Defending Pre-trained Language Models from Adversarial Word Substitution Without Performance Sacrifice

Rongzhou Bao, Jiayi Wang, Hai Zhao*

Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
MoE Key Lab of Artificial Intelligence, AI Institute,
Shanghai Jiao Tong University, Shanghai, China
rongzhou.bao@outlook.com
{wangjiayi_102_23}@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

Abstract

Pre-trained contextualized language models (PrLMs) have led to strong performance gains in downstream natural language understanding tasks. However, PrLMs can still be easily fooled by adversarial word substitution, which is one of the most challenging textual adversarial attack methods. Existing defence approaches suffer from notable performance loss and complexities. Thus, this paper presents a compact and performance-preserved framework, **A**nomaly **D**etection with **F**requency-**A**ware **R**andomization (ADFAR). In detail, we design an auxiliary anomaly detection classifier and adopt a multi-task learning procedure, by which PrLMs are able to distinguish adversarial input samples. Then, in order to defend adversarial word substitution, a frequency-aware randomization process is applied to those recognized adversarial input samples. Empirical results show that ADFAR significantly outperforms those newly proposed defense methods over various tasks with much higher inference speed. Remarkably, ADFAR does not impair the overall performance of PrLMs. The code is available at https://github.com/LilyNLP/ADFAR.

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in various areas. However, previous works show that DNNs are vulnerable to adversarial samples (Goodfellow et al., 2015; Kurakin et al., 2017; Wang et al., 2021), which are inputs with small, intentional modifications that cause the model to make false predictions. Pre-trained language models (PrLMs) (Devlin et al., 2019; Liu et al., 2019; Clark et al., 2020; Zhang et al., 2020, 2019) are widely adopted as an essential component for various NLP systems. However, as DNN-based models, PrLMs can still be easily fooled by textual adversarial samples (Wallace et al., 2019; Jin et al., 2019; Nie et al., 2020; Zang et al., 2020). Such vulnerability of PrLMs keeps raising potential security concerns, therefore researches on defense techniques to help PrLMs against textual adversarial samples are imperatively needed.

Different kinds of textual attack methods have been proposed, ranging from character-level word misspelling (Gao et al., 2018), word-level substitution (Alzantot et al., 2018; Ebrahimi et al., 2018; Ren et al., 2019; Jin et al., 2019; Zang et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020), phrase-level insertion and removal (Liang et al., 2018), to sentence-level paraphrasing (Ribeiro et al., 2018; Iyyer et al., 2018). Thanks to the discrete nature of natural language, attack approaches that result in illegal or unnatural sentences can be easily detected and restored by spelling correction and grammar error correction (Islam and Inkpen, 2009; Sakaguchi et al., 2017; Pruthi et al., 2019). However, attack approaches based on adversarial word substitution can produce high-quality and efficient adversarial samples which are still hard to be detected by existing methods. Thus, the adversarial word substitution keeps posing a larger and more profound challenge for the robustness of PrLMs. Therefore, this paper is devoted to overcome the challenge posed by adversarial word substitution.

Several approaches are already proposed to mitigate issues posed by adversarial word substitution (Zhou et al., 2019; Jia et al., 2019; Huang et al., 2019; Cohen et al., 2019; Ye et al., 2020; Si et al., 2021). Although these defense methods manage to alleviate the negative impact of adversarial word

[*] Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011). This work was supported by Huawei Noah's Ask Lab

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3248–3258
August 1–6, 2021. ©2021 Association for Computational Linguistics

substitution, they sometimes reduce the prediction accuracy for non-adversarial samples to a notable extent. Given the uncertainty of the existence of attack in real application, it is impractical to sacrifice the original prediction accuracy for the purpose of defense. Moreover, previous defense methods either have strong limitations over the attack space to certify the robustness, or require enormous computation resources during training and inference. Thus, it is imperatively important to find an efficient performance-preserved defense method.

For such purpose, we present a compact and performance-preserved framework, **A**nomaly **D**etection with **F**requency-**A**ware **R**andomization (ADFAR), to help PrLMs defend against adversarial word substitution without performance sacrifice. Xie et al. (2018) show that introducing randomization at inference can effectively defend adversarial attacks. Moreover, (Mozes et al., 2020) indicate that the usual case for adversarial samples is replacing words with their less frequent synonyms, while PrLMs are more robust to frequent words. Therefore, we propose a frequency-aware randomization process to help PrLMs defend against adversarial word substitution.

However, simply applying a randomization process to all input sentences would reduce the prediction accuracy for non-adversarial samples. In order to preserve the overall performance, we add an auxiliary anomaly detector on top of PrLMs and adopt a multi-task learning procedure, by which PrLMs are able to determine whether each input sentence is adversarial or not, and not introduce extra model. Then, only those adversarial input sentences will undergo the randomization procedure, while the prediction process for non-adversarial input sentences remains the same.

Empirical results show that as a more efficient method, ADFAR significantly outperforms previous defense methods (Ye et al., 2020; Zhou et al., 2019) over various tasks, and preserves the prediction accuracy for non-adversarial sentences. Comprehensive ablation studies and analysis further prove the efficiency of our proposed method, and indicate that the adversarial samples generated by current heuristic word substitution strategy can be easily detected by the proposed auxiliary anomaly detector.

2 Related Work

2.1 Adversarial Word Substitution

Adversarial word substitution (AWS) is one of the most efficient approaches to attack advanced neural models like PrLMs. In AWS, an attacker deliberately replaces certain words by their synonyms to mislead the prediction of the target model. At the same time, a high-quality adversarial sample should maintain grammatical correctness and semantic consistency. In order to craft efficient and high-quality adversarial samples, an attacker should first determine the vulnerable tokens to be perturbed, and then choose suitable synonyms to replace them.

Current AWS models (Alzantot et al., 2018; Ebrahimi et al., 2018; Ren et al., 2019; Jin et al., 2019; Li et al., 2020; Garg and Ramakrishnan, 2020) adopt heuristic algorithms to locate vulnerable tokens in sentences. To illustrate, for a given sample and a target model, the attacker iteratively masks the tokens and checks the output of the model. The tokens which have significant influence on the final output logits are regarded as vulnerable.

Previous works leverage word embeddings such as GloVe (Pennington et al., 2014) and counter-fitted vectors (Mrkšić et al., 2016) to search the suitable synonym set of a given token. Li et al. (2020); Garg and Ramakrishnan (2020) uses BERT (Devlin et al., 2019) to generate perturbation for better semantic consistency and language fluency.

2.2 Defense against AWS

For general attack approaches, adversarial training (Goodfellow et al., 2015; Jiang et al., 2020) is widely adopted to mitigate adversarial effect, but (Alzantot et al., 2018; Jin et al., 2019) shows that this method is still vulnerable to AWS. This is because AWS models leverage dynamic algorithms to attack the target model, while adversarial training only involves a static training set.

Methods proposed by Jia et al. (2019); Huang et al. (2019) are proved effective for defence against AWS, but they still have several limitations. In these methods, Interval Bound Propagation (IBP) (Dvijotham et al., 2018), an approach to consider the worst-case perturbation theoretically, is leveraged to certify the robustness of models. However, IBP-based methods can only achieve the certified robustness under a strong limitation over the attack space. Furthermore, they are difficult to adapt to PrLMs for their strong reliance on the assumption

🟥 Rare word substitution 🟦 Random substitution

Original	The most **hopelessly monotonous** film, **noteworthy** only for the **gimmick** of being filmed as a **single unbroken** 87 **minute** take.
Randomized	The most **desperately boring** film, **amazing** only for the **trick** of being filmed as a **unitary continuous** 87 **minutes** take.
Original	A small **gem** of a **movie** that **defies classification** and is as **thought** provoking as it is funny, scary and **sad**.
Randomized	A small **essence** of a **film** that **challenge ranking** and is as **thinking** provoking as it is funny, scary and **gloomy**.

Figure 1: Frequency-aware randomization examples.

of model architecture.

Two effective and actionable methods (DISP (Zhou et al., 2019) and SAFER Ye et al. (2020)) are proposed to overcome the challenge posed by AWS, and therefore adopted as the baselines for this paper. DISP (Zhou et al., 2019) is a framework based on perturbation discrimination to block adversarial attack. In detail, when facing adversarial inputs, DISP leverages two auxiliary PrLMs: one to detect perturbed tokens in the sentence, and another to restore the abnormal tokens to original ones. Inspired by randomized smoothing (Cohen et al., 2019), Ye et al. (2020) proposes SAFER, a novel framework that guarantees the robustness by smoothing the classifier with synonym word substitution. To illustrate, based on random word substitution, SAFER smooths the classifier by averaging its outputs of a set of randomly perturbed inputs. SAFER outperforms IBP-based approaches and can be easily applied to PrLMs.

2.3 Randomization

In recent years, randomization has been used as a defense measure for deep learning in computer vision (Xie et al., 2018). Nevertheless, direct extensions of these measures to defend against textual adversarial samples are not achievable, since the text inputs are discrete rather than continuous. Ye et al. (2020) indicates the possibility of extending the application of the randomization approach to NLP by randomly replacing the words in sentences with their synonyms.

3 Method

3.1 Frequency-aware Randomization

Since heuristic attack methods attack a model by substituting each word iteratively until it successfully alters the model's output, it is normally difficult for static strategies to defense such kind of dynamic process. Rather, dynamic strategies, such as randomization, can better cope with the problem. It is also observed that replacing words with their more frequent alternatives can better mitigate the adversarial effect and preserve the original performance. Therefore, a frequency-aware randomization strategy is designed to perplex AWS strategy.

Figure 1 shows several examples of the frequency-aware randomization. The proposed approach for the frequency-aware randomization is shown in Algorithm 1, and consists of three steps. Firstly, rare words with lower frequencies and a number of random words are selected as substitution candidates. Secondly, we choose synonyms with the closest meanings and the highest frequencies to form a synonym set for each candidate word. Thirdly, each candidate word is replaced with a random synonym within its own synonym set. To quantify the semantic similarity between two words, we represent words with embeddings from (Mrkšić et al., 2016), which is specially designed for synonyms identification. The semantic similarity of two words are evaluated by cosine similarity of their embeddings. To determine the frequency of a word, we use a frequency dictionary provided by FrequencyWords Repository[*].

3.2 Anomaly Detection

Applying the frequency-aware randomization process to every input can still reduce the prediction accuracy for normal samples. In order to overcome this issue, we add an auxiliary anomaly detection head to PrLMs and adopt a multi-task learning procedure, by which PrLMs are able to classify the input text and distinguish the adversarial samples at the same time, and not introduce extra model. In inference, the frequency-aware randomization only applied to the samples that are detected as adversarial. In this way, the reduction of accuracy is largely avoided, since non-adversarial samples are not affected.

Zhou et al. (2019) also elaborates the idea of perturbation discrimination to block attack. However, their method detects anomaly on token-level and requires two resource-consuming PrLMs for detection and correction, while ours detects anomaly on sentence-level and requires no extra models. Com-

[*] https://github.com/hermitdave/FrequencyWords

Algorithm 1 Frequency-aware Randomization

```
Input: Sentence $X = \{w_1, w_2, ..., w_n\}$, word embeddings $Emb$ over the vocabulary $Vocab$
Output: Randomized sentence $X_{rand}$
Initialization: $X_{rand} \leftarrow X$
Create a set $W_{rare}$ of all rare words with frequencies less than $f_{thres}$, denote $n_{rare} = |W_{rare}|$.
Create a set $W_{rand}$ by randomly selecting $n * r - n_{rare}$ words $w_j \notin W_{rare}$, where $r$ is the pre-defined ratio of substitution.
Create the substitution candidates set, $W_{sub} \leftarrow W_{rare} + W_{rand}$, and $|W_{sub}| = n * r$.
Filter out the stop words in $W_{sub}$.
for each word $w_i$ in $W_{sub}$ do
    Create a set $S$ by extracting the top $n_s$ synonyms using $CosSim(Emb_{w_i}, Emb_{w_{word}})$ for each word in $Vocab$.
    Create a set $S_{freq}$ by selecting the top $n_f$ frequent synonyms from $S$.
    Randomly choose one word $w_s$ from $S$.
    $X_{rand} \leftarrow$ Replace $w_i$ with $w_s$ in $X_{rand}$.
end for
```

pared to Zhou et al. (2019), our method is two times faster in inference speed and can achieve better accuracy for sentence-level anomaly detection.

3.3 Framework

In this section, we elaborate the framework of ADFAR in both training and inference.

3.3.1 Training

Figure 2 shows the framework of ADFAR in training. We extend the baseline PrLMs by three major modifications: 1) the construction of training data, 2) the auxiliary anomaly detector and 3) the training objective, which will be introduced in this section.

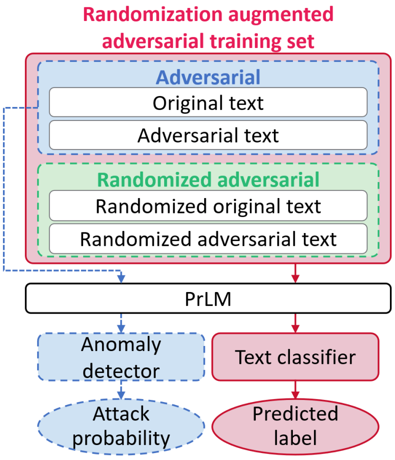


Figure 2: Framework of ADFAR in training.

Construction of Training Data As shown in Figure 2, we combine the idea of both adversarial training and data augmentation (Wei and Zou, 2019) to construct our randomization augmented adversarial training data. Firstly, we use a heuristic AWS model (e.g. TextFooler) to generate adversarial samples based on the original training set. Following the common practice of adversarial training, we then combine the adversarial samples with the original ones to form an adversarial training set. Secondly, in order to let PrLMs better cope with randomized samples in inference, we apply the frequency-aware randomization on the adversarial training set to generate a randomized adversarial training set. Lastly, the adversarial training set and the randomized adversarial training set are combined to form a randomization augmented adversarial training set.

Auxiliary Anomaly Detector In addition to the original text classifier, we add an auxiliary anomaly detector to the PrLMs to distinguish adversarial samples. For an input sentence, the PrLMs captures the contextual information for each token by self-attention and generates a sequence of contextualized embeddings $\{h_0, \ldots h_m\}$. For text classification task, $h_0 \in R^H$ is used as the aggregate sequence representation. The original text classifier leverages h_0 to predict the probability that X is labeled as class $\hat{y_c}$ by a logistic regression with softmax:

$$\begin{aligned} y_c &= Prob(\hat{y_c}|x), \\ &= \text{softmax}(W_c(dropout(h_0)) + b_c), \end{aligned}$$

For the anomaly detector, the probability that X is labeled as class $\hat{y_d}$ (if X is attacked, $\hat{y_d} = 1$; if X is normal, $\hat{y_d} = 0$) is predicted by a logistic regression with softmax:

$$\begin{aligned} y_d &= Prob(\hat{y_d}|x), \\ &= \text{softmax}(W_d(dropout(h_0)) + b_d), \end{aligned}$$

As shown in Figure 2, the original text classifier is trained on the randomization augmented adversarial training set, whereas the anomaly detector is only trained on the adversarial training set.

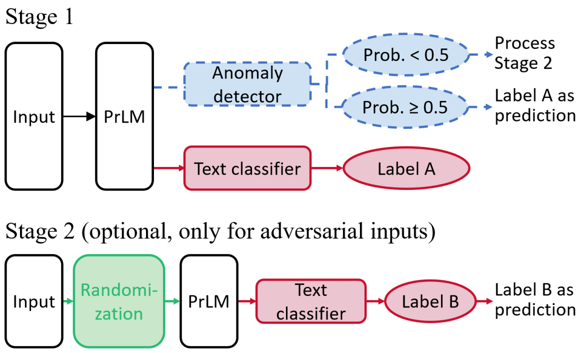


Figure 3: Framework of ADFAR in inference.

Training Objective We adopt a multi-task learning framework, by which PrLM is trained to classify the input text and distinguish the adversarial samples at the same time. We design two parallel training objectives in the form of minimizing cross-entropy loss: $loss_c$ for text classification and $loss_d$ for anomaly detection. The total loss function is defined as their sum:

$$\begin{aligned}
loss_c &= -[y_c * \log(\hat{y_c}) + (1 - y_c) * \log(1 - \hat{y_c})] \\
loss_d &= -[y_d * \log(\hat{y_d}) + (1 - y_d) * \log(1 - \hat{y_d})] \\
Loss &= loss_c + loss_d
\end{aligned}$$

3.3.2 Inference

Figure 3 shows the framework of ADFAR in inference. Firstly, the anomaly detector predicts whether an input sample is adversarial. If the input sample is determined as non-adversarial, the output of the text classifier (Label A) is directly used as its final prediction. If the input sample is determined as adversarial, the frequency-aware randomization process is applied to the original input sample. Then, the randomized sample is sent to the PrLM again, and the second output of the text classifier (Label B) is used as its final prediction.

4 Experimental Implementation

4.1 Tasks and Datasets

Experiments are conducted on two major NLP tasks: text classification and natural language inference. The dataset statistics are displayed in Table 1. We evaluate the performance of models on the non-adversarial test samples as the original accuracy. Then we measure the after-attack accuracy of models when facing AWS. By comparing these two accuracy scores, we can evaluate how robust the model is.

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.

Text Classification We use three text classification datasets with average text lengths from 20 to 215 words, ranging from phrase-level to document-level tasks. **SST2** (Socher et al., 2013): phrase-level binary sentiment classification using fine-grained sentiment labels on movie reviews. **MR** (Pang and Lee, 2005): sentence-level binary sentiment classification on movie reviews. We take 90% of the data as training set and 10% of the data as test set as (Jin et al., 2019). **IMDB** (Maas et al., 2011): document-level binary sentiment classification on movie reviews.

Natural Language Inference NLI aims at determining the relationship between a pair of sentences based on semantic meanings. We use Multi-Genre Natural Language Inference (**MNLI**) (Nangia et al., 2017), a widely adopted NLI benchmark with coverage of transcribed speech, popular fiction, and government reports.

4.2 Attack Model and Baselines

We use TextFooler[†](Jin et al., 2019) as the major attack model for AWS. Moreover, we implement (Ren et al., 2019) and GENETIC (Alzantot et al., 2018) based on the TextAttack (Morris et al., 2020) code base to further verify the efficiency of our proposed method.

We compare ADFAR with DISP (Zhou et al., 2019) and SAFER (Ye et al., 2020). The implementation of DISP is based on the repository offered by Zhou et al. (2019). For SAFER, we also leverage the code proposed by Ye et al. (2020). Necessary modifications are made to evaluate these methods' performance under heuristic attack models.

4.3 Experimental Setup

The implementation of PrLMs is based on PyTorch[‡]. We leverage, BERT$_{\text{BASE}}$ (Devlin et al., 2019), RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) and ELECTRA$_{\text{BASE}}$ (Clark et al., 2020) as baseline

[†] https://github.com/jind11/TextFooler
[‡] https://github.com/joey1993/bert-defender
[§] https://github.com/lushleaf/Structure-free-certified-NLP

Model	MR		SST2		IMDB		MNLI	
	Orig. Acc.	Adv. Acc.	Orig. Acc.	Adv. Acc.	Orig. Acc.	Adv. Acc.	Orig. Acc.	Adv. Acc.
BERT	86.2	16.9	**93.1**	39.8	92.4	12.4	**84.0**	11.3
BERT + Adv Training	85.6	34.6	92.6	48.8	92.2	34.2	82.3	33.4
BERT + DISP	82.0	42.2	91.6	70.4	91.7	82.0	76.3	35.1
BERT + SAFER	79.0	55.4	91.3	75.6	91.3	88.1	82.1	54.7
BERT + ADFAR	**86.6**	**66.0**	92.4	**75.6**	**92.8**	**89.2**	82.6	**67.8**

Table 2: The performance of ADFAR and other defense frameworks using BERT$_{\text{BASE}}$ as PrLM and TextFooler as attack model. Orig. Acc. is the prediction accuracy of normal samples and Adv. Acc. is the after-attack accuracy of models when facing AWS. The results are based on the average of five runs.

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 epochs is set to 5.

For the frequency-aware randomization process, we set $f_{thres} = 200$, $n_s = 20$ and $n_f = 10$. In the adopted frequency dictionary, 5.5k out of 50k words have a frequency lower than $f_{thres} = 200$ and therefore regarded as rare words. r is set to different values for training (25%) and inference (30%) due to different aims. In training, to avoid introducing excessive noise and reduce the prediction accuracy for non-adversarial samples, r is set to be relatively low. On the contrary, in inference, our aim is to perplex the heuristic attack mechanism. The more randomization we add, the more perplexities the attack mechanism receives, therefore we set a relatively higher value for r. More details on the choice of these hyperparameters will be discussed in the analysis section.

5 Experimental Results

5.1 Main results

Following (Jin et al., 2019), we leverage BERT$_{\text{BASE}}$ (Devlin et al., 2019) as baseline PrLM and TextFooler as attack model. Table 2 shows the performance of ADFAR and other defense frameworks. Since randomization may lead to a variance of the results, we report the results based on the average of five runs. Experimental results indicate that ADFAR can effectively help PrLM against AWS. Compared with DISP (Zhou et al., 2019) and SAFER (Ye et al., 2020), ADFAR achieves the best performance for adversarial samples. Meanwhile, ADFAR does not hurt the performance for non-adversarial samples in general. On tasks such as MR and IMDB, ADFAR can even enhance the baseline PrLM.

DISP leverages two extra PrLMs to discriminate and recover the perturbed tokens, which introduce extra complexities. SAFER makes the prediction of an input sentence by averaging the prediction results of its perturbed alternatives, which multiply the inference time. As shown in Table 3, compared with previous methods, ADFAR achieves a significantly higher inference speed.

Model	Parameters	Inference Time
BERT$_{\text{BASE}}$	110M	15.7ms (100%)
BERT$_{\text{BASE}}$ + DISP	330M	38.9ms (247%)
BERT$_{\text{BASE}}$ + SAFER	110M	27.6ms (176%)
BERT$_{\text{BASE}}$ + ADFAR	110M	18.1ms (115%)

Table 3: Parameters and Inference Time statistics. The inference time indicate the average inference time for one sample in MR dataset using one NVIDIA RTX3090.

5.2 Results with Different Attack Strategy

Since ADFAR leverages the adversarial samples generated by TextFooler (Jin et al., 2019) in training, it is important to see whether ADFAR also performs well when facing adversarial samples generated by other AWS models. We leverage PWWS (Ren et al., 2019) and GENETIC (Alzantot et al., 2018) to further study the performance of ADFAR.

Attack	MR		SST2	
	BERT	+ADFAR	BERT	+ADFAR
Attack-Free	86.2	86.6	93.1	92.3
PWWS	34.2	74.2	54.3	80.5
Genetic	21.3	70.4	38.7	72.2
TextFooler	16.9	66.0	39.8	73.8

Table 4: Performance of BERT and BERT with ADFAR when facing various AWS models. The results are based on the average of five runs.

As shown is Table 4, the performance of ADFAR is not affected by different AWS models, which further proves the efficacy of our method.

[¶] https://github.com/huggingface

5.3 Results with Other PrLMs

Table 5 shows the performance of ADFAR leveraging RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) and ELECTRA$_{\text{BASE}}$ (Clark et al., 2020) as PrLMs. In order to enhance the robustness and performance of the PrLM, RoBERTa extends BERT with a larger corpus and using more efficient parameters, while ELECTRA applies a GAN-style architecture for pre-training. Empirical results indicate that ADFAR can further improve the robustness of RoBERTa and ELECTRA while preserving their original performance.

PrLM	MR		SST2	
	Orig. Acc.	Adv. Acc.	Orig. Acc.	Adv. Acc.
BERT	86.2	16.9	93.1	39.8
+ADFAR	86.6	66.0	92.3	73.8
RoBERTa	88.3	30.4	93.4	37.4
+ADFAR	87.2	71.0	93.2	77.6
ELECTRA	90.1	33.6	94.2	40.4
+ADFAR	90.4	71.2	95.0	83.0

Table 5: Results based with various PrLMs.

6 Analysis

6.1 Ablation Study

ADFAR leverages three techniques to help PrLMs defend against adversarial samples: adversarial training, frequency-aware randomization and anomaly detection. To evaluate the contributions of these techniques in ADFAR, we perform ablation studies on MR and SST2 using BERT$_{\text{BASE}}$ as our PrLMs, and TextFooler as the attack model. As shown in Table 6, the frequency-aware randomization is the key factor which helps PrLM defense against adversarial samples, while anomaly detection plays an important role in preserving PrLM's prediction accuracy for non-adversarial samples.

Model	MR		SST2	
	Orig. Acc.	Adv. Acc.	Orig. Acc.	Adv. Acc.
BERT	86.2	16.9	93.1	39.8
+ Adv	85.6	34.6	92.6	48.8
+ FR	85.0	72.8	90.6	82.6
+ AD	86.6	66.0	92.3	73.8

Table 6: Ablation study on MR and SST2 using BERT$_{\text{BASE}}$ as PrLM, and TextFooler as attack model. Adv represents adversarial training, FR indicates frequency-aware randomization and AD means anomaly detection. The results are based on the average of five runs.

6.2 Anomaly Detection

In this section, we compare the anomaly detection capability between ADFAR and DISP (Zhou et al., 2019). ADFAR leverages an auxiliary anomaly detector, which share a same PrLM with the original text classifier, to discriminate adversarial samples. DISP uses an discriminator based on an extra PrLMs to identify the perturbed adversarial inputs, but on token level. For DISP, in order to detect anomaly on sentence level, input sentences with one or more than one adversarial tokens identified by DISP are regarded as adversarial samples. We respectively sample 500 normal and adversarial samples from the test set of MR and SST to evaluate the performance of ADFAR and DISP for anomaly detection.

Table 7 shows the performance of ADFAR and DISP for anomaly detection. Empirical results show that ADFAR can predict more precisely, since it achieves a significantly higher F_1 score than DISP. Moreover, ADFAR has a simpler framework, as its anomaly detector shares the same PrLM with the classifier, while DISP requires an extra PrLM. The results also indicate that the current heuristic AWS strategy is vulnerable to our anomaly detector, which disproves the claimed undetectable feature of this very adversarial strategy.

Method	MR			SST2		
	Precision	Recall	F_1	Precision	Recall	F_1
DISP	68.0	92.0	73.2	59.5	94.2	72.9
ADFAR	90.1	84.0	86.9	88.0	90.0	88.9

Table 7: Performance for anomaly detection.

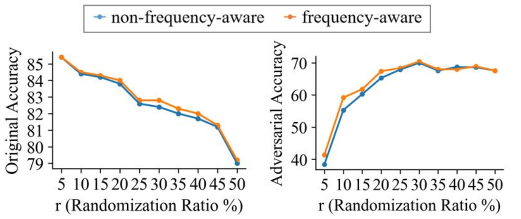
6.3 Effect of Randomization Strategy

As the ablation study reveals, the frequency-aware randomization contributes the most to the defense. In this section, we analyze the impact of different hyperparameters and strategies adopted by the frequency-aware randomization approach, in inference and training respectively.

6.3.1 Inference

The frequency-aware randomization process is applied in inference to mitigate the adversarial effects. Substitution candidate selection and synonym set construction are two critical steps during this process, in which two hyperparameters (r and n_s) and the frequency-aware strategy are examined.

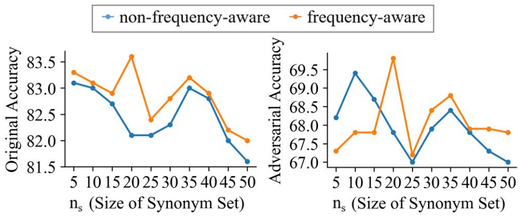
Selection of Substitution Candidates The influence of different strategies for substitution candidate selection in inference is studied in this section. The impact of two major factors are measured: 1) the substitution ratio r and 2) whether to apply a frequency-aware strategy. In order to exclude the disturbance from other factors, we train BERT on the original training set and fix n_s to 20. Firstly, we alter the value of r from 5% to 50%, without applying the frequency-aware strategy. As illustrated by the blue lines in Figure 4, as r increases, the original accuracy decreases, while the adversarial accuracy increases and peaks when r reaches 30%. Secondly, a frequency-aware strategy is added to the experiment, with $f_{thres} = 200$. As depicted by the yellow lines in Figure 4, both original and adversarial accuracy, the general trends coincide with the non-frequency-aware scenario, but overall accuracy is improved to a higher level. The highest adversarial is obtained when r is set to 30% using frequency-aware strategy.



(a) Results on normal samples (b) Results on adversarial samples

Figure 4: Effect of the substitution ratio r and the frequency-aware strategy in substitution candidate selection during inference.

Construction of Synonym Set The influence of different strategies for synonym set construction in inference is evaluated in this section. The impact of two major factors are measured: 1) the size of a single synonym set n_s and 2) whether to apply a frequency-aware strategy. In order to exclude the disturbance from other factors, we train BERT on the original training set and fix r to 30% . Firstly, we alter the value of n_s from 5 to 50, without applying the frequency-aware strategy. The resulted original and adversarial accuracy are illustrated by the blue lines in Figure 5. Secondly, a frequency-aware strategy is added to the experiment, with $n_f = 50\% * n_s$. As depicted by the yellow lines in Figure 5, the original accuracy and the adversarial accuracy both peaks when $n_s = 20$, and the overall accuracy is improved to a higher level compared to the non-frequency-aware scenario.

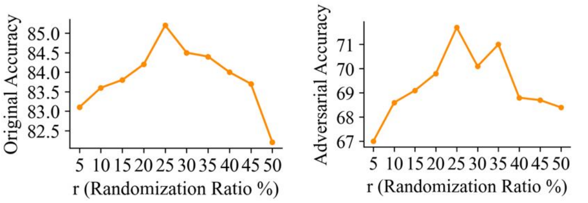


(a) Results on normal samples (b) Results on adversarial samples

Figure 5: Effect of the size of synonym set n_s and the frequency-aware strategy in construction of synonym set.

6.3.2 Training

The frequency-aware randomization process is applied in training to augment the training data, and hereby enables the PrLM to better cope with randomized samples inference. Based on this purpose, the frequency-aware randomization process in training should resemble the one in inference as much as possible. Therefore, here we set an identical process for synonym set construction, i.e. $n_s = 20$ and $n_f = 50\% * n_s$. However, for the substitution selection process, to avoid introducing excessive noise and maintain the accuracy for the PrLM, the most suitable substitution ratio r might be different than the one in inference. Experiments are conducted to evaluate the influence of r in training. We alter the value of r from 5% to 50%. In Figure 6, we observe that $r = 25\%$ results in highest original and adversarial accuracy.



(a) Results on normal samples (b) Results on adversarial samples

Figure 6: Effect of the size of synonym set n_s and the frequency-aware strategy in construction of synonym set.

7 Conclusion

This paper proposes ADFAR, a novel framework which leverages the frequency-aware randomization and the anomaly detection to help PrLMs defend against adversarial word substitution. Empirical results show that ADFAR significantly outperforms those newly proposed defense methods over various tasks. Meanwhile, ADFAR achieves a remarkably higher inference speed and does not reduce the prediction accuracy for non-adversarial sentences, from which we keep the promise for this research purpose.

Comprehensive ablation study and analysis indicate that 1) Randomization is an effective method to defend against heuristic attack strategy. 2) Replacement of rare words with their more common alternative can help enhance the robustness of PrLMs. 3) Adversarial samples generated by current heuristic adversarial word substitution models can be easily distinguished by the proposed auxiliary anomaly detector. We hope this work could shed light on future studies on the robustness of PrLMs.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320.

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

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. 2018. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

J. Gao, J. Lanchantin, M. L. Soffa, and Y. 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1905.07129*.

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.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using Google Web 1T 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249, Singapore. 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, New Orleans, Louisiana. Association for Computational Linguistics.

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

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

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.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

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

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.

John 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis D. Griffin. 2020. Frequency-guided word substitutions for detecting textual adversarial examples. *arXiv preprint arXiv:2004.05887*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

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*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. 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.

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, 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, Melbourne, Australia. Association for Computational Linguistics.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning.

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.

Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2021. Towards a robust deep neural network in texts: A survey.

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.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

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.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2019. Sg-net: Syntax-guided machine reading comprehension.

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)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.