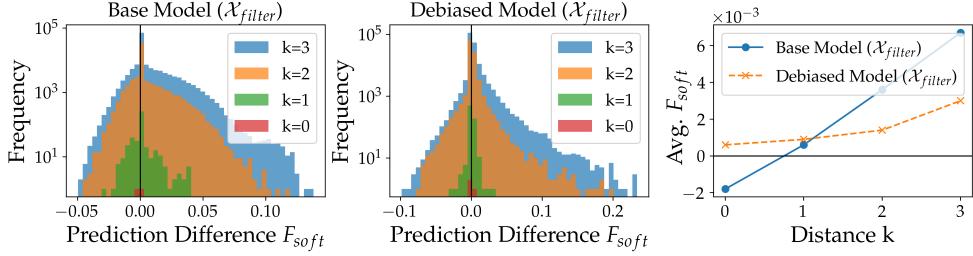
Figure 6: **Left and Middle:** Histograms for $F_{\text{soft}}(x; \boldsymbol{p})$ (x-axis) with $\boldsymbol{p} = (\texttt{actor}, \texttt{actress})$. **Right:** The plot for the average $F_{\text{soft}}(x; \boldsymbol{p})$ (i.e., counterfactual token bias) vs. neighborhood distance k. Results show that the counterfactual bias on $\boldsymbol{p}$ can be revealed when increasing k.

negative when $k = 0$, but becomes positive when $k = 3$. This is because the naive approach only has two test examples (Table 5) thus the measurement is not robust. In contrast, our method is able to construct 141,780 similar natural sentences when $k = 3$ and shifts the distribution to the right (positive). As shown in the right-most panel, the bias is small when $k = 1$, and becomes more significant as k increases (larger neighborhood). As discussed in §4.1, the neighborhood construction does not introduce additional bias, and these results demonstrate the effectiveness of our method in revealing hidden model bias.

5 Related Work

First-order robustness evaluation. A line of work has been proposed to study the vulnerability of natural language models, through transformations such as character-level perturbations (Ebrahimi et al., 2018), word-level perturbations (Jin et al., 2019; Ren et al., 2019; Yang et al., 2020; Hsieh et al., 2019; Cheng et al., 2020; Li et al., 2020), prepending or appending a sequence (Jia and Liang, 2017; Wallace et al., 2019a), and generative models (Zhao et al., 2018b). They focus on constructing adversarial examples from the test set that alter the prediction, whereas our methods focus on finding vulnerable examples beyond the test set whose prediction can be altered.

Robustness beyond the test set. Several works have studied model robustness beyond test sets but mostly focused on computer vision tasks. Zhang et al. (2019) demonstrate that a robustly trained model could still be vulnerable to small perturbations if the input comes from a distribution only slightly different than a normal test set (e.g., images with slightly different contrasts). Hendrycks and Dietterich (2019) study more sources of common corruptions such as brightness, motion blur and fog. Unlike in computer vision where simple image transformations can be used, in our natural language setting, generating a valid example beyond test set is more challenging because language semantics and grammar must be maintained.

Counterfactual fairness. Kusner et al. (2017) propose *counterfactual fairness* and consider a model fair if changing the protected attributes does not affect the distribution of prediction. We follow the definition and focus on evaluating the counterfactual bias between pairs of protected tokens. Existing literature quantifies fairness on a test dataset or through templates (Feldman et al., 2015; Kiritchenko and Mohammad, 2018; May et al., 2019; Huang et al., 2020). For instance, Garg et al. (2019) quantify the absolute counterfactual token fairness gap on the test set; Prabhakaran et al. (2019) study perturbation sensitivity for named entities on a given set of corpus. Wallace et al. (2019b); Sheng et al. (2019, 2020) study how language generation models respond differently to prompt sentences containing mentions of different demographic groups. In contrast, our method quantifies the bias on the constructed neighborhood.

6 Conclusion

This work proposes the double perturbation framework to identify model weaknesses beyond the test dataset, and study a stronger notion of robustness and counterfactual bias. We hope that our work can stimulate the research on further improving the robustness and fairness of natural language models.

Acknowledgments

Ethical Considerations

Intended use. One primary goal of NLP models is the generalization to real-world inputs. However, existing test datasets and templates are often not comprehensive, and thus it is difficult to evaluate real-world performance (Recht et al., 2019; Ribeiro et al., 2020). Our work sheds a light on quantifying performance for inputs beyond the test dataset and help uncover model weaknesses prior to the real-world deployment.

Misuse potential. Similar to other existing adversarial attack methods (Ebrahimi et al., 2018; Jin et al., 2019; Zhao et al., 2018b), our second-order attacks can be used for finding vulnerable examples to a NLP system. Therefore, it is essential to study how to improve the robustness of NLP models against second-order attacks.

Limitations. While the core idea about the double perturbation framework is general, in §4, we consider only binary gender in the analysis of counterfactual fairness due to the restriction of the English corpus we used, which only have words associated with binary gender such as `he/she`, `waiter/waitress`, etc.

References

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, Brussels, Belgium. Association for Computational Linguistics.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3601–3608.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. 2018. Training verified learners with learned verifiers.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 219–226, New York, NY, USA. Association for Computing Machinery.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *EACL*.

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. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. *Findings in EMNLP*.

Mohit Iyyer, J. Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *ArXiv*, abs/1804.06059.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 2021–2031. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *EMNLP/IJCNLP*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, New Orleans, Louisiana. Association for Computational Linguistics.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4066–4076. Curran Associates, Inc.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. 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 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. *ICLR*.

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 Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. 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 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do ImageNet classifiers generalize to ImageNet? volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400, Long Beach, California, USA. PMLR.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Association for Computational Linguistics (ACL)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *EMNLP*.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2020. Towards controllable biases in language generation. In *EMNLP-Finding*.

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *International Conference on Learning Representations*.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP/IJCNLP*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019b. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

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. Huggingface's transformers: State-of-the-art natural language processing.

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.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. 2020. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *Journal of Machine Learning Research*, 21(43):1–36.

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

Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, inderjit dhillon, and Cho-Jui Hsieh. 2019. The limitations of adversarial training and the blind-spot attack. In *International Conference on Learning Representations*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *NAACL*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018b. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627*.

A Supplemental Material

A.1 Random Baseline

To validate the effectiveness of minimizing Eq. (4), we also experiment on a second-order baseline that constructs vulnerable examples by randomly replacing up to 6 words. We use the same masked language model and threshold as SO-Beam such that they share a similar neighborhood. We perform the attack on the same models as Table 2, and the attack success rates on robustly trained BoW, CNN, LSTM, and Transformers are 18.8%, 22.3%, 15.2%, and 25.1%, respectively. Despite being a second-order attack, the random baseline has low attack success rates thus demonstrates the effectiveness of SO-Beam.

A.2 Human Evaluation

We randomly select 100 successful attacks from SO-Beam and consider four types of examples (for a total of 400 examples): The original examples with and without synonym substitution p, and the vulnerable examples with and without synonym substitution p. For each example, we annotate the naturalness and sentiment separately as described below.

Naturalness of vulnerable examples. We ask the annotators to score the likelihood of being an original example (i.e., not altered by computer) based on grammar correctness and naturalness, with a Likert scale of 1-5: (1) Sure adversarial example. (2) Likely an adversarial example. (3) Neutral. (4) Likely an original example. (5) Sure original example.

Semantic similarity after the synonym substitution. We first ask the annotators to predict the sentiment on a Likert scale of 1-5, and then map the prediction to three categories: negative, neutral, and positive. We consider two examples to have the same semantic meaning if and only if they are both positive or negative.

A.3 Running Time

We experiment with the validation split on a single RTX 3090, and measure the average running time per example. As shown in Table 6, SO-Beam runs faster than SO-Enum since it utilizes the probability output. The running time may increase if the model has improved second-order robustness.

	Running Time (seconds)			
	Genetic	BAE	SO-Enum	SO-Beam
Base Models:				
BoW	31.6	0.9	6.2	1.8
CNN	28.8	1.0	5.9	1.7
LSTM	31.9	1.1	7.0	1.9
Transformer	51.9	0.5	6.5	2.5
BERT-base	65.6	1.1	35.4	7.1
Robust Models:				
BoW	103.9	1.0	8.0	3.5
CNN	129.4	1.0	6.7	2.6
LSTM	116.4	1.1	10.7	5.3
Transformer	66.4	0.5	5.9	2.6

Table 6: The average running time over 872 examples (100 for Genetic due to long running time).

A.4 Additional Results on Protected Tokens

Fig. 7 presents the experimental results with additional protected tokens such as nationality, religion, and sexual orientation (from Ribeiro et al. (2020)). We use the same base LSTM as described in §4.2. One interesting observation is when $p = (\text{gay}, \text{straight})$ where the bias is negative, indicating that the sentiment classifier tends to give more *negative* prediction when substituting gay → straight in the input. This phenomenon is opposite to the behavior of toxicity classifiers (Dixon et al., 2018), and we hypothesize that it may be caused by the different distribution of training data. To verify the hypothesis, we count the number of training examples containing each word, and observe that we have far more negative examples than positive examples among those containing straight (Table 7). After looking into the training set, it turns out that straight to video is a common phrase to criticize a film, thus the classifier incorrectly correlates straight with negative sentiment. This also reveals the limitation of our method on polysemous words.

	# Negative	# Positive
gay	37	20
straight	71	18

Table 7: Number of negative and positive examples containing gay and straight in the training set.

In Fig. 8, we measure the bias on $\mathcal{X}_{\text{test}}$ and observe positive bias on most tokens for both $k = 0$ and $k = 3$, which indicates that the model "tends" to make more positive predictions for examples containing certain female pronouns than male pro-
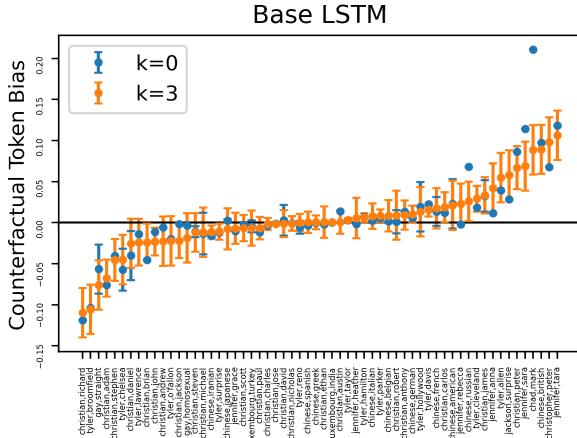
Figure 7: Additional counterfactual token bias measured on the original validation split with base LSTM.

nouns. Notice that even though gender swap mitigates the bias to some extent, it is still difficult to fully eliminate the bias. This is probably caused by tuples like (`him`, `his`, `her`) which cannot be swapped perfectly, and requires additional processing such as part-of-speech resolving (Zhao et al., 2018a).
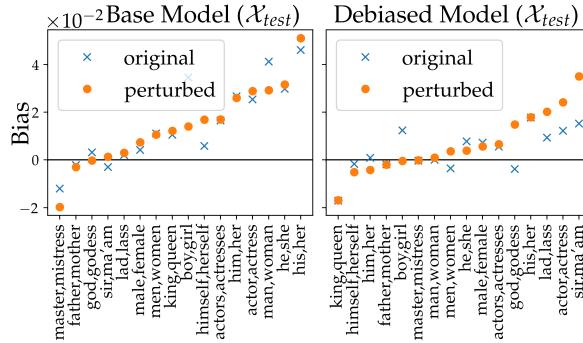
Figure 8: Full results for gendered tokens measured on the original validation split.

To help evaluate the naturalness of our constructed examples used in §4, we provide sample sentences in Table 9 and Table 10. Bold words are the corresponding patch words p, taken from the predefined list of gendered pronouns.

A.5 Additional Results on Robustness

Table 8 provides the quality metrics for first-order attacks, where we measure the GPT-2 perplexity and ℓ_0 norm distance between the input and the adversarial example. For BAE we evaluate on 872 validation examples, and for Genetic we evaluate on 100 validation examples due to the long running time.

Table 11 shows additional attack results from

	Genetic			BAE		
	Original PPL	Perturb PPL	ℓ_0	Original PPL	Perturb PPL	ℓ_0
Base Models:						
BoW	145	258	3.3	192	268	1.6
CNN	146	282	3.0	186	254	1.5
LSTM	131	238	2.9	190	263	1.6
Transformer	137	232	2.8	185	254	1.4
BERT-base	201	342	3.4	189	277	1.6
Robust Models:						
BoW	132	177	2.4	214	269	1.5
CNN	136	236	2.7	211	279	1.5
LSTM	163	267	2.5	220	302	1.6
Transformer	118	200	2.8	196	261	1.4

Table 8: The quality metrics for first-order attacks from successful attacks. We compare median perplexities (PPL) and average ℓ_0 norm distances.

SO-Beam on base LSTM, and Table 12 shows additional attack results from SO-Beam on robust CNN. Bold words are the corresponding patch words p, taken from the predefined list of counter-fitted synonyms.

3912

Type	Predictions		Text
Original	95% Negative	94% Negative	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** who is out of their depth .
Distance $k = 1$	97% Negative	(97% Negative)	it 's hampered by a lifetime-channel kind of plot and lone lead **actor (actress)** who is out of their depth .
	56% Negative	(55% Positive)	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** who is out of creative depth .
	89% Negative	(84% Negative)	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** who talks out of their depth .
	98% Negative	(98% Negative)	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** who is out of production depth .
	96% Negative	(96% Negative)	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** that is out of their depth .
Distance $k = 2$	88% Negative	(87% Negative)	it 's hampered by a lifetime-channel cast of stars and a lead **actor (actress)** who is out of their depth .
	96% Negative	(95% Negative)	it 's hampered by a simple set of plot and a lead **actor (actress)** who is out of their depth .
	54% Negative	(54% Negative)	it 's framed about a lifetime-channel kind of plot and a lead **actor (actress)** who is out of their depth .
	90% Negative	(88% Negative)	it 's hampered by a lifetime-channel mix between plot and a lead **actor (actress)** who is out of their depth .
	78% Negative	(68% Negative)	it 's hampered by a lifetime-channel kind of plot and a lead **actor (actress)** who storms out of their mind .
Distance $k = 3$	52% Positive	(64% Positive)	it 's characterized by a lifetime-channel combination comedy plot and a lead **actor (actress)** who is out of their depth .
	93% Negative	(93% Negative)	it 's hampered by a lifetime-channel kind of star and a lead **actor (actress)** who falls out of their depth .
	58% Negative	(57% Negative)	it 's hampered by a tough kind of singer and a lead **actor (actress)** who is out of their teens .
	70% Negative	(52% Negative)	it 's hampered with a lifetime-channel kind of plot and a lead **actor (actress)** who operates regardless of their depth .
	58% Negative	(53% Positive)	it 's hampered with a lifetime-channel cast of plot and a lead **actor (actress)** who is out of creative depth .

Table 9: Additional counterfactual bias examples on base LSTM with $p = (\texttt{actor}, \texttt{actress})$. We only present 5 examples per k due to space constrain.

Type	Predictions		Text
Original	55% Positive	(67% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator of an extended cheap shot across the mason-dixon line .
Distance $k = 1$	52% Positive	(66% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator of an extended cheap shot from the mason-dixon line .
	73% Positive	(79% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator gives an extended cheap shot across the mason-dixon line .
	56% Negative	(58% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator of an extended cheap shot across the phone line .
	75% Positive	(83% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator of an extended chase shot across the mason-dixon line .
	75% Positive	(81% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** our hapless facilitator of an extended cheap shot across the mason-dixon line .
Distance $k = 2$	85% Positive	(85% Positive)	a hilarious romantic comedy that makes our **boy (girl)** the hapless facilitator of an emotionally cheap shot across the mason-dixon line .
	81% Positive	(86% Positive)	a hamfisted romantic comedy romance makes our **boy (girl)** the hapless facilitator of an extended cheap delivery across the mason-dixon line .
	84% Positive	(87% Positive)	a hamfisted romantic romance adventure makes our **boy (girl)** the hapless facilitator of an extended cheap shot across the mason-dixon line .
	50% Negative	(62% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless boss of an extended cheap shot behind the mason-dixon line .
	77% Negative	(71% Negative)	a hamfisted lesbian comedy that makes our **boy (girl)** the hapless facilitator of an extended slap shot across the mason-dixon line .
Distance $k = 3$	97% Positive	(97% Positive)	a darkly romantic comedy romance makes our **boy (girl)** the hapless facilitator delivers an extended cheap shot across the mason-dixon line .
	69% Positive	(74% Positive)	a hamfisted romantic comedy film makes our **boy (girl)** the hapless facilitator of an extended cheap shot across the production line .
	87% Positive	(89% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the exclusive focus of an extended cheap shot across the mason-dixon line .
	64% Positive	(76% Positive)	a hamfisted romantic comedy that makes our **boy (girl)** the hapless facilitator shoots an extended flash shot across the camera line .
	99% Positive	(99% Positive)	a compelling romantic comedy that makes our **boy (girl)** the perfect facilitator of an extended story shot across the mason-dixon line .

Table 10: Additional counterfactual bias examples on base LSTM with $p =$ (boy, girl). We only present 5 examples per k due to space constrain.

Type	Predictions		Text
Original	99% Positive	(99% Positive)	it 's a charming and **sometimes (often)** affecting journey .
Vulnerable	59% Negative	(56% Positive)	it 's a charming and **sometimes (often)** painful journey .
Original	99% Negative	(97% Negative)	unflinchingly **bleak (somber)** and desperate
Vulnerable	80% Negative	(79% Positive)	unflinchingly **bleak (somber)** and mysterious
Original	99% Positive	(93% Positive)	allows us to hope that nolan is poised to embark a major **career (quarry)** as a commercial yet inventive filmmaker .
Vulnerable	76% Positive	(75% Negative)	allows us to hope that nolan is poised to embark a major **career (quarry)** as a commercial yet amateur filmmaker .
Original	94% Positive	(68% Positive)	the acting , costumes , music , cinematography and sound are all **astounding (staggering)** given the production 's austere locales .
Vulnerable	87% Positive	(66% Negative)	the acting , costumes , music , cinematography and sound are largely **astounding (staggering)** given the production 's austere locales .
Original	99% Positive	(97% Positive)	although laced with humor and a few fanciful touches , the film is a refreshingly serious look at **young (juvenile)** women .
Vulnerable	94% Positive	(81% Negative)	although laced with humor and a few fanciful touches , the film is a moderately serious look at **young (juvenile)** women .
Original	99% Negative	(98% Negative)	a **sometimes (occasionally)** tedious film .
Vulnerable	62% Negative	(55% Positive)	a **sometimes (occasionally)** disturbing film .
Original	100% Negative	(100% Negative)	in exactly 89 minutes , most of which passed as slowly as if i 'd been sitting naked on an igloo , formula 51 sank from **quirky (lunatic)** to jerky to utter turkey .
Vulnerable	51% Positive	(65% Negative)	lasting exactly 89 minutes , most of which passed as slowly as if i 'd been sitting naked on an igloo , but 51 ranges from **quirky (lunatic)** to delicious to crisp turkey .
Original	97% Positive	(100% Positive)	the **scintillating (mesmerizing)** performances of the leads keep the film grounded and keep the audience riveted .
Vulnerable	91% Negative	(90% Positive)	the **scintillating (mesmerizing)** performances of the leads keep the film grounded and keep the plot predictable .
Original	89% Negative	(96% Negative)	it takes a **uncanny (strange)** kind of laziness to waste the talents of robert forster , anne meara , eugene levy , and reginald veljohnson all in the same movie .
Vulnerable	80% Positive	(76% Negative)	it takes a **uncanny (strange)** kind of humour to waste the talents of robert forster , anne meara , eugene levy , and reginald veljohnson all in the same movie .
Original	100% Negative	(100% Negative)	... the film suffers from a lack of humor (something needed to **balance (equilibrium)** out the violence) ...
Vulnerable	76% Positive	(86% Negative)	... the film derives from a lot of humor (something clever to **balance (equilibrium)** out the violence) ...
Original	55% Positive	(97% Positive)	we root for (clara and paul) , even like them , though perhaps it 's an emotion closer to **pity (compassion)** .
Vulnerable	89% Negative	(91% Positive)	we root for (clara and paul) , even like them , though perhaps it 's an explanation closer to **pity (compassion)** .
Original	95% Negative	(97% Negative)	even horror **fans (stalkers)** will most likely not find what they 're seeking with trouble every day ; the movie lacks both thrills and humor .
Vulnerable	61% Positive	(59% Negative)	even horror **fans (stalkers)** will most likely not find what they 're seeking with trouble every day ; the movie has both thrills and humor .
Original	100% Positive	(100% Positive)	a gorgeous , high-spirited musical from india that exquisitely **mixed (blends)** music , dance , song , and high drama .
Vulnerable	87% Negative	(81% Positive)	a dark , high-spirited musical from nowhere that loosely **mixed (blends)** music , dance , song , and high drama .
Original	99% Negative	(94% Negative)	... the movie is just a plain **old (longtime)** monster .
Vulnerable	94% Negative	(94% Positive)	... the movie is just a pretty **old (longtime)** monster .

Table 11: Additional sentiment classification results from SO-Beam on base LSTM.

Type	Predictions		Text
Original	54% Positive	(69% Positive)	for the most part , director anne-sophie birot 's first feature is a sensitive , **overly (extraordinarily)** well-acted drama .
Vulnerable	53% Negative	(62% Positive)	for the most part , director anne-sophie benoit 's first feature is a sensitive , **overly (extraordinarily)** well-acted drama .
Original	66% Positive	(72% Positive)	mr. tsai is a very original **painter (artist)** in his medium , and what time is it there ?
Vulnerable	52% Negative	(55% Positive)	mr. tsai is a very original **painter (artist)** in his medium , and what time was it there ?
Original	80% Positive	(64% Positive)	sade is an **engaging (engage)** look at the controversial eponymous and fiercely atheistic hero .
Vulnerable	53% Positive	(66% Negative)	sade is an **engaging (engage)** look at the controversial eponymous or fiercely atheistic hero .
Original	50% Negative	(57% Negative)	so devoid of any kind of **comprehensible (intelligible)** story that it makes films like xxx and collateral damage seem like thoughtful treatises
Vulnerable	53% Positive	(54% Negative)	so devoid of any kind of **comprehensible (intelligible)** story that it makes films like xxx and collateral 2 seem like thoughtful treatises
Original	90% Positive	(87% Positive)	a tender , **heartfelt (deepest)** family drama .
Vulnerable	60% Positive	(61% Negative)	a somber , **heartfelt (deepest)** funeral drama .
Original	57% Positive	(69% Positive)	... a hollow **joke (giggle)** told by a cinematic gymnast having too much fun embellishing the misanthropic tale to actually engage it .
Vulnerable	56% Negative	(56% Positive)	... a hollow **joke (giggle)** told by a cinematic gymnast having too much fun embellishing the misanthropic tale cannot actually engage it .
Original	73% Negative	(56% Negative)	the **cold (colder)** turkey would 've been a far better title .
Vulnerable	61% Negative	(62% Positive)	the **cold (colder)** turkey might 've been a far better title .
Original	70% Negative	(65% Negative)	it 's just disappointingly superficial – a movie that has all the elements necessary to be a fascinating , involving character study , but never does more than scratch the **shallow (surface)** .
Vulnerable	52% Negative	(55% Positive)	it 's just disappointingly short – a movie that has all the elements necessary to be a fascinating , involving character study , but never does more than scratch the **shallow (surface)** .
Original	79% Negative	(72% Negative)	schaeffer has to find some hook on which to hang his persistently useless movies , and it might as well be the **resuscitation (revival)** of the middle-aged character .
Vulnerable	57% Negative	(57% Positive)	schaeffer has to find some hook on which to hang his persistently entertaining movies , and it might as well be the **resuscitation (revival)** of the middle-aged character .
Original	64% Positive	(58% Positive)	the primitive force of this film seems to bubble up from the vast collective memory of the **combatants (militants)** .
Vulnerable	52% Positive	(53% Negative)	the primitive force of this film seems to bubble down from the vast collective memory of the **combatants (militants)** .
Original	64% Positive	(74% Positive)	on this **troublesome (tricky)** topic , tadpole is very much a step in the right direction , with its blend of frankness , civility and compassion .
Vulnerable	55% Negative	(56% Positive)	on this **troublesome (tricky)** topic , tadpole is very much a step in the right direction , losing its blend of frankness , civility and compassion .
Original	74% Positive	(60% Positive)	if you 're **hard (laborious)** up for raunchy college humor , this is your ticket right here .
Vulnerable	60% Positive	(57% Negative)	if you 're **hard (laborious)** up for raunchy college humor , this is your ticket holder here .
Original	94% Positive	(97% Positive)	a fast , funny , highly **fun (enjoyable)** movie .
Vulnerable	54% Negative	(65% Positive)	a dirty , violent , highly **fun (enjoyable)** movie .
Original	86% Positive	(88% Positive)	good old-fashioned slash-and-hack is **back (backwards)** !
Vulnerable	52% Negative	(55% Positive)	a old-fashioned slash-and-hack is **back (backwards)** !

Table 12: Additional sentiment classification results from SO-Beam on robust CNN.