Distinguishing Non-natural from Natural Adversarial Samples for More Robust Pre-trained Language Model

Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, Hai Zhao*
Department of Computer Science and Engineering, Shanghai Jiao Tong University
Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
wangjiayi_102_23@sjtu.edu.cn, rongzhou.bao@outlook.com
zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Recently, the problem of robustness of pre-trained language models (PrLMs) has received increasing research interest. Latest studies on adversarial attacks achieve high attack success rates against PrLMs, claiming that PrLMs are not robust. However, we find that the adversarial samples that PrLMs fail are mostly non-natural and do not appear in reality. We question the validity of current evaluation of robustness of PrLMs based on these non-natural adversarial samples and propose an anomaly detector to evaluate the robustness of PrLMs with more natural adversarial samples. We also investigate two applications of the anomaly detector: (1) In data augmentation, we employ the anomaly detector to force generating augmented data that are distinguished as non-natural, which brings larger gains to the accuracy of PrLMs. (2) We apply the anomaly detector to a defense framework to enhance the robustness of PrLMs. It can be used to defend all types of attacks and achieves higher accuracy on both adversarial samples and compliant samples than other defense frameworks. The code is available at https://github.com/LilyNLP/Distinguishing-Non-Natural.

1 Introduction

Pre-trained language models (PrLMs) have achieved state-of-the-art performance across a wide variety of natural language understanding tasks (Devlin et al., 2018; Liu et al., 2019a; Clark et al., 2020). Most works of PrLMs mainly focus on designing stronger model structures and training objectives to improve the accuracy or training efficiency. However, in real industrial applications, there exist noises that can mislead the predictions of PrLMs (Malykh, 2019), which raise potential security risks and limit the application efficacy of PrLMs in practice. To solve this challenge, studies around the robustness of PrLMs have received increasing research interest. Recent studies demonstrated that, due to the lack of supervising signals and data noises in the pre-training stage, PrLMs are vulnerable to adversarial attacks, which can generate adversarial samples to fool the model (Zhang et al., 2020). A variety of attack algorithms have been proposed to use spelling errors (Li et al., 2019), synonym substitutions (Jin et al., 2020), phrase insertions (Le et al., 2020) or sentence structure reconstructions (Zhao et al., 2018) to generate adversarial samples. Some of these attack algorithms have achieved an over 90% attack success rate on PrLMs (Li et al., 2020; Garg and Ramakrishnan, 2020). Thus they claim that existing PrLMs are not robust.

However, we investigate the adversarial samples on which PrLMs fail, and find that most of them are not natural and fluent, thus can be distinguished by humans. These samples are unlikely to appear in reality and are against the principle that adversarial samples should be imperceptible to humans (Zhang et al., 2020). Therefore it is not reasonable to judge the robustness of PrLMs based on these non-natural adversarial samples. By adopting a PrLM-based anomaly detector and a two-stage training strategy, we empirically demonstrate that most of the non-natural adversarial samples can be detected by the machine. Furthermore, we adopt the anomaly score (the output probability of the anomaly detector) as a constraint metric to help adversarial attacks generate more natural samples. Under this new constraint of generating natural samples, the attack success rates of existing attack methods sharply decrease. These experimental results demonstrate that the robustness of PrLMs is not as fragile as previous works claimed.

Then we explore two application scenarios of the anomaly detector. Firstly, we wonder whether the anomaly detection can generalize to other appli-

*Corresponding author. This work was supported in part by the Key Projects of National Natural Science Foundation of China under Grants U1836222 and 61733011.

Findings of the Association for Computational Linguistics: ACL 2022, pages 905 - 915

cations using artificially modified sentences. Thus we think of the data augmentation scenario. The objective of data augmentation is to increase the diversity of training data without explicitly collecting new data (Wei and Zou, 2019). For an original sequence and a data augmentation technique, there exist many possible augmented sequences. We apply the anomaly detector to select among these possibilities the augmented sequence that can bring more diversity into training data. For each original sequence, we continuously generate augmented sequences until the anomaly detector distinguishes one as anomaly. The augmented data under this constraint can further increase the prediction accuracy of PrLMs than ordinary data augmentation.

Secondly, we integrate the anomaly detector into a defense framework to enhance the robustness of PrLMs. Inspired by the defense methods in the computer vision domain (Liu et al., 2019b; Das et al., 2017; Raff et al., 2019) which apply transformations like JPEG-based compression to mitigate the adversarial effect, we use textual transformations to restore adversarial samples. We consider a candidate set of transformation functions including back translation, MLM suggestion, synonym swap, adverb insertion, tense change, and contraction. For the input sequence that is detected as an adversarial sample, we randomly apply k transformation functions from the candidate set to the sequence. We send the k transformed sequences to the PrLM classifier to get their prediction scores. The final prediction is based on the average of these k prediction scores. Empirical results demonstrate that this defense framework achieves higher accuracy than other defense frameworks on both adversarial samples and compliant samples (By compliant samples, we mean the non-adversarial samples in original datasets).

2 Related Work

The study of the robustness of PrLMs is based on the competition between adversarial attacks and defenses. Adversarial attacks find the adversarial samples where PrLMs are not robust, while defenses enhance the robustness of PrLMs by utilizing these adversarial samples or modifying model structure against the attack algorithm.

2.1 Adversarial Attacks

Problem Formulation Adversarial attacks generate adversarial samples against a victim model F, which is a PrLM-based text classifier in this paper. Given an input sequence X, the victim model F predicts its label $F(X) = y$. The corresponding adversarial sample X_{adv} should alter the prediction of the victim model and meanwhile be similar to original sequence:

$$\begin{aligned} & F(X_{adv}) \neq F(X) \\ & s.t. \quad d(X_{adv}, X) < \sigma, \end{aligned} \tag{1}$$

where $d()$ measures the size of perturbations, and σ is a predefined threshold.

Classification of attacks Adversarial attacks can be conducted in both white-box and black-box scenarios. In the white-box scenario (Meng and Wattenhofer, 2020), adversarial attacks can access all information of their victim models. In the black-box scenario, adversarial attacks can only get the output of the victim models: if they get prediction scores, they are score-based attacks (Jin et al., 2020); if they get the prediction label, they are decision-based attacks (Wallace et al., 2020).

According to the granularity of perturbations, textual attacks can be classified into character-level, word-level, and sentence-level attacks. Character-level attacks (Gao et al., 2018) introduce noises by replacing, inserting, or deleting a character in several words. Word-level attacks substitute several words by their synonyms to fool the model (Jin et al., 2020; Garg and Ramakrishnan, 2020). Sentence-level attacks generate adversarial samples by paraphrasing the original sentence (Iyyer et al., 2018) or using a generative adversarial network (GAN) (Zhao et al., 2018).

Metrics to constrain perturbations To evaluate the robustness of PrLMs, it is important that the adversarial samples are within a perturbation constraint. An adversarial sample must have similar semantic meaning to the original sample, while syntactically correct and fluent as a natural language sequence. Existing attack methods adopt the following metrics to realize this requirement:

(1) **Semantic Similarity**: semantic similarity is the most popular metric used in existing attack works (Jin et al., 2020; Li et al., 2020). They use Universal Sentence Encoder (USE) (Cer et al., 2018) to encode original sentence and adversarial sentence into vectors and use their cosine similarity to define semantic similarity.

(2) **Perturbation Rate**: perturbation rate is often used in word-level attacks (Jin et al., 2020) (Li

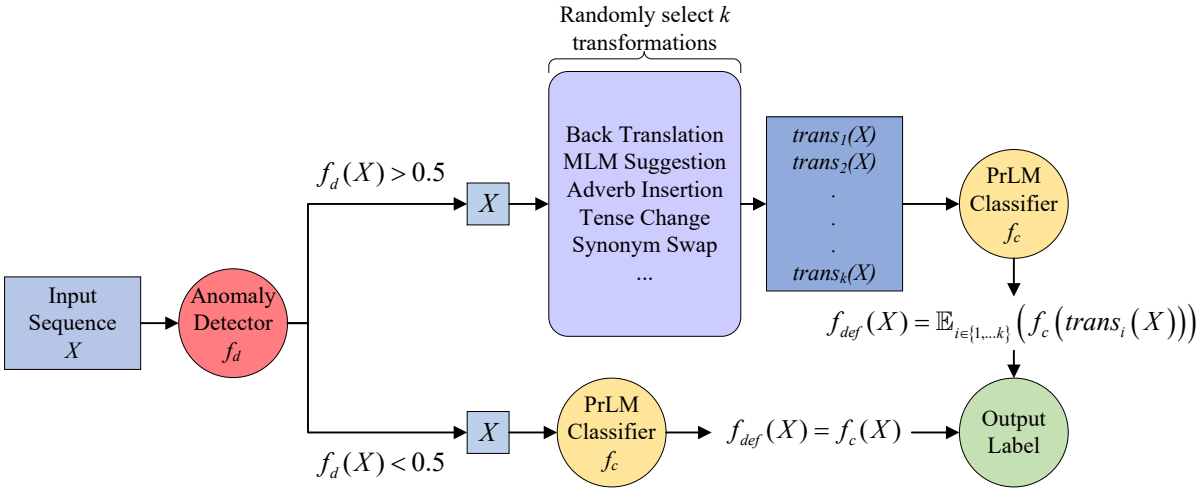

Figure 1: Defense framework.

et al., 2020) to indicate the rate between the number of modified words and total words.

(3) **Number of Increased Grammar Errors**: it is the number of increased grammatical errors in the adversarial sample compared to the original sample. This metric is used in (Maheshwary et al., 2020), (Li et al., 2021) and is calculated using LanguageTool (Naber, 2003).

(4) **Levenshtein Distance**: levenshtein distance is often used in character-level attacks (Gao et al., 2018). It refers to the number of editing operations to convert one string to another.

2.2 Adversarial Defenses

The objective of adversarial defenses is to design a model which can achieve high accuracy on both compliant and adversarial samples. One direction of adversarial defenses is adversarial training. By augmenting original training data with adversarial samples, the model is trained to be more robust to the perturbations seen in the training stage (Goodfellow et al., 2015). However, it is impossible to explore all potential perturbations within a limited number of adversarial samples. Empirical results demonstrate that the improvement of robustness brought by adversarial training alone is quite limited when faced with strong dynamic attacks (Jin et al., 2020; Maheshwary et al., 2020).

Another direction is modifying the model structure against a specific type of adversarial attack. For character-level attacks, ScRNN (Pruthi et al., 2019) leverages an RNN semi-character architecture to identify and restore the modified characters. For word-level attacks, DISP (Zhou et al., 2019) utilizes a perturbation discriminator followed by an embedding estimator to restore adversarial samples. For sentence-level attacks, DARCY (Le et al., 2021) greedily searches and injects multiple trapdoors into the model to catch potential UniTrigger attacks (Wallace et al., 2019).

Certified robustness is a particular branch of defense whose aim is to ensure that the model predictions are unchanged within a perturbation scope. For example, (Jia et al., 2019) and (Huang et al., 2019) certify the robustness of the model when input word embeddings are perturbed within the convex hull formed by the embeddings of its synonyms. However, certified robustness is hard to scale to deep networks and harms the model's accuracy on compliant samples due to the looser outer bound.

3 Methods

3.1 Anomaly Detector

We adopt a PrLM-based binary classifier as the anomaly detector to distinguish adversarial samples from compliant samples. For an input sequence X, X is firstly separated into sub-word tokens with a special token `[CLS]` at the beginning. A PrLM then encodes the tokens and generates a sequence of contextual embeddings $\{h_0, h_1, h_2, ..., h_n\}$, in which $h_0 \in \mathbb{R}^H$ is the contextual representation of `[CLS]`. For text classification tasks, h_0 is used as the aggregate sequence representation which contains the sentence-level information. So the anomaly detector leverages h_0 to predict the probability that X is labeled as class $\hat{y}_d$ (if X is adversarial sample, $\hat{y}_d = 1$; if X is compliant sample,

	Example
Original	You wonder why enough wasn't just a music video rather than a full-length movie.
Back Translation	You wonder why enough **was not** just a music video, **but** a full-length movie.
MLM Suggestion	You wonder why enough wasn't just a music video rather than a full-length **film**.
Adverb Insertion	You **profoundly** wonder why enough **aside** wasn't just a music video rather than a full-length movie.
Tense Change	You **wondered** why enough **isn't** just a music video rather than a full-length movie.
Synonym Swap	You **doubt** why **full** wasn't **only** a music video rather than a full-length movie.
Contraction	You wonder why enough **was not** just a music video rather than a full-length movie.

Figure 2: Examples of transformation functions used in the defense framework.

	Label	Example	Anomaly Score
Original	**Positive**	Seldom has a movie so closely matched the spirit of a man and his work.	0.3%
DeepWordBug	**Negative**	Seldom has a moviMe so closely matched the sptrit of a man and his work.	99.2%
TextFooler	**Negative**	Seldom has a **movies** so closely **confronted** the **esprit** of a **fella** and his **cooperate**.	98.7%
BERT-Attack	**Negative**	Seldom **possesses** a movie**,** closely matched the spirit of a man and his work **because**	98.3%
SCPN	**Negative**	**There's rarely a film of a man and his job.**	0.4%

Figure 3: Examples of adversarial samples generated by four adversarial attacks.

$\hat{y_d} = 0$) by a logistic regression with softmax:

$$y_d = \text{softmax}(W_d(dropout(h_0)) + b_d). \quad (2)$$

And we use the binary cross entropy loss function to train the anomaly detector :

$$loss_d = -y_d * \log\hat{y_d} - (1-y_d) * \log(1-\hat{y_d}). \quad (3)$$

We adopt a two-stage training strategy for the anomaly detector. In the first stage, we generate the "artificial samples" using the same way each attack modifies the sentence (details of how attacks modify the sentences are described in Section 4.2). But the artificial samples are not required to alter the prediction result of the PrLM, so the modification is only applied once. For example, to generate artificial samples simulating the word-level attack TextFooler, we substitute a portion of words by their synonyms in a synonym set according to WordNet. The training data consist of original samples (labeled as 0) in the train set and their corresponding artificial samples (labeled as 1). We train the detector on these data so that it can learn to distinguish artificially modified sequences from natural sequences. In the second stage, we generate the adversarial samples (labeled as 1) from the original samples (labeled as 0) in the train set, and train the anomaly detector to distinguish adversarial samples from original samples. In this way, the detector can distinguish non-naturally modified examples, and especially the adversarial ones among them. The experimental results in section 5.1 demonstrate that the anomaly detector can accurately distinguish adversarial samples from compliant samples.

Task	Dataset	Train	Test	Avg Len
Classification	MR	9K	1K	20
	SST2	67K	1.8K	20
	IMDB	25K	25K	215
Entailment	MNLI	433K	10K	11

Table 1: Dataset statistics.

3.2 Evaluation of Robustness under Anomaly Score Constraint

Existing adversarial samples have applied some thresholds to limit the anomaly of adversarial samples. However, the generated adversarial samples are still not natural, indicating that existing metrics are not effective enough. In order to measure the robustness of PrLMs with more natural adversarial samples, we use a new metric: anomaly score, to constrain the perturbations. Given a sentence X, we leverage the probability that X is adversarial sample predicted by anomaly detector as the anomaly score of X:

$$Score(X) = Prob(\hat{y_d} = 1|X). \quad (4)$$

For existing attacks, we add a threshold on anomaly score to enforce the attacks to generate more natural and undetectable adversarial samples.

	MR			SST2			IMDB			MNLI		
	TPR.	FPR.	F1.	TPR.	FPR.	F1.	TPR.	FPR.	F1.	TPR.	FPR.	F1.
DeepWordBug	96.2	1.3	97.4	98.5	3.7	97.4	94.4	1.6	96.3	97.6	9.2	94.4
TextFooler	80.2	3.8	87.2	90.6	18.9	86.5	83.6	2.6	89.8	87.6	11.0	88.2
BERT-Attack	72.6	4.0	81.9	86.5	12.8	87.1	87.2	3.2	91.6	86.4	13.0	86.7
SCPN	94.5	4.1	95.2	94.6	12.6	88.2	-	-	-	93.0	13.4	90.0

Table 2: Performance of anomaly detector trained on each dataset and each attack method.

	MR		SST2		IMDB		MNLI	
	w/o Cons.	w Cons.	w/o Cons.	w Cons.	w/o Cons.	w Cons.	w/o Cons.	w Cons.
Deepwordbug	82.2	8.5	78.3	2.8	74.2	23.2	76.8	25.2
TextFooler	80.5	35.2	61.0	31.4	86.6	40.4	86.5	38.3
BERT-Attack	84.7	13.9	87.2	11.5	87.5	18.9	89.8	15.2

Table 3: The attack success rate of attacks using BERT as victim model without and with the anomaly score constraint on MR, SST2, IMDB, MNLI.

The attack problem formulation now becomes:

$$\begin{aligned} & F(X_{adv}) \neq F(X) \\ s.t. \quad & d(X_{adv}, X) < \sigma, \\ & Score(X_{adv}) < 0.5, \end{aligned} \tag{5}$$

where $d()$ measures the perceptual difference between X_{adv} and X. Each attack has its own definition of $d()$ and threshold σ. And we add on a new constraint that the anomaly score of X_{adv} should be smaller than 0.5. We investigate the robustness of PrLMs under the constraint of anomaly score and find that PrLMs are more robust than previously claimed.

3.3 Application in Data Augmentation

In data augmentation, PrLM is trained on original sentences and their artificially augmented sentences to improve the diversity of training data. We consider random synonym substitution as the augmentation technique for experiments. For an original sequence of n words, we randomly select $p\% * n$ words and substitute them with their synonyms to form the augmented sequence. For each replaced word, the replacing synonym is randomly selected among its s most similar synonyms. So we will have in total $C_n^{p\%*n} * s^{p\%*n}$ possible augmented sequences. In order to select the augmented sequence that can bring more diversity into training data, we apply the anomaly detector to select the augmented sequence that is distinguished as anomaly. For each original sequence, we continuously apply random synonym substitution to form candidate augmented sequences until the detector distinguishes one as anomaly.

3.4 Application in Enhancing Robustness

There are two ways to apply the anomaly detector in enhancing the robustness of PrLMs: (1) detect and then directly block the adversarial samples; (2) distinguish the adversarial samples and conduct operations on them to make the PrLMs give correct predictions. The first application is trivial so we explore the second way.

We propose a defense framework as shown in Figure 1. We firstly build a transformation function set containing t transformation function candidates: *Back Translation* (translate the original sentence into another language and translate it back to original language); *MLM Suggestion* (mask several tokens in the original sentence and use masked language model to predict the masked tokens); *Adverb Insertion* (insert adverbs before verbs); *Tense Change* (change the tense of verbs into another tense); *Synonym Swap* (swap several words with their synonyms according to WordNet), *Contraction* (contract or extend the original sentence by common abbreviations). We implement these transformation functions based on (Wang et al., 2021) [1]. The examples of these transformation functions are displayed in Figure 2.

For each input sequence X, we apply the anomaly detector f_d to identify whether it is adversarial ($f_d(X) > 0.5$) or not ($f_d(X) < 0.5$). If the X is recognized as compliant sample, it will be directly sent to the PrLM classifier f_c to get the final output probability of the defense framework: $f_{def}(X) = f_c(X)$. If X is recognized as adversarial sample, we will randomly select k transformation functions from the transformation candidate

[1] https://github.com/textflint/textflint

	BERT		RoBERTa		ELECTRA	
	w/o Cons.	w Cons.	w/o Cons.	w Cons.	w/o Cons.	w Cons.
Deepwordbug	82.2	8.5	83.8	10.4	79.4	7.9
TextFooler	80.5	35.2	67.6	36.3	63.6	33.6
BERT-Attack	84.7	13.9	73.7	17.4	70.8	14.2

Table 4: The attack success rate of attacks without and with the anomaly score constraint using different PrLMs as victim models on MR.

	No Augmentation	Augmentation w/o Selection	Augmentation w Selection
BERT	86.4	87.1	88.3
RoBERTa	88.3	89.1	89.5
ELECTRA	90.1	90.2	90.4

Table 5: The accuracy of no augmentation, after the data augmentation without and with the selection of detector on MR.

set and apply them to X. We send the k transformed sequences $trans_i(X), i \in \{1, ..., k\}$ to the PrLM classifier to get their prediction probabilities $f_c(trans_i(X)), i \in \{1, ..., k\}$, and the final prediction probability of the defense framework is the expectation over the k transformed probabilities $f_{def}(X) = \mathbb{E}_{i \in \{1,...,k\}}(f_c(trans_i(X)))$.

Since the detector is not perfect, there always exist a small number of compliant samples that are misclassified into adversarial samples. In order to minimize the harm to the accuracy of PrLMs on compliant samples, during the training stage of PrLMs, we augment the training data with their transformed data. In this way, the PrLMs are more stable to transformations on compliant samples, and data augmentation itself also brings gains to the accuracy of PrLMs.

4 Experimental Implementation

4.1 PrLMs

We investigate three PrLMs: $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018), $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019a) and $\text{ELECTRA}_{\text{BASE}}$ (Clark et al., 2020). The PrLMs are all implemented in their base-uncased version based on PyTorch [2]: they each have 12 layers, 768 hidden units, 12 heads and around 100M parameters. For most experiments on attacks and defenses, we use $\text{BERT}_{\text{BASE}}$ as the victim model for an easy comparison between our results and those of previous works.

[2]https://github.com/huggingface

4.2 Adversarial Attacks

We investigate four adversarial attacks from character level, word level to sentence level. Examples of adversarial samples generated by these four attacks are demonstrated in Figure 3.

Character-level attack For character-level attack, we consider Deepwordbug, which applies four types of character-level modifications (substitution, insertion, deletion and swap) to words in the original sample. Edit distance is used to constrain the similarity between original and adversarial sentences.

Word-level attack We select two classic word-level attack methods: TextFooler (Jin et al., 2020) and BERT-Attack (Li et al., 2020). They both sort the words in the original sample by importance scores, and then substitute the words in order with their synonyms until the PrLM is fooled. TextFooler selects the substitution word from a synonym set of the original word according to WordNet (Mrkšić et al., 2016). BERT-Attack masks the original word and uses the masked language model (MLM) to predict the substitution word. Semantic similarity and perturbation rate are used to constrain the perturbation size.

Sentence-level attack We select SCPN [3] (Iyyer et al., 2018) to generate sentence-level adversarial samples. SCPN applies syntactic transformations to original sentences and automatically labels the sentences with their syntactic transformations. Based on these labeled data, SCPN trains a neural encoder-decoder model to generate syntactically controlled paraphrased adversarial samples. Semantic similarity is used to ensure that the semantic meaning remains unchanged.

4.3 Datasets

Experiments are conducted on four datasets: **SST2** (Socher et al., 2013), **MR** (Pang and Lee, 2005), **IMDB** (Maas et al., 2011), **MNLI** (Nangia et al.,

[3]https://github.com/thunlp/OpenAttack

	MR		SST2		IMDB		MNLI	
	w/o Def.	w Def.	w/o Def.	w Def.	w/o Def.	w Def.	w/o Def.	w Def.
DeepWordBug	16.3	57.5	19.7	62.3	24.3	81.4	18.7	70.3
TextFooler	16.7	66.8	36.2	73.3	12.4	90.3	11.3	69.2
BERT-Attack	13.3	61.5	12.8	65.2	11.8	85.9	9.5	65.4
SCPN	64.2	74.3	70.8	81.5	-	-	66.9	75.0

Table 6: The adversarial accuracy with and without defense using BERT as victim model.

	MR	SST2	IMDB	MNLI
w/o Def.	86.4	92.6	92.4	84.0
w Def.	87.0	92.6	92.5	84.0

Table 7: The original accuracy with and without defense using BERT as victim model.

2017), covering two major NLP tasks: text classification and natural language inference (NLI). The dataset statistics are displayed in Table 1.

For text classification task, we use three datasets with average text lengths from 20 to 215 words in English: (1) **SST2** (Socher et al., 2013): a phrase-level binary sentiment classification dataset on movie reviews; (2) **MR** (Pang and Lee, 2005): a sentence-level binary sentiment classification dataset on movie reviews; (3) **IMDB** (Maas et al., 2011) : a document-level binary sentiment classification dataset on movie reviews. For the NLI task, we use **MNLI** (Nangia et al., 2017), a widely adopted NLI benchmark with coverage of the transcribed speech, popular fiction, and government reports. When attacking the NLI task, we keep the original premises unchanged and generate adversarial hypotheses.

4.4 Experimental Setup

The hyperparameter k in the defense framework is 3. For the victim PrLMs under attack, we fine-tune PrLMs on the training set of each dataset. For the anomaly detector, we use $\text{BERT}_{\text{BASE}}$ as the base PrLM and fine-tune it on the training data indicated in Section 3.1. For the data augmentation, we fine-tune PrLMs on the augmented training set of each dataset. During the fine-tuning of all these PrLMs, we use AdamW (Loshchilov and Hutter, 2018) as our optimizer with a learning rate of 3e-5 and a batch size of 16. The number of training epochs is set to 5. To avoid randomness, we report the results of applications in data augmentation and defense framework based on the average of 3 runs.

5 Experimental Results

5.1 Anomaly Detector

We consider three metrics to evaluate the performance of the anomaly detector: *F1 score* (F1); *True Positive Rate* (TPR): the percentage of adversarial samples that are correctly identified; *False Positive Rate* (FPR): the percentage of compliant samples that are misidentified as adversarial. The experimental results are shown in Table 2. The results of SCPN on the IMDB dataset are unavailable since SCPN cannot tackle document-level texts. Empirical results demonstrate that the anomaly detector can achieve an average F1 score over 90%, an average TPR over 88%, and an average FPR less than 10% for adversarial attacks from character-level, word-level to sentence-level.

5.2 Evaluation of Robustness under Anomaly Score Constraint

We now conduct different types of attacks under the constraint that the anomaly score of generated adversarial samples should be less than 0.5. Table 3 compares the attack success rate of different attacks with and without the anomaly score constraint when the victim PrLM is BERT. We can observe a sharp decrease in attack success rate with the new constraint for all levels of attacks. This result is surprising in that the attackers examined are dynamic. Despite their iterative attempts to attack the model, the attackers fail to generate a natural adversarial sample that can bypass the anomaly detector.

To ensure that this phenomenon holds for other PrLMs, we conduct experiments on RoBERTa and ELECTRA. As shown in Table 4, the attack success rates also drop markedly under the constraint of anomaly score for these PrLMs. These empirical results demonstrate that PrLMs are more robust than previous attack methods have claimed, given that most of the adversarial samples generated by previous attacks are non-natural and detectable. However, there still exist a little portion

	MR		SST2		IMDB		MNLI	
	Orig%	Adv%	Orig%	Adv%	Orig%	Adv%	Orig%	Adv%
No Defense	86.4	16.7	92.6	36.2	92.4	12.4	84.0	11.3
Adv Training	85.4	35.2	92.1	48.5	92.2	34.3	82.3	33.5
DISP	82.0	42.1	91.1	69.8	91.7	81.9	76.3	35.2
SAFER	79.0	55.3	90.8	**75.1**	91.3	88.1	82.1	54.7
Ours	**87.0**	**66.8**	**92.6**	73.3	**92.5**	**90.3**	**84.0**	**69.2**

Table 8: The performance of our defense framework compared with other word-level defenses using BERT as PrLM and TextFooler as attack. Orig% is the original accuracy and Adv% is the adversarial accuracy.

of undetectable adversarial samples that can successfully mislead PrLMs.

5.3 Application in Data Augmentation

We consider the random synonym substitution that substitutes 30% words with their synonyms selected in 50 most similar words. Table 5 compares the accuracy after data augmentation without and with the selection of anomaly detector. We can observe a further increase in accuracy with the selection of the anomaly detector. However, the stronger the PrLM is, the smaller the increase is.

5.4 Application in Enhancing Robustness of PrLMs

We evaluate the performance of the defense framework based on original accuracy and adversarial accuracy. The original accuracy is the prediction accuracy of the defense framework on original compliant samples. The adversarial accuracy is the accuracy of the defense framework after the attack. Here we consider the situation that the attack algorithm can iteratively generate adversarial samples against our defense framework until it succeeds or exceeds the upper limit of attempts.

Table 6 shows the adversarial accuracy with and without the defense using BERT as the victim PrLM. We can see a large improvement in the adversarial accuracy with the defense for all levels of attacks. Table 7 shows the original accuracy with and without the defense. We find that the original accuracy gets even higher with the defense. This is because with anomaly detection, the transformations are only applied to detected adversarial examples. For the very few compliant sentences that are detected by mistake as anomaly and then applied transformations, the data augmentation in the training stage has trained the PrLMs to be stable to transformations on compliant samples. Besides, the data augmentation alone brings an increase to the original accuracy. Therefore the proposed framework does not harm and even increases the prediction accuracy for non-adversarial samples, which is important in real application scenarios.

Since word-level attacks are the most influential and widely-used type of attack, we compare the performance of our defense framework with several state-of-the-art word-level defenses (adversarial training, DISP, SAFER) while facing TextFooler as the attack model. DISP (Zhou et al., 2019) detects and restores adversarial examples by leveraging a perturbation discriminator and an embedding estimator. SAFER (Ye et al., 2020) smooths the classifier by averaging the outputs of a set of randomized examples. As shown in Table 8, although DISP and SAFER are especially designed for word-level attacks, our defense framework outperforms them in most cases on both original accuracy and adversarial accuracy.

6 Discussion

There are two trade-offs for the defense framework:

(1) The trade-off between original accuracy and adversarial accuracy. If we abandon the anomaly detector and apply random transformations to all input sequences, then the adversarial accuracy can further increase by 5-7%, but the original accuracy will decrease by 1-3%. Since in real applications it is not reasonable to sacrifice too much precision for possible security problems, we adopt the anomaly detector to preserve the original accuracy. However, by developing a stronger detector with a higher TPR, the defense framework has the potential to achieve higher adversarial accuracy.

(2) The trade-off between training efficiency and original accuracy. To preserve the original accuracy, we apply data augmentation in the training stage of the defense framework to make it more stable to transformations on compliant samples. However, the training cost is now multiplied by the size of the transformation set n ($n = 6$ in the experimental realization). If we abandon the data augmentation in training stage, the training effi-

ciency of the defense framework is the same as the vanilla fine-tuning of PrLM, but the original accuracy will decrease by 0.5-1.5%.

A limitation of our work is that the attacks we examined are black-box or grey-box attacks, but do not include white-box (gradient-based) attacks. However, since more than 75% of the existing textual attacks are not based on gradient [4], the defense framework is effective for the majority of attacks. We will investigate white-box attacks in future works.

7 Conclusion

In this study, we question the validity of the current evaluation of robustness of PrLMs based on non-natural adversarial samples, and propose an anomaly detector to evaluate the robustness of PrLMs with more natural adversarial samples. To increase the precision of PrLMs, we employ the anomaly detector to select the augmented data that are distinguished as anomaly to introduce more diversity in the training stage. The data augmentation after selection brings larger gains to the accuracy of PrLMs. To enhance the robustness of PrLMs, we integrate the anomaly detector to a defense framework using expectation over randomly selected transformations. This defense framework can be used to defend all levels of attacks, while achieving higher accuracy on both adversarial samples and compliant samples than other defense frameworks targeting specific levels of attack.

References

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression.

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*.

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.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Thai Le, Noseong Park, and Dongwon Lee. 2021. A sweet rabbit hole by DARCY: Using honeypots to detect universal trigger's adversarial attacks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3831–3844, Online. Association for Computational Linguistics.

Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: Generating malicious comments to attack neural fake news detection models.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack.

[4] https://github.com/textflint/textflint

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. *Proceedings 2019 Network and Distributed System Security Symposium*.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. 2019b. Feature distillation: Dnn-oriented jpeg compression against adversarial examples.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

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*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2020. Generating natural language attacks in a hard label black box setting.

Valentin Malykh. 2019. Robust to noise models in natural language processing tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 10–16, Florence, Italy. Association for Computational Linguistics.

Zhao Meng and Roger Wattenhofer. 2020. A geometry-inspired attack for generating natural language adversarial examples. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6679–6689, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

D. Naber. 2003. *A Rule-Based Style and Grammar Checker*. GRIN Verlag.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *RepEval*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. 2019. Barrage of random transforms for adversarially robust defense. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6521–6530.

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. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5531–5546, Online. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, et al. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

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

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. 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 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics.