Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification

Yichao Zhou*, **Jyun-Yu Jiang***, **Kai-Wei Chang and Wei Wang**
Computer Science Department
University of California, Los Angeles
{yz, jyunyu, kwchang, weiwang}@cs.ucla.edu

Abstract

Adversarial attacks against machine learning models have threatened various real-world applications such as spam filtering and sentiment analysis. In this paper, we propose a novel framework, learning to discriminate perturbations (DISP), to identify and adjust malicious perturbations, thereby blocking adversarial attacks for text classification models. To identify adversarial attacks, a perturbation discriminator validates how likely a token in the text is perturbed and provides a set of potential perturbations. For each potential perturbation, an embedding estimator learns to restore the embedding of the original word based on the context and a replacement token is chosen based on approximate kNN search. DISP can block adversarial attacks for any NLP model without modifying the model structure or training procedure. Extensive experiments on two benchmark datasets demonstrate that DISP significantly outperforms baseline methods in blocking adversarial attacks for text classification. In addition, in-depth analysis shows the robustness of DISP across different situations.

1 Introduction

Deep learning techniques (Goodfellow et al., 2016) have achieved enormous success in many fields, such as computer vision and NLP. However, complex deep learning models are often sensitive and vulnerable to a tiny modification. In other words, malicious attackers can destroy the models by adding a few inconspicuous perturbations into input data, such as masking images with unrecognizable filters and making low-key modifications for texts. Therefore, developing techniques to equip models against adversarial attacks becomes a prominent research problem.

Existing studies on adversarial attacks can be classified into two groups, generation of adversarial examples and defense against adversarial attacks (Yuan et al., 2019). In the field of NLP, most of the existing studies focus on the former. For example, Ebrahimi et al. (2017); Alzantot et al. (2018) replace a word with synonyms or similar words while Gao et al. (2018); Liang et al. (2017); Ebrahimi et al. (2017) conduct character-level manipulations to fool the models. Moreover, it is not straightforward to adapt existing approaches for blocking adversarial attacks, such as data augmentation (Krizhevsky et al., 2012; Ribeiro et al., 2018; Ren et al., 2019) and adversarial training (Goodfellow et al., 2015; Iyyer et al., 2018; Marzinotto et al., 2019; Cheng et al., 2019; Zhu et al., 2019), to NLP applications. Hence, the defense against adversarial attacks in NLP remains a challenging and unsolved problem.

Recognizing and removing the inconspicuous perturbations are the core of defense against adversarial attacks. For instance, in computer vision, denoising auto-encoders (Warde-Farley and Bengio, 2017; Gu and Rigazio, 2015) are applied to remove the noises introduced by perturbations; Prakash et al. (2018) manipulate the images to make the trained models more robust to the perturbations; Samangouei et al. (2018) apply generative adversarial networks to generate perturbation-free images. However, all of these approaches cannot straightforwardly apply to the NLP tasks for the following two reasons. First, images consist of continuous pixels while texts are discrete tokens. As a result, a token can be replaced with another semantically similar token that drops the performance, so perturbations with natural looks cannot be easily recognized compared to previous approaches that capture unusual differences between the intensities of neighboring pixels. Second, sentences consist of words with an enormous

*Equal contribution. Listing order is random.

vocabulary size, so it is intractable to enumerate all of the possible sentences. Therefore, existing defense approaches in computer vision that rely on pixel intensities cannot be directly used for the NLP tasks.

After recognizing the perturbed tokens, the naïve way to eliminate the perturbations for blocking adversarial attacks is to remove these perturbed tokens. However, removing words from sentences results in fractured sentences, causing the performance of NLP models to degrade. Therefore, it is essential to recover the removed tokens. Nevertheless, training a satisfactory language model requires myriad and diverse training data, which is often unavailable. An inaccurate language model that incoherently patches missing tokens can further worsen the prediction performance. To tackle this difficult challenge, we propose to recover the tokens from discriminated perturbations by a masked language model objective with contextualized language modeling.

In this paper, we propose *Learning to Discriminate Perturbations* (DISP), as a framework for blocking adversarial attacks in NLP. More specifically, we aim to defend the model against adversarial attacks without modifying the model structure and the training procedure. DISP consists of three components, perturbation discriminator, embedding estimator, and hierarchical navigable small world graphs. Given a perturbed testing data, the perturbation discriminator first identifies a set of perturbed tokens. For each perturbed token, the embedding estimator optimized with a corpus of token embeddings infers an embedding vector to represent its semantics. Finally, we conduct an efficient kNN search over a hierarchical taxonomy to translate each of the embedding vectors into appropriate token to replace the associated perturbed word. We summarize our contributions in the following.

- To the best of our knowledge, this paper is the first work for blocking adversarial attacks in NLP without retraining the model.
- We propose a novel framework, DISP, which is effective and significantly outperforms other baseline methods in defense against adversarial attacks on two benchmark datasets.
- Comprehensive experiments have been conducted to demonstrate the improvements of DISP. In addition, we will release our implementations and the datasets to provide a testbed and facilitate future research in this direction.

2 Related Work

Adversarial examples crafted by malicious attackers expose the vulnerability of deep neural networks when they are applied to down-streaming tasks, such as image recognition, speech processing, and text classifications (Wang et al., 2019; Goodfellow et al., 2015; Nguyen et al., 2015; Moosavi-Dezfooli et al., 2017).

For adversarial attacks, white-box attacks have full access to the target model while black-box attacks can only explore the models by observing the outputs with limited trials. Ebrahimi et al. (2017) propose a gradient-based white-box model to attack character-level classifiers via an atomic flip operation. Small character-level transformations, such as swap, deletion, and insertion, are applied on critical tokens identified with a scoring strategy (Gao et al., 2018) or gradient-based computation (Liang et al., 2017). Samanta and Mehta (2017); Alzantot et al. (2018) replace words with semantically and syntactically similar adversarial examples.

However, limited efforts have been done on adversarial defense in the NLP fields. Texts as discrete data are sensitive to the perturbations and cannot transplant most of the defense techniques from the image processing domain such as Gaussian denoising with autoencoders (Meng and Chen, 2017; Gu and Rigazio, 2014). Adversarial training is the prevailing counter-measure to build a robust model (Goodfellow et al., 2015; Iyyer et al., 2018; Marzinotto et al., 2019; Cheng et al., 2019; Zhu et al., 2019) by mixing adversarial examples with the original ones during training the model. However, these adversarial examples can be detected and deactivated by a genetic algorithm (Alzantot et al., 2018). This method also requires retraining, which can be time and cost consuming for large-scale models.

Spelling correction (Mays et al., 1991; Islam and Inkpen, 2009) and grammar error correction (Sakaguchi et al., 2017) are useful tools which can block editorial adversarial attacks, such as swap and insertion. However, they cannot handle cases where word-level attacks that do not cause spelling and grammar errors. In our paper, we propose a general schema to block both word-level and character-level attacks.

3 DISP for Blocking Adversarial Attacks

In this section, we first formally define the goal of adversarial defense and then introduce the proposed framework DISP, learning to discriminate perturbations, for blocking adversarial attacks.

Problem Statement. Given an NLP model $F(X)$, where $X = \{t_1, \ldots, t_N\}$ is the input text of N tokens while t_i indicates the i-th token. A malicious attacker can add a few inconspicuous perturbations into the input text and generate an adversarial example X_a so that $F(X) \neq F(X_a)$ with unsatisfactory prediction performance. For example, a perturbation can be an insertion, a deletion of a character in a token, a replacement of a token with its synonym. In this paper, we aim to block adversarial attacks for general text classification models. More specifically, we seek to preserve the model performances by recovering original input text and universally improve the robustness of any text classification model.

3.1 Framework Overview

Figure 1 illustrates the overall schema of the proposed framework. DISP consists of three components, (1) a perturbation discriminator, (2) an embedding estimator, and (3) a token embedding corpus with the corresponding small world graphs G. In the training phase, DISP constructs a corpus D from the original corpus for training the perturbation discriminator so that it is capable of recognizing the perturbed tokens. The corpus of token embeddings C is then applied to train the embedding estimator to recover the removed tokens after establishing the small world graphs G of the embedding corpus. In the prediction phase, for each token in testing data, the perturbation discriminator predicts if the token is perturbed. For each potential perturbation that is potentially perturbed, the embedding estimator generates an approximate embedding vector and retrieve the token with the closest distance in the embedding space for token recovery. Finally, the recovered testing data can be applied for prediction. Note that the prediction model can be any NLP model.Moreover, DISP is a general framework for blocking adversarial attacks, so the model selection for the discriminator and estimator can also be flexible.

3.2 Perturbation Discrimination

Perturbation Discriminator. The perturbation discriminator plays an important role to classify whether a token t_i in the input X_a is perturbed based on its neighboring tokens. We adopt contextualized language modeling, such as BERT (Devlin et al., 2018), to derive d-dimension contextualized token representation T_i^D for each token t_i and then cascade it with a binary logistic regression classifier to predict if the token t_i is perturbed or not. Figure 2 illustrates the perturbation discriminator based on a contextualized word encoder. The discriminator classifies a token t_i into two classes $\{0, 1\}$ with logistic regression based on the contextual representation T_i^D to indicate if the token is perturbed. More formally, for each token t_i, the discriminator predictions r_i can then be derived as:

$$r_i = \underset{c}{\operatorname{argmax}}\, y_i^c = \underset{c}{\operatorname{argmax}} \left(\boldsymbol{w_c} \cdot T_i^D + b_c \right),$$

where y_i^c is the logit for the class c; $\boldsymbol{w_c}$ and b_c are the weights and the bias for the class c. Finally, the potential perturbations R is the set of tokens with positive discriminator predictions $R = \{t_i \mid r_i = 1\}$.

3.3 Efficient Token-level Recovery with Embedding Estimator

After predicting the perturbations R, we need to correct these disorders to preserve the prediction performance. One of the most intuitive approaches to recover tokens with context is to exploit language models. However, language models require sufficient training data while the precision to exact tokens can be dispensable for rescuing prediction performance. Moreover, over-fitting limited training data can be harmful to the prediction quality. To resolve this problem, we assume that replacing the perturbed word with a word with similar meanings to the original word is sufficient for the downstream models to make the correct prediction. Based on the assumption, DISP first predicts the embeddings of the recovered tokens for the potential perturbations with an embedding estimator based on context tokens. The tokens can then be appropriately recovered by an efficient k-nearest neighbors (kNN) search in the embedding space of a token embedding corpus C.

Embedding Estimator. Similar to the perturbation discriminator, any regression model can be employed as an embedding estimator based on the proposed concept. Here we adopt the contextualized language modeling again as an example of the embedding estimator. For each token t_i, the

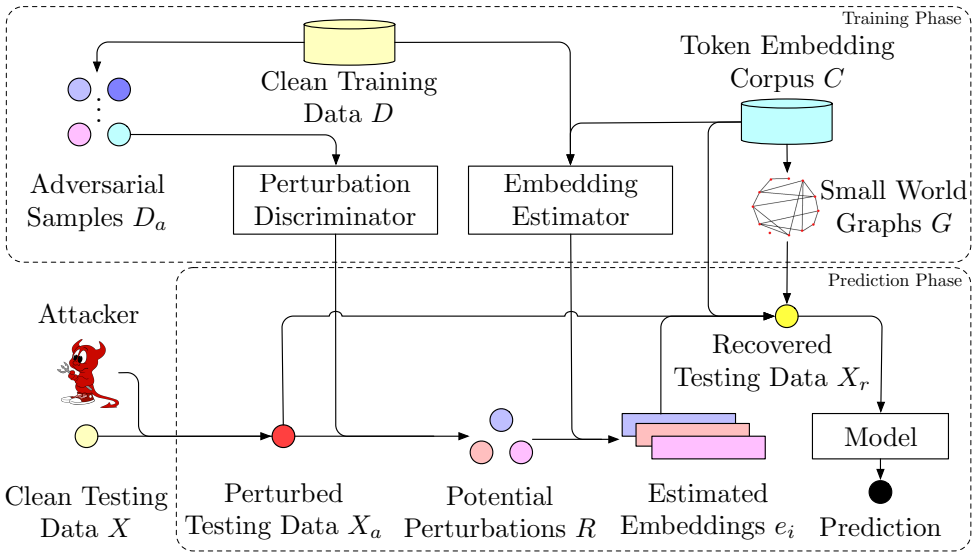


Figure 1: Schema of the proposed framework DISP.

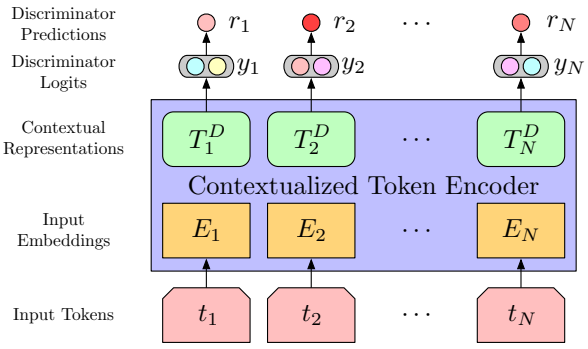


Figure 2: The illustration of the perturbation discriminator in DISP.

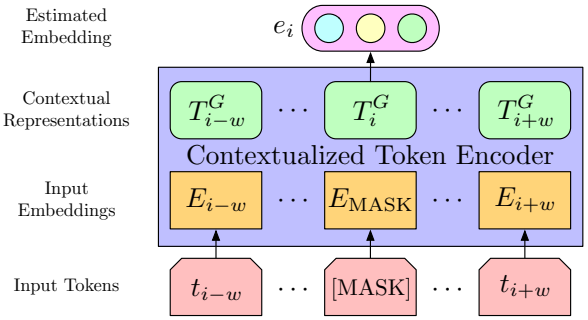


Figure 3: The illustration of the embedding estimator in DISP with a window size $2w+1$ for the token at the position i.

contextualized token embedding can be derived as a d-dimensional contextual representation vector T_i^G to be features for estimating appropriate embeddings.

Figure 3 shows the embedding estimator based on BERT. For each potential perturbation $t_i \in R$, $2w$ neighboring tokens are selected as the context for estimating the appropriate embedding, where w decides the window size. More precisely, a segment of tokens with a window size $2w+1$ from t_{i-w} to t_{i+w} is the input tokens for BERT, where t_i is replaced with a `[MASK]` token as the perturbed position. Finally, for the target t_i, a weight matrix $W^G \in \mathbb{R}^{d \times k}$ projects the contextual representation T_i^G to a k-dimensional estimated embedding e_i as follows:

$$e_i = T_i^G W^G,$$

where the dimension size k is required to be consistent with the embedding dimension in the token embedding corpus C.

Efficient Token-level Recovery. Finally, we recover the input sentence based on the predicted recover embeddings from the embedding estimator. Specifically, the input text X needs to be recovered from the perturbed text X_a by fixing token-level perturbations based on its approximate embeddings.

Given the token embedding corpus C, it is simple to transform an embedding to a token by finding the nearest neighbor token in the embedding space. However, a naïve kNN search query can take $O(kn)$ time complexity, where n is the number of embeddings in C; k is the embedding dimension. To accelerate the search process, we apply hierarchical navigable small world graphs (SWGs) (Malkov and Yashunin, 2018) for fast approximate kNN search. More precisely, em-

Algorithm 1: Efficient Token-level Recovery

Input: Perturbed text X_a, potential perturbations R, estimated embeddings $\{e_i\}$, small world graphs G, token embedding corpus C.
Output: Recovered text X_r.

$X_r = X_a$;
for $t_i \in R$ **do**
    index = QuerySmallWorldGraph(G, e_i);
    $z = C$[index].token;
    Replace t_i in X_r with z;
return X_r;

beddings are transformed into a hierarchical set of SWGs based on the proximity between different embeddings. To conduct kNN searches, the property of degree distributions in SWGs significantly reduces the search space of each kNN query from $O(n)$ to $O(\log n)$ by navigating on the graphs, so a kNN query can be efficiently completed in $O(k \log n)$ time complexity. Finally, the recovered text X_r can be obtained by replacing the perturbations R in X_a as shown in Algorithm 1.

3.4 Learning and Optimization

To learn a robust discriminator, we randomly sample adversarial examples from both character-level and word-level attacks in each training epoch. The loss function optimizes the cross-entropy between the labels and the probabilistic scores computed by the logits y_i and the softmax function.

The learning process of embedding estimator is similar to masked language models. The major difference is that language models optimize the likelihood to generate the same original token while the embedding estimator minimizes the distance between the derived embedding and the original token embedding. To learn the embedding estimator, a size-$(2w+1)$ sliding window is applied to enumerate $(2w+1)$-gram training data for approximating embeddings with context tokens. For optimization, the embedding estimator is learned to minimize the mean square error (MSE) from the inferred embeddings to the original token embeddings.

To take advantage of hierarchical navigable SWGs for an efficient recovery, although a preprocess to construct SWGs G is required, the preprocess can be fast. The established SWGs can also be serialized in advance. More precisely, the time complexity is $O(kn \log n)$ for one-time construction of reusable SWGs, where n is the number of embeddings in the embedding corpus C.

Dataset	Train	Test	Length		
			Max.	Min.	Avg.
SST-2	67,349	1,821	56	1	19
IMDb	25,000	25,000	2,738	8	262

Table 1: The statistics of datasets.

Attack Type	Example
No Attack	Old-form moviemaking at its best.
Insertion	Old-form moviemaking at its **beast**.
Deletion	Old-form moviemaking at its **be~~s~~t**.
Swap	Old-form moviemaking at its **bets**.
Random	Old-form moviemaking at its **aggrandize**.
Embed	Old-form moviemaking at its **way**.

Table 2: Examples of each type of attack

4 Experiments

In this section, we conduct extensive experiments to evaluate the performance of DISP in improving model robustness.

4.1 Experimental Settings

Experimental Datasets. Experiments are conducted on two benchmark datasets: (1) Stanford Sentiment Treebank Binary (SST-2) (Socher et al., 2013) and (2) Internet Movie Database (IMDb) (Maas et al., 2011). SST-2 and IMDb are both sentiment classification datasets which involve binary labels annotating sentiment of sentences in movie reviews. Detailed statistics of two datasets are listed in Table 1.

Attack Generation. We consider three types of character-level attacks and two types of word-level attacks. The character-level attacks consist of *insertion*, *deletion*, and *swap*. *Insertion* and *deletion* attacks inject and remove a character, respectively, while a *swap* attack flips two adjacent characters. The word-level attacks include *random* and *embed*. A *random* attack randomly samples a word to replace the target word while a *embed* attack replaces the word with a word among the top-10 nearest words in the embedding space. The examples of each attack type are illustrated in Table 2. To obtain strong adversarial attack samples,w we consider to leverage oracle to identify the perturbations that cause prediction changes. Specifically, for each test sample we construct 50 adversarial examples by perturbing the test data. We sample one example in which model prediction changes after perturbing. If none of them can

Dataset	Method	Metric	Character-level Attacks			Word-level Attacks		Overall
			Insertion	Deletion	Swap	Random	Embed	Attacks
SST-2	SC	Precision	0.5087	0.4703	0.5044	0.1612	0.1484	0.3586
		Recall	0.9369	0.8085	0.9151	0.1732	0.1617	0.5991
		F1	0.6594	0.5947	0.6504	0.1669	0.1548	0.4452
	DISP	Precision	0.9725	0.9065	0.9552	0.8407	0.4828	0.8315
		Recall	0.8865	0.8760	0.8680	0.6504	0.5515	0.7665
		F1	**0.9275**	**0.8910**	**0.9095**	**0.7334**	**0.5149**	**0.7952**
IMDb	SC	Precision	0.0429	0.0369	0.0406	0.0084	0.0064	0.0270
		Recall	0.9367	0.8052	0.8895	0.1790	0.1352	0.5891
		F1	0.0820	0.0706	0.0777	0.0161	0.0122	0.0517
	DISP	Precision	0.9150	0.8181	0.8860	0.5233	0.2024	0.6690
		Recall	0.5068	0.4886	0.5000	0.3876	0.2063	0.4179
		F1	**0.6523**	**0.6118**	**0.6392**	**0.4454**	**0.2044**	**0.5106**

Table 3: Performance of SC and DISP on identifying perpetuated tokens.

change the prediction, the sample with the least confidence is selected.

Base Model and Baselines. We consider BERT (Devlin et al., 2018) as the base model as it achieves strong performance in these benchmarks. To evaluate the performance of DISP, we consider the following baseline methods: (1) Adversarial Data Augmentation (ADA) samples adversarial examples to increase the diversity of training data; (2) Adversarial Training (AT) samples different adversarial examples in each training epoch; (3) Spelling Correction (SC) is used as a baseline for discriminating perturbations and blocking character-level attacks. Note that ADA and AT require to re-train BERT with the augmented training data, while DISP and SC modify the input text and then exploit the original model for prediction. SC is also the only baseline for evaluating discriminator performance. In addition, we also try to ensemble DISP and SC (DISP+SC) by conducting DISP on the spelling corrected input.

Evaluation Metrics. We evaluate the performance of the perturbation discriminator by precision, recall and F1 scores, and evaluate the overall end-to-end performance by classification accuracy that the models recover.

Implementation Details. The model is implemented in PyTorch (Paszke et al., 2017). We set the initial learning and dropout parameter to be 2×10^{-5} and 0.1. We use crawl-300d-2M word embeddings from *fastText* (Mikolov et al., 2018) to search similar words. The dimensions of word embedding k and contextual representation d are set as 300 and 768. w is set as 2. We follow $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018) to set the numbers of layers (i.e., Transformer blocks) and self-attention heads as 12.

4.2 Experimental Results

Performance on identifying perpetuated tokens. Table 3 shows the performance of DISP and SC in discriminating perturbations. Compared to SC, DISP has an absolute improvement by 35% and 46% on SST-2 and IMDb in terms of F1-score, respectively. It also proves that the context information is essential when discriminating the perturbations. An interesting observation is that SC has high recall but low precision scores for character-level attacks because it is eager to correct misspellings while most of its corrections are not perturbations. Conversely, DISP has more balances of recall and precision scores since it is optimized to discriminate the perturbed tokens. For the word-level attacks, SC shows similar low performance on both *random* and *embed* attacks while DISP behaves much better. Moreover, DISP works better on the *random* attack because the embeddings of the original tokens tend to have noticeably greater Euclidean distances to randomly-picked tokens than the distances to other tokens.

Defense Performance. Table 4 reports the accuracy scores of all methods with different types of adversarial attacks on two datasets. Compared to the baseline BERT model, all of the methods alleviate the performance drops. All methods perform better on blocking character-level attacks than word-level attacks because word-level attacks eliminate more information. For the base-

Dataset	Method	Attack -free	Character-level Attacks			Word-level Attacks		Overall Attacks
			Insertion	Deletion	Swap	Random	Embed	
SST-2	BERT	**0.9232**	0.6498	0.6544	0.6774	0.5385	0.6556	0.6351
	SC	0.9174	**0.9082**	0.8186	**0.8840**	0.5993	0.7003	0.7821
	ADA	0.9174	0.8071	0.8071	0.8209	0.7394	0.7681	0.7885
	AT	0.9186	0.8186	0.8175	0.8025	0.6935	0.7646	0.7793
	DISP	**0.9232**	0.8278	**0.8278**	0.8301	**0.7773**	**0.7784**	**0.8083**
	DISP+SC	0.9197	**0.9128**	**0.8681**	**0.9060**	**0.7784**	**0.7853**	**0.8501**
IMDb	BERT	**0.9431**	0.8586	0.8599	0.8568	0.8468	0.8615	0.8567
	SC	0.9193	0.8834	0.8794	0.8825	0.8695	0.8753	0.8780
	ADA	0.9393	0.8766	0.8765	0.8754	0.8722	0.8755	0.8752
	AT	0.8998	0.8958	0.8822	0.8787	0.8886	0.8822	0.8855
	DISP	0.9378	**0.9310**	**0.9297**	**0.9301**	**0.9281**	**0.9347**	**0.9307**
	DISP+SC	0.9395	**0.9316**	0.8772	**0.9313**	0.8755	0.9292	0.9090

Table 4: The accuracy scores of methods with different adversarial attacks on two datasets.

lines, consistent with Table 3, SC performs the best for character-level attacks and the worst for word-level attacks. In contrast, ADA and AT are comparably more stable across different types of attacks. The differences between performance for character- and word-level attacks are less obvious in IMDb because documents in IMDb tend to be longer with more contexts to support the models. DISP works well to block all types of attacks. Compared with the best baseline models, DISP significantly improves the classification accuracy by 2.51% and 5.10% for SST-2 and IMDb, respectively. By ensembling SC and DISP, DISP+SC achieves better performance for blocking all types of attacks. However, the improvements are not consistent in IMDb. In particular, SC performs worse with lower discrimination accuracy and over-correcting the documents. In addition, DISP has a stable defense performance across different types of attacks on IMDb because richer context information in the documents benefits token recovery.

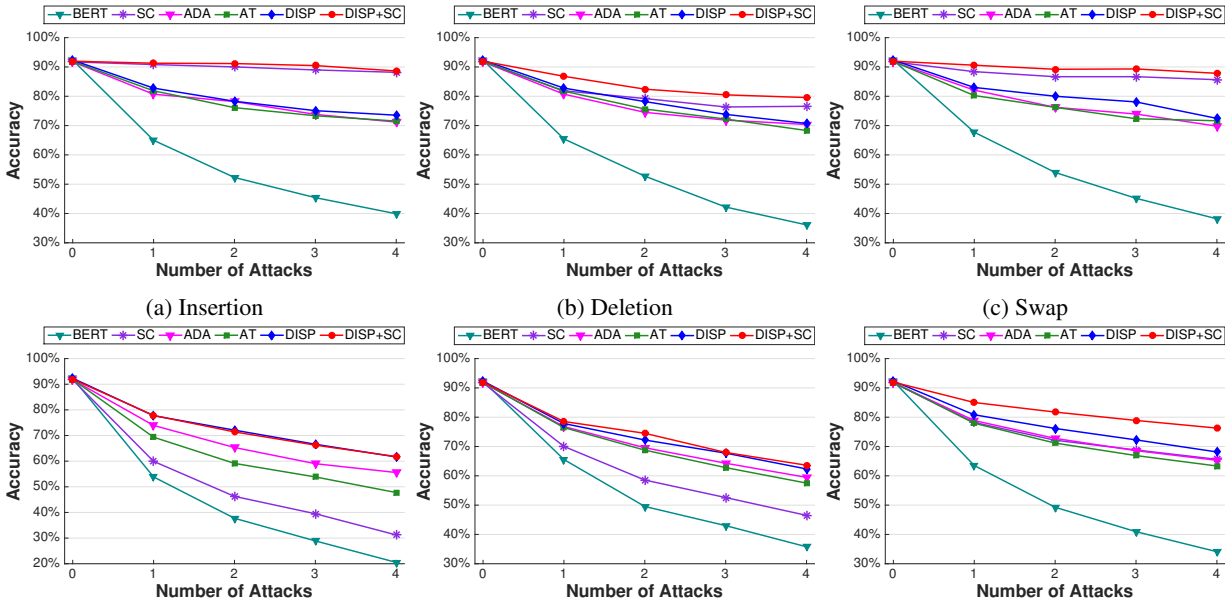
Number of Attacks. Figure 4 shows the classification accuracy of all methods over different numbers of attacks, i.e., perturbations, for different types of adversarial attacks. Without using a defense method, the performance of BERT dramatically decreases when the number of attacks increases. With defense approaches, the performance drops are alleveated. Moreover, the relations between the performance of methods are consistent across different perturbation numbers. DISP+SC consistently performs the best for all of the cases when DISP outperforms all of the single methods for most of the situations. These results demonstrate the robustness of the proposed approach.

Method	Insertion	Delete	Swap
BERT	0.6498	0.6544	0.6774
$\text{DISP}_{\text{SST-2}}$	0.8278	0.8278	0.8301
$\text{DISP}_{\text{IMDb}}$	0.8243	0.8197	0.8278
Method	**Random**	**Embed**	**Overall**
BERT	0.5385	0.6556	0.6351
$\text{DISP}_{\text{SST-2}}$	0.7773	0.7784	0.8083
$\text{DISP}_{\text{IMDb}}$	0.7623	0.7681	0.8005

Table 5: The accuracy of DISP over different types of attacks on the SST-2 dataset with the tokens recovered by the perturbation discriminator and the embedding estimator trained on the IMDb dataset for robust transfer defense. Note that DISP_x indicates the framework is established on the dataset x.

Robust Transfer Defense. In practice, we may not have access to the original training corpus of a prediction model. In the following, we investigate if the perturbation discriminator can transfer across different corpora. We first train the discriminator and the estimator on IMDb denoted as $\text{DISP}_{\text{IMDb}}$ and then apply it to defend the prediction model on SST-2. Table 5 shows the experimental results of robust transfer defense. $\text{DISP}_{\text{IMDb}}$ achieves similar performance as the performance of $\text{DISP}_{\text{SST-2}}$ trained on the same training set. Hence, it shows that DISP can transfer the ability to recover perpetuated token across different sentiment copora.

Case Study of Recovered Text. Table 6 lists four



(a) Insertion (b) Deletion (c) Swap

(d) Random (e) Embed (f) Overall

Figure 4: The accuracy of methods over different numbers and types of attacks.

#	Attacked Sentence	Recovered Token	Label	Pred
1	Mr. Tsai is a very **orig~~i~~nal** artist in his medium, and what time is it there?	imaginative	positive	positive
2	Old-form moviemaking at its **be~~s~~t**.	best	positive	positive
3	My reaction in a word: **disappo**n**itment**.	that	negative	positive
4	a **painfuli**l**y fun**t**ny** ode to g**bad** behavior.	painfully; silly; one	positive	negative

Table 6: A case study of recovered tokens in SST-2. Note that Label and Pred represent the ground-truth label and the predicted label.

documents from SST-2 for a case study. We successfully recovered the attacked words from "original" and "bet" in the cases 1 and 2 to "imaginative" and "best". It demonstrates that embeddings generated by the embedding estimator are robust to recover the appropriate tokens and block adversarial attacks. However, DISP performs worse when the remaining sentence is lack of informative contexts as case 3. When multiple attacks exist, the incorrect context may also lead to unsatisfactory recoveries, e.g., DISP converts "funny" to "silly" in case 4, thus flipping the prediction. This experiment depicts a disadvantage of DISP and demonstrates that DISP+SC can gain further improvements.

Embedding Estimator. Although DISP is not required to recover the ground-truth perturbed tokens, the embedding estimator plays an important role to derive appropriate embedding vectors that obtain the original semantics. We first evaluate the performance of embedding estimator as a regression task. The RMSE scores of estimated embeddings are 0.0442 and 0.1030 in SST-2 and IMDb datasets, which are small enough to derive satisfactory tokens. To further demonstrate the robustness of the embedding estimator and estimated embeddings, we identify the perturbations with our discriminator and replace them with the ground-truth tokens. Table 7 shows the accuracy scores over different types of attacks in the SST-2 dataset. DISP and DISP$_G$ denotes the recovery performance with our estimator and gound-truth tokens, respectively. More specifically, the accuracy of DISP$_G$ presents the upperbound performance gained by the embedding estimator. The experimental results demonstrate the robustness of

Method	Insertion	Delete	Swap
DISP$_G$	0.8773	0.8681	0.8796
DISP	0.8278	0.8278	0.8301
Method	Random	Embed	Overall
DISP$_G$	0.7970	0.7924	0.8429
DISP	0.7773	0.7784	0.8083

Table 7: The performance of DISP using ground-truth and recovered tokens over different types of attacks in SST-2. Result are in accuracy. Note that DISP$_G$ denotes DISP using ground-truth tokens.

Method	Insertion	Delete	Swap
BERT	0.1160	0.1407	0.1806
DISP	0.5856	0.5684	0.6008
Method	Random	Embed	Overall
BERT	0.0855	0.0817	0.1209
DISP	0.4848	0.5114	0.5502

Table 8: The accuracy scores of BERT and DISP over different types of attacks on the CoLA dataset for the task of linguistic acceptability classification. The accuracy score of BERT without any attack is 0.8519.

the embedding estimator while the estimated embeddings only slightly lower the accuracy of DISP.

Linguistic Acceptability Classification. In addition to the task of sentiment analysis, we also evaluate the performance of DISP in linguistic acceptability classification. The Corpus of Linguistic Acceptability (CoLA) is a binary classification task. The goal of this task is to predict whether an English sentence is linguistically acceptable or not (Warstadt et al., 2018). Table 8 presents the accuracy scores of BERT and DISP on the CoLA dataset with one adversarial attack of each type. It is interesting that the original BERT is extremely vulnerable to the adversarial attacks. This is because the linguistic acceptability can be easily affected by perturbations. The experimental results also depict that DISP can significantly alleviate the performance drops. DISP is capable of blocking adversarial attacks across different NLP tasks.

5 Conclusions

In this paper, we propose a novel approach to discriminate perturbations and recover the text semantics, thereby blocking adversarial attacks in NLP. DISP not only correctly identifies the perturbations but also significantly alleviates the performance drops caused by attacks.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments. The work was supported by NSF DGE-1829071 and NSF IIS-1760523.

References

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

Minhao Cheng, Wei Wei, and Cho-Jui Hsieh. 2019. Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent. 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 3325–3335.

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

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (2015)*.

Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.

Shixiang Gu and Luca Rigazio. 2015. Towards deep neural network architectures robust to adversarial examples. In *ICLR*.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using google web it 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1241–1249. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Yury A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*.

Gabriel Marzinotto, Géraldine Damnati, Frédéric Béchet, and Benoit Favre. 2019. Robust semantic parsing with adversarial learning for domain generalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 166–173.

Eric Mays, Fred J Damerau, and Robert L Mercer. 1991. Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Dongyu Meng and Hao Chen. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. 2018. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580.

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 Conference of the Association for Computational Linguistics*, pages 1085–1097.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 366–372.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *ICLR*.

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

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.

Wenqi Wang, Benxiao Tang, Run Wang, Lina Wang, and Aoshuang Ye. 2019. A survey on adversarial attacks and defenses in text. *arXiv preprint arXiv:1902.07285*.

David Warde-Farley and Yoshua Bengio. 2017. Improving generative adversarial networks with denoising feature matching. In *ICLR*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*.

Qingfu Zhu, Lei Cui, Wei-Nan Zhang, Furu Wei, and Ting Liu. 2019. Retrieval-enhanced adversarial training for neural response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3763–3773, Florence, Italy. Association for Computational Linguistics.