Word Level Robustness Enhancement: Fight Perturbation with Perturbation

Pei Huang,[1,2*] **Yuting Yang,**[2,4*] **Fuqi Jia,**[1,2] **Minghao Liu,**[1,2] **FeiFei Ma,**[1,2,3†] **Jian Zhang**[1,2†]

[1]State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences (ISCAS), Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
[3] Laboratory of Parallel Software and Computational Science, ISCAS, Beijing, 100190, China
[4] Key Lab of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
yangyuting@ict.ac.cn, {huangpei, jiafq, liumh, maff, zj}@ios.ac.cn

Abstract

State-of-the-art deep NLP models have achieved impressive improvements on many tasks. However, they are found to be vulnerable to some perturbations. In this paper, we design a robustness enhancement method to defend against word substitution perturbation, whose basic idea is to fight perturbation with perturbation. We find that: although many well-trained deep models are not robust in the setting of the presence of adversarial samples, they satisfy weak robustness. That means they can handle most non-crafted perturbations well. Taking advantage of the weak robustness property of deep models, we utilize non-crafted perturbations to resist the adversarial perturbations crafted by attackers. Our method contains two main stages. The first stage is using randomized perturbation to conform the input to the data distribution. The second stage is using randomized perturbation to eliminate the instability of prediction results and enhance the robustness guarantee. Experimental results show that our method can significantly improve the ability of deep models to resist the state-of-the-art adversarial attacks while maintaining the prediction performance on the original clean data.

Introduction

Deep neural networks (DNNs) have been broadly applied in various domains. However, they are vulnerable to adversarial examples that are intentionally crafted by attackers aiming at misleading the prediction result (Goodfellow, Shlens, and Szegedy 2015). The vulnerability of deep neural networks has been exposed in many NLP tasks, including text classification (Samanta and Mehta 2017; Liang et al. 2018; Alzantot et al. 2018), machine translation (Zhao, Dua, and Singh 2018; Cheng et al. 2020), dialogue systems (Cheng, Wei, and Hsieh 2019), reading comprehension (Jia and Liang 2017), and dependency parsing (Zheng et al. 2020). In particular, as deep learning-based models are increasingly used in safety-critical applications, the vulnerability of deep models has raised concerns as a critical issue.

In recent years, a series of adversarial attack algorithms have been proposed to interfere with the predictions of the network, ranging from character-level word misspelling, word-level substitution, phrase-level insertion and removal, to sentence-level paraphrasing. Unlike the image area, attack approaches that result in illegal text sentences can be easily detected and restored by spelling correction and grammar error correction (Islam and Inkpen 2009; Sakaguchi, Post, and Durme 2017; Pruthi, Dhingra, and Lipton 2019). Among these attacks, word substitution attack is hard to be detected at the grammatical level. The hacker will craft adversarial examples by replacing words with their synonyms in an input text to deceive the model while maintaining semantic and fluency. Therefore, word substitution-based attacks have attracted many researchers and continue to pose a profound challenge for the robustness of deep NLP models.

Several empirical approaches have been proposed to defend against word perturbation attacks. For example, adversarial training (Wang et al. 2021) incorporates adversarial examples into training and can maintain a greater degree of accuracy in NLP. However, its defensive efficiency is very limited and may fall when encountering stronger attacks. Certified defense methods like interval bound propagation (IBP) (Jia et al. 2019; Huang et al. 2019) and randomized smoothing (Cohen, Rosenfeld, and Kolter 2019) have been applied to improve the robustness of models in the worst case, but they may reduce the prediction accuracy for clean data to a notable extent. Many studies show that adversarial robustness may be odd at accuracy (Raghunathan et al. 2020; Tsipras et al. 2019). Defending against attacks effectively while maintaining clean accuracy is the main difficulty for robustness improvement.

Although deep models are vulnerable to word substitution attacks, they generalize well to the noised input in practice. While investigating the adversarial examples from the view of quantification, an interesting phenomenon appears. That is, there is only a small proportion of adversarial examples in the perturbation space around the data distribution. Take BERT trained on IMDB task as an example, it achieves 92.27% prediction accuracy but can be attacked with a high probability of 87.1%. However, we find that less than 1/3 samples are adversarial examples for over 97.61% perturbation spaces. We call this property weak robustness (Figure 1), which means the model is stable to most random perturbations. Although the model is not robust enough to adver-

*These authors contributed equally.
†Corresponding Authors

sarial perturbation, the adversarial example itself may not be robust to perturbation either. Inspired by fighting fire with fire, we propose a defense framework, **F**ight **P**erturbation with **P**erturbation (**FPP**), to enhance the weakly robust classifier f to a strong defense model F.

FPP contains two steps: (1) Randomly perturb the user's input sequence X according to the n-gram frequency information in the data distribution. (2) Apply random word substitutions to X and utilize the ensemble of samples to make a prediction. These two steps are stochastic processes, which increase the analysis difficulty for hackers in developing their attack methods. The purpose of the first step is to conform the input to the data distribution and block the transferability of adversarial inputs generated for the base model or other models. (Mozes et al. 2021) shows that the usual case for adversarial samples is replacing words with the combination of their less frequent synonyms. The perturbation in the first step will block such a situation. The second step is inspired by randomized smoothing (Cohen, Rosenfeld, and Kolter 2019; Ye, Gong, and Liu 2020) which is used to certify robustness. However, FPP does not calculate the upper bound and lower bound of the probability of each class. Our second step is to make predictions based on the voting results of samples generated by random substitutions. This step can reduce the randomness introduced in the first step and obtain a rigorous robustness guarantee.

Overall, our robustness enhancement strategy has several advantages: (1) It can be easily applied to various pre-trained language models as long as they satisfy weak robustness well. It is only based on the statistical properties of the model and not on the structure. (2) It is scalable to large deep neural networks. (3) Compared with some empirical defense methods, it has good interpretability for robustness enhancement.

We experiment on various model architectures and multiple data sets. The experimental results show that our robustness enhancement method FPP achieves better performance in accuracy and defense capability compared with existing defense methods. In particular, for LSTM trained on IMDB task, FPP can maintain prediction accuracy with only a decrease of 0.64% and reduce the successful attacking ratio to 3.7% (an improvement of 10.27% compared with the existing defense methods) under a strong attacking method TextFooler. With respect to the comprehensive robustness accuracy, FPP achieves 91% with more than 14% improvement compared with existing methods, indicating its superiority in the trade-off between prediction accuracy and defense capability.

Preliminary

Adversarial Example

Given a natural language classifier $f : \mathcal{X} \to \mathcal{Y}$, which is a mapping function from an input space to an output label space. The input space $\mathcal{X}$ contains all possible texts $X = w_1, w_2, ..., w_n$ and output space $\mathcal{Y} = \{y_1, y_2, ..., y_c\}$ contains c possible predictions of an input. w_i is usually a word embedding or one-hot vector. $f_y(\cdot)$ is the prediction score for the y label. Let $P = \{p_1, p_2, ..., p_m\}$ be the set of perturbable positions. For each perturbable position $p \in P$, there is a set $S(X, p)$ which contains all candidate words for substitution without changing the semantics (original word w_p is also in $S(X, p)$). Assuming that $X' = w'_1, w'_2, ..., w'_n$ is a text generated by perturbing X and y^* is the gold label of X, then we say that X' is an adversarial example if:

$$f(X') \neq y^*$$

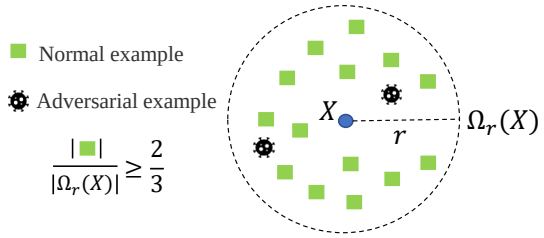

Figure 1: Diagram of Weak Robustness.

Robustness

Definition 1. *A perturbation space $\Omega_r(X)$ of an input sequence X is a set containing all perturbations X' generated by substituting the original word by candidate words in $S(X, p)$ for each perturbable position $p \in \mathcal{P}$ and $\|X' - X\|_0 \leq r$, where $\|\cdot\|_0$ denotes the number of substituted words.*

Definition 2 (Strong Robustness)**.** *Consider a classifier $F(x)$. Given a sequence X with gold label y^*, classifier f is said to be robust on the perturbation space $\Omega_r(X)$ if the following formula holds:*

$$\forall X'.\ X' \in \Omega_r(X) \Rightarrow f(X') = y^* \tag{1}$$

If formula (1) is proved to be true, the classifier f is safe on $\Omega_r(X)$. In other words, f is certified robust to all possible perturbations at X. This is what a safe model pursues, but sometimes quantifier $\forall$ is too strict. So we relax the $\forall$ in formula 1 to a quantification version: weak robustness.

Definition 3 (Weak Robustness)**.** *If the value of $PR > 2/3$, f is said to be **weakly robust** on the perturbation space $\Omega_r(X)$, where PR is defined as:*

$$PR := \frac{|\{X' : X' \in \Omega_r(X) \wedge f(X') = y^*\}|}{|\Omega_r(X)|} \tag{2}$$

In fact, PR can be any value greater than $1/2$. As $2/3$ is the simplest fractional number greater than $1/2$, we choose it. Besides, if a network can only identify samples in $\Omega_r(X)$ at a level just greater than $1/2$, that means the network is not confident about its results.

As exactly computing PR is time-consuming, we estimate it via a Monte Carlo method (see website[1]). If $\hat{PR}$ is the estimation value, we can ensure that the probability of $\hat{PR}$ deviating from its real value PR by a certain amount ϵ is less than δ. Table 1 shows the percentage of perturbation spaces around the data in the test set that satisfies weak robustness. We set r to the 25% of the length of the sentence, which is the maximum substitution ratio followed by most attack methods.

Model	Dataset	$PR > 2/3$
BiLSTM	MR	96.84%
	IMDB	96.94%
	SNLI	85.95%
BERT	MR	98.88%
	IMDB	97.61%
	SNLI	96.96%

Table 1: The percentage of perturbation spaces around the data in the test set that neural network satisfies weak robustness ($\eta = 0.025$, $\delta = 0.005$).

Method

In Table 1, we show that although a well-trained model f is not robust to adversarial attack, they can satisfy weak robustness on most perturbation spaces. In this section, we make full use of this property to enhance the robustness of the models. The basic idea of our method is to fight fire with fire: using perturbation to resist perturbation. It contains two main steps: randomly perturbing input sequences according to the n-gram frequency and enhancing the final prediction by voting on all samples generated by random word substitutions. The algorithm is presented in Algorithm 1.

Step1: Input Perturbation

We know that a normal input can become an adversarial example via some substitutions. However, is the adversarial example itself robust? An adversarial example is usually generated by a subtle combination of several substitutions. If such a combination is destroyed, the neural network will have more chance to classify it correctly.

So, for an input sentence $X = w_1, w_2, ..., w_n$, we first randomly substitute some words based on the 1-gram and 2-gram frequency of the data distribution. The substitution has two purposes: (1) Destroying the attacker's perturbation on the sentence. (2) Conforming the input X to the data distribution as much as possible. Besides, we also introduce some randomness in this process. Most attack methods are based on iteratively replacing words for searching adversarial examples, and the decision of the next step is based on the information of the current step. So, introducing randomness will increase the difficulty for an attacking algorithm to make decisions.

We use $\mathbb{P}(w|D)$ to denote the frequency of w in the data distribution D which is obtained from the training set. For the position p of X, the **1-gram relative frequency** of each candidate $\hat{w_p}$ in $S(X, p)$ is denoted as:

$$\mathbb{P}_1(\hat{w}_p|D, X) := \frac{\mathbb{P}(\hat{w}_p|D)}{\sum_{w \in S(X,p)} \mathbb{P}(w|D)}$$

We use $\mathbb{P}_2(w, w'|D)$ to denote the frequency of 2-gram (w, w'). The **2-gram relative frequency** of each candidate $\hat{w_p}$ in $S(X, p)$ is:

$$\mathbb{P}_2(\hat{w}_p|w_{p+1}, D, X) := \frac{\mathbb{P}(\hat{w}_p, w_{p+1}|D)}{\sum_{w \in S(X,p)} \mathbb{P}(w, w_{p+1}|D)}$$

Algorithm 1: Enhancement Classifier F

Input:X
Parameter: A base classifier f
Output: Prediction $\widetilde{y}$

1: **for all** $p \in P$ **do**
2: $\quad s \sim U(0, 1)$;
3: $\quad$ **if** $\Delta_{12}(p) > s$ **then**
4: $\quad\quad X_{w_p \leftarrow w_p*}$ \\ Replace w_p via w_p*
5: $\quad$ **end if**
6: **end for**
7: $N \leftarrow -2 \ln \epsilon / (2 * 2/3 - 1)^2$
8: $r \leftarrow \kappa n$
9: **for** $i \leftarrow 0$ **to** $N - 1$ **do**
10: $\quad X_i \sim \Omega_r(X)$;
11: $\quad l_i \leftarrow f(X_i)$;
12: **end for**
13: $\widetilde{y} \leftarrow \arg\max_{y \in \mathcal{Y}} \sum_{i=0}^{N-1} \mathbb{I}(l_i = y)$
14: **return** $\widetilde{y}$

Then we define $\mathbb{P}_{12}(\hat{w}_p|D, X)$ (**synthesized frequency**) to synthesize the relative frequency of 1-gram and 2-gram:

$$\mathbb{P}_{12}(\hat{w}_p|D, X) := (1-\lambda)\mathbb{P}_1(\hat{w}_p|D, X) + \lambda \mathbb{P}_2(\hat{w}_p|w_{p+1}, D, X)$$

where $\lambda \in [0, 1]$. Let w_p* be:

$$w_p* := \underset{\hat{w}_p \in S(X,p)}{\arg\max}\ \mathbb{P}_{12}(\hat{w}_p|D, X)$$

which is the word with maximum synthesized frequency in $S(X, p)$. Let Δ_{12} be the difference of synthesized frequency of w_p* and w_p:

$$\Delta_{12}(p) := \mathbb{P}_{12}(w_p * |D, X) - \mathbb{P}_{12}(w_p|D, X)$$

It is easy to know that $\Delta_{12}(p) \in [0, 1]$. Our algorithm decides whether to replace the original word w_p by w_p* with probability $\Delta_{12}(p)$. A larger value of $\Delta_{12}(p)$ indicates that X will be more consistent with the original distribution when w_p is replaced by w_p*. In such a situation, the algorithm will replace w_p with a higher probability.

Our first-stage perturbation is shown in lines 1-6 of Algorithm 1. The randomness in this process increases the difficulty for the attacker to analyze the system.

Step2: Voting

As shown in Table 1, a well-trained neural network can satisfy weak robustness in most perturbation spaces around data distribution. Inspired by the weak robustness property and ensemble learning, we propose to enhance the prediction result of an instance via the voting results of its random perturbations. If we sample X_i uniformly from $\Omega(X)$, the neural network f will make a correct decision ($f(X_i) = y*$) with a probability greater than PR. So we can utilize this statistical property to eliminate the randomness introduced in the first stage and obtain a robustness guarantee. Step2 is presented in Line 7-14 of Algorithm 1.

Two popular voting methods can be utilized in this process: majority voting and plurality voting.

Majority Voting: The final output class label $\widetilde{y}$ is the one that receives more than half of the votes. If none of the class labels receives more than half of the votes, a rejection option will be given and F outputs no prediction. The ensemble prediction is:

$$\widetilde{y} := \begin{cases} y & \text{if } y \text{ takes at least half of votes} \\ Rejection & else \end{cases}$$

Plurality Voting: In contrast to majority voting, which requires the final winner to take at least half of the votes, plurality voting takes the class label which receives the largest number of votes as the final winner. That is, the output class label of the ensemble is:

$$\widetilde{y} := \arg\max_{y \in \mathcal{Y}} \sum_{i=0}^{N-1} \mathbb{I}(f(X_i) = y)$$

For resisting adversarial attack, majority voting is safer than plurality voting. During majority voting, the classifier F will reject inputs located in the input space where the base classifier f does not satisfy weak robustness. However, it comes at the cost of not being able to process all the inputs.

Sample Size Each X_i is a random perturbation version of X. So we regard this step as a second-stage perturbation. To eliminate randomness and obtain robustness guarantee, sample size N is important. The more samples are drawn, the lower the probability that F commits an error. However, it will consume more computation resources and time. So, we need to calculate an appropriate sample size N.

The error rate (including majority voting and plurality voting) of the ensemble with values of robustness metric PR and sample size N is:

$$\begin{aligned} \mathbb{P}(F(X) \neq y^*) &\leq \mathbb{P}(\sum_{y \neq y*} \sum_{i=0}^{N-1} \mathbb{I}(f(X_i) = y) > N/2) \\ &= \sum_{k=0}^{\lfloor N/2 \rfloor} \binom{k}{N} (1 - PR)^{N-k} PR^k \\ &\leq exp(-\frac{1}{2} N(2PR - 1)^2) \end{aligned}$$

For majority voting, '$\leq$' can be rewritten as '$=$' in the first line of the derivation.

If we want the error rate of the voting result to be less than some certain value ϵ, the following inequation must hold:

$$exp(-\frac{1}{2} N(2PR - 1)^2) < \epsilon$$

Then we have:

$$N > \frac{-2 \ln \epsilon}{(2PR - 1)^2}$$

Theorem 1. *If a base classifier f satisfies weak robustness on perturbation space $\Omega_r(X)$ and its robustness metric is PR, then classifier F will output a wrong decision in the voting process with probability less than ϵ when sample size N is greater than $(-2 \ln \epsilon / (2PR - 1)^2)$.*

This result also shows that why weak robustness is defined as $PR > 2/3$. When $PR \to 1/2$, it needs too many samples to give a certified robustness result. When $\epsilon = 10^{-5}$, $PR > 2/3$, we have $N > 207$.

Base Classifier Improvement

We know that the robustness of the classifier F depends on the weak robustness of the base classifier f. Sometimes f does not satisfy weak robustness, or it still has room for improvement. In such a situation, we can use adversarial training to improve the weak robustness of the base classifier f.

Adversarial training is a popular paradigm to improve robustness. It can eliminate some adversarial examples in the perturbation space. We propose to use adversarial training to eliminate adversarial samples in regions that do not meet the weak robustness criterion. It means that we only focus on areas where weak robustness is not satisfied. Unlike previous adversarial training, adversarial examples are carefully selected in our training. We provide two possible methods here:

1) Adding adversarial examples to training set. These adversarial examples are drawn from the area where robustness score PR is less than $2/3$ near the training data.
2) Modifying the training loss function as:

$$\widetilde{\mathcal{L}}(\theta, X, y) := \begin{cases} \mathcal{L}(\theta, X, y) & if\ PR > 2/3 \\ \alpha \mathcal{L}(\theta, X, y) + (1 - \alpha) \mathcal{L}(\theta, X_{adv}, y) & else \end{cases}$$

The adversarial example X_{adv} is generated from the input sentence X while measuring the value of PR. The loss function is denoted as $\mathcal{L}(\cdot)$ and coefficient is used to trade off the loss generated by normal samples and adversarial samples.

These two methods are similar in essence. As the training process is to pursue the global optimization of the loss function, increasing adversarial examples without choice may obscure the impact of important adversarial examples on network training. We make the network more focused on the region that does not meet the weak robustness criterion via these selected adversarial examples. Besides, our adversarial training method does not rely on adversarial attack algorithms, which are not efficient enough especially for attacking large-scale training data.

Experiments

We conduct experiments on two important NLP tasks: text classification and natural language inference. BiLSTM (Conneau et al. 2017) and BERT (Devlin et al. 2019), which represent two popular architectures of deep neural networks, were used to evaluate our robustness enhancement method[1] under two representative attack algorithms.

Data Set MR (Pang and Lee 2005), IMDB (Maas et al. 2011) and SNLI (Bowman et al. 2015) are chosen as data sets. MR and IMDB are classical data sets for sentence-level and document-level sentiment classification respectively. They are binary classification tasks with an average sentence length of 20 and 215 words respectively. Following (Jin et al. 2020), 90% of the MR data is used as the training set and 10% is the test set. SNLI (Bowman et al. 2015) is the data set that is used to learn to judge the relationship between two sentences: whether the second sentence (hypothesis) can be derived from the first sentence (premise)

[1]Code is available at https://github.com/YANG-Yuting/fight-perturbation-with-perturbation

Dataset	Method	LSTM Acc	LSTM Textfooler Suc↓	LSTM Textfooler Rob	LSTM SemPSO Suc↓	LSTM SemPSO Rob	BERT Acc	BERT Textfooler Suc↓	BERT Textfooler Rob	BERT SemPSO Suc↓	BERT SemPSO Rob
MR	f	82.47	69.70	25.00	81.82	15.00	89.60	48.35	47.00	73.63	24.00
	Adv	79.85	65.82	27.00	82.27	14.00	**88.00**	35.91	58.00	73.08	24.50
	FGWS	78.73	56.60	34.50	76.73	18.50	83.88	23.98	65.00	56.14	37.50
	SAFER	77.60	22.08	60.00	27.10	56.50	86.32	**7.30**	**82.00**	13.50	67.50
	F	**81.16**	**14.65**	**67.00**	**25.79**	**59.00**	87.72	8.89	**82.00**	**10.87**	**72.50**
IMDB	f	89.94	86.24	13.00	99.45	0.50	93.68	82.63	16.5	92.51	7.00
	Adv	87.64	71.03	25.50	99.95	0.50	91.00	38.95	58.00	58.42	41.00
	FGWS	85.70	77.84	19.50	92.61	6.50	89.60	62.30	34.50	88.52	40.50
	SAFER	86.60	13.97	77.00	25.28	66.50	88.00	7.07	85.50	-	-
	F	**89.30**	**3.70**	**91.00**	**9.89**	**82.00**	**93.40**	**3.11**	**93.50**	-	-
SNLI	f	84.35	72.05	22.50	50.93	39.50	86.77	69.94	26.00	71.10	25.00
	Adv	**84.35**	75.16	20.00	60.87	31.50	82.53	52.98	39.50	54.17	38.50
	FGWS	72.40	38.06	41.50	37.31	42.00	75.60	44.06	40.00	44.76	39.50
	SAFER	56.60	**19.66**	47.00	**17.24**	48.00	67.00	26.90	53.00	27.08	52.50
	F	80.27	22.22	**59.50**	26.53	**54.00**	**83.90**	**15.34**	**69.00**	**24.84**	**59.00**

Table 2: Robustness evaluation results of different defense methods. Acc is the clean accuracy on test set. Suc is the successful attacking ratio. Rob is the robustness accuracy. Only for Suc, the lower the value, the better the defense capability of the model. It is noted with ↓. The numbers in bold denote the best performance for the metric.

with entailment, contradiction, or neutral relationship. It is a multi-classification task with an average of 8 words and the adversary is only allowed to change the hypothesis.

Models For each task, we choose two widely used and representative models, word-based long-short term memory (BiLSTM) and the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) to do experiments. For BiLSTM, we used a 1-layer bidirectional LSTM with 150 hidden units, and 300-dimensional pre-trained GloVe word embeddings (Pennington, Socher, and Manning 2014). We used the 12-layer based version of BERT model with 768 hidden units and 12 heads, with 110M parameters. All models are trained on cross-entropy loss, and we use Adam (Loshchilov and Hutter 2018) as our optimizer.

Perturbation Set The candidate set for the perturbated position is generated based on HowNet (Dong and Dong 2006) and similarities of word embeddings. We first utilize Stanford POS tagger (Ratnaparkhi 1996) to get part-of-speech and then look up the corresponding synonyms in HowNet, which is arranged by the sememe and can find the potential semantic-preserving words. Then, top η ($\eta = 5$) synonyms are reserved as final candidates for each position ranked according to the cosine similarity of GloVe word embedding space.

Attacking Methods We use two SOTA adversarial attacking methods involving different search paradigms: TextFooler (Jin et al. 2020) and SemPSO (Zang et al. 2020). TextFooler represents SOTA greedy algorithm based on the word importance which is measured as the prediction change before and after deleting the word in a sentence. SemPSO represents SOTA attacks based on evolutionary computation (Wolsey and Nemhauser 1999). It regards word-level attacking as a combinatorial optimization problem and introduces particle swarm optimization-based search algorithm. Compared with genetic algorithms, it has a higher attack success rate. On different tasks and models, they have been verified to have strong attack ability. The adversarial examples with modification rates less than 25% are considered valid. Attacking is conducted on the randomly sampled 200 test data.

Defense Baselines Three recent defense methods are utilized as baselines (Adv, FGWS (Mozes et al. 2021) and SAFER (Ye, Gong, and Liu 2020)). Adv denotes the adversarial training by data augmentation. We retrain the model by original data and the adversarial examples. FGWS is based on identifying adversarial examples via the frequency properties of adversarial word substitutions. It will generate a sequence X' from the input sequence X by replacing all words with their synonyms which have higher occurrence frequencies in the corpus. Then it will decide whether X is an adversarial example via the difference of prediction confidence on class y before and after transformation exceeds the threshold. SAFER is a certified robustness defense method based on randomized smoothing. SAFER will take the corrupted copies of each input sentence as inputs, in which every word of the sentence is randomly replaced with one of its synonyms, then it will make a decision by the difference between the prediction with the highest probability and the second-highest prediction. As our second stage is similar to randomized smoothing, we choose it as the baseline for comparison. Especially, our second stage reduces the number of ensembles to 256 instead of 5000 used in SAFER.

This reduction will greatly improve the efficiency of large models at the inference time.

Setting For our defense method, we set $\lambda = 0.5$, $\kappa = 0.25$ and sample size N is 256. Based on theorem 1, if N is greater than 207, the error rate ϵ is less than 10^{-5}. For a fair comparison, plurality voting is applied in our method as it will not reject an input.

Evaluation Metrics Clean acc (**Acc**) is the prediction accuracy and (**Suc**) is the ratio of successful attacking. Attacks are performed on texts classified correctly. We also utilize robustness accuracy (**Rob**) to estimate the prediction accuracy for a model under attack. It actually can be seen as a comprehensive indicator considering both clean accuracy and successful attacking ratio.

Results of Defense Methods

The experimental results for different defense methods on randomly sampled 200 test data are presented in Table 2 where f is the base classifier and F is our robustness enhancement version. The results of SemPSO on SAFER and F are not shown since the attack time for an instance exceeds 3h. We find that:

- Compared with all the baselines, F achieves the highest robustness accuracy on all three data sets and two different models. It indicates that fighting perturbation with perturbation is a promising method to improve adversarial robustness.
- Our method has a good trade-off between clean accuracy and robustness. F always maintains the accuracy with a decrease of less than 4% and reduces the attack success rate with a ratio larger than 40% compared with the base classifier f. Take the BERT trained on MR task as an instance, adversarial training (Adv) can almost maintain the clean accuracy (88.00% with 1.6% decrease) but is attacked successfully with 73.08% under SemPSO attack. Thus, it achieves a very low robustness accuracy (24.50%). The opposite one, SAFER, which performs well in defending attacks with only 13.50% Suc, ignores the clean accuracy which drops from 89.60% to 86.32%. It also indicates the difficulty of trading off between clean accuracy and robustness accuracy.
- Consistent with our theoretical analysis, weak robustness has a great impact on strengthening the classifier F. Take the BERT-IMDB as an instance, more than 97.61% regions are weakly robust, so its enhancement classifier F will not drop too much in clean accuracy (0.28%) and performs well in robustness accuracy. For LSTM-SNLI, enhancement classifier F drops the most in both clean accuracy and robustness accuracy, which is attributed to its worst performance in weak robustness as shown in Table 1. As the weak robustness of BERT-SNLI (f) is better than LSTM-SNLI (f), BERT-SNLI (F) performs better than LSTM-SNLI (F) in defense attacks. For all tasks, the weak robustness of BERT is better, so the robustness of its enhancement classifier is more satisfactory.
- Under the attacking from the relatively weak attack method (TextFooler) to a stronger one (SemPSO), although the robustness accuracy of F drops, it still performs better than the other three baselines. The reason for decreasing in robustness accuracy is that the perturbation in the first stage cannot conform some specific inputs to the data distribution. There is still room for improvement in the first stage.

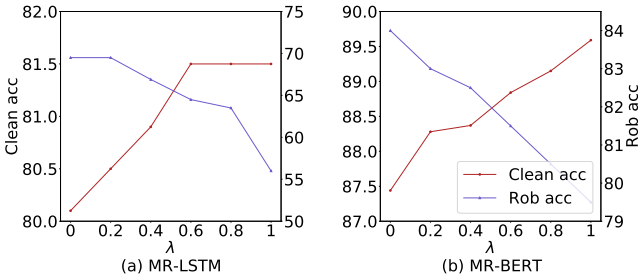

Figure 2: The effect of λ on clean accuracy and robustness of accuracy.x-axis is the value of λ. Left y-axis is the value of clean accuracy and right y-axis is the value of robustness accuracy.

The influence of hyperparameter λ Figure 2 shows the impact of the value of λ on accuracy and robustness. When $\lambda = 0$, the first stage perturbation only considers 1-gram information. When $\lambda = 1$, the first stage perturbation only uses 2-gram information. We can see that, with λ increases, the clean accuracy of the enhancement classifier F increases, but the robustness decreases. So we choose $\lambda = 0.5$ to balance defense performance and clean accuracy.

Influence of Improving Base Classifier

Then, we investigate the effect of our adversarial training method on the robustness improvement. In this experiment, we fix the outer enhancement framework and substitute the inner base classifier f with the other two versions: f**+Adv** (retraining f with traditional adversarial data augmentation) and f**+AdvPR** (retraining f with our adversarial data augmentation). For Adv, the number of adversarial samples added is 25% of the original training set. Our training method AdvPR has two advantages over Adv: (1) As it only adds adversarial examples to areas that have not reached weak robustness, fewer adversarial examples will be added during adversarial training. (2) It does not rely on adversarial attack algorithms and is more efficient. Since two models of IMDB have good weak robustness, we do not retrain their base classifiers.

From the results shown in Table 3, we observe that Adversarial training is helpful for improving weak robustness. In most cases, using the base classifier after adversarial training can obtain a more robust F. Besides, our adversarial training achieves a competitive result with less adversarial examples. AdvPR always achieves a lower value of Suc and a higher value of Rob compared with Adv. It indicates the effect of the adversarial training we proposed, which precisely focuses on the weakly robust parts and improves them.

Dataset	Method	LSTM Acc	LSTM Textfooler Suc↓	LSTM Textfooler Rob	LSTM SemPSO Suc↓	LSTM SemPSO Rob	BERT Acc	BERT Textfooler Suc↓	BERT Textfooler Rob	BERT SemPSO Suc↓	BERT SemPSO Rob
MR	$F(f)$	**81.16**	14.65	67.00	25.79	59.00	87.72	**8.89**	82.00	10.87	72.50
	$F'(f$+Adv)	79.30	16.35	66.50	31.64	54.00	**87.82**	9.84	**82.50**	5.31	68.03
	$F''(f$+AdvPR)	80.70	**14.11**	**70.00**	**25.25**	**62.00**	87.06	9.30	**82.50**	**4.05**	**73.50**
SNLI	$F(f)$	**80.27**	22.22	59.50	26.53	54.00	**83.90**	15.34	69.00	24.84	59.00
	$F'(f$+Adv)	80.16	20.41	58.50	25.85	54.50	82.05	16.27	69.50	20.13	61.50
	$F''(f$+AdvPR)	79.54	**16.89**	**61.50**	**21.05**	**60.00**	81.90	**12.96**	**70.50**	**15.09**	**67.50**

Table 3: Robustness evaluation results of different inner classifiers to be enhanced. f is the base classifier. f+Adv denotes retraining base classifier f with traditional adversarial training. f+AdvPR denotes retraining base classifier f with our adversarial training. These classifiers are enhanced by FPP.

Related work

Adversarial Attack In recent years, synonyms substitution is one of the most popular approaches to attack advanced pre-trained neural models (Ren et al. 2019; Jin et al. 2020; Li et al. 2020; Garg and Ramakrishnan 2020). An attacker deliberately perturbs certain words by their synonyms to mislead the prediction of the target model. At the same time, a high-quality adversarial sample should be imperceptible to humans, which means maintaining grammatical correctness and semantic consistency. Current word-level attacks adopt heuristic algorithms to craft adversarial examples. In each decision step, the algorithm will first pick a vulnerable token to be perturbed, and then choose a suitable synonym to replace them.

Defense Methods For image, adversarial training (Goodfellow, Shlens, and Szegedy 2015; Athalye, Carlini, and Wagner 2018) is widely adopted to mitigate the adversarial effect. However, (Alzantot et al. 2018; Jin et al. 2020) showed that this method has limitations to improve the robustness of NLP model to defend word substitution attack. A retraining model with a limited number of adversarial examples cannot guarantee to eliminate all adversarial examples in the perturbation space. Besides, synonym substitution-based attack methods are usually much less efficient to be incorporated into adversarial training or generate a large number of adversarial examples for large pre-trained models.

Some verification methods like Interval Bound Propagation (IBP), originally proposed for images (Gowal et al. 2019), have been introduced to certify and improve robustness against adversarial word substitution (Jia et al. 2019; Huang et al. 2019). (Shi et al. 2020) and (Xu et al. 2020) proposed the robustness verification and training method for transformers based on linear relaxation-based perturbation analysis. Although these over-approximate methods have rigorous soundness guarantees, they often lead to loose upper bounds for arbitrary networks and result in a higher cost of clean accuracy. Furthermore, due to the computational difficulty of verification, certified defense methods are usually not scalable to large and deep neural networks. To scale up to large models, (Ye, Gong, and Liu 2020) proposed a certified robust method called SAFER which is inspired by randomized smoothing (Cohen, Rosenfeld, and Kolter 2019). Although it can provide a rigorous robustness guarantee on some input space, as shown in our experiments, there is still much room for improvement in resisting attacks. Randomized smoothing methods are promising but very much underexplored in NLP field. One reason is that word substitutions are discrete and the perturbation is not defined on ℓ_2 norm.

In recent years, some empirical defense methods like FGWS (frequency-guided word substitution) (Mozes et al. 2021) and DISP (learning to discriminate perturbations) (Zhou et al. 2019) have been proposed. FGWS exploits the frequency properties of adversarial word substitutions for the detection of adversarial examples. DISP learns a perturbation discriminator to identify malicious perturbations and block adversarial attacks. However, black-box attack algorithms can still successfully break the defense when the model and detector are considered as a whole.

Conclusion

In this paper, we investigate the possibility to enhance the robustness of deep NLP models via fighting perturbation with perturbation, which is inspired by fighting fire with fire. We first implement our idea to improve the word-level robustness of the models. Our methods only have two steps. The first is to perturb input sequence via 1-gram and 2-gram frequency. The second step is to randomly perturb the sequence and vote for an ensemble result. Our method is structure-free, scalable and simple, which can be incorporated with little effort in various deep models as long as they have weak robustness. Extensive experiments demonstrate that our method can enhance the robustness without sacrificing their performance too much on clean data. Besides, our defense method has good interpretability. Overall, we show that using perturbation to resist perturbation is a promising framework for robustness enhancement. We will transfer this idea to defend other types of attacking for NLP models in future work.

Acknowledgments

This work is supported by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences (Grant No. QYZDJ-SSW-JSC036), National Natural Science Foundation of China (NSFC) under grant No.61972384. Feifei Ma

is also supported by the Youth Innovation Promotion Association CAS under grant No. Y202034. The authors would like to thank the anonymous reviewers for their comments and suggestions.

References

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.; Srivastava, M. B.; and Chang, K. 2018. Generating Natural Language Adversarial Examples. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing EMNLP 18*, 2890–2896. Association for Computational Linguistics.

Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 274–283.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, 632–642.

Cheng, M.; Wei, W.; and Hsieh, C. 2019. Evaluating and Enhancing the Robustness of Dialogue Systems: A Case Study on a Negotiation Agent. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, 3325–3335. Association for Computational Linguistics.

Cheng, M.; Yi, J.; Chen, P.; Zhang, H.; and Hsieh, C. 2020. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 3601–3608. AAAI Press.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, 1310–1320. PMLR.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, 670–680.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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, NAACL-HLT 2019*, 4171–4186.

Dong, Z.; and Dong, Q. 2006. HowNet and the Computation of Meaning. World Scientific.

Garg, S.; and Ramakrishnan, G. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, 6174–6181. Association for Computational Linguistics.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015*.

Gowal, S.; Dvijotham, K.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Arandjelovic, R.; Mann, T. A.; and Kohli, P. 2019. Scalable Verified Training for Provably Robust Image Classification. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*, 4841–4850.

Huang, P.; Stanforth, R.; Welbl, J.; Dyer, C.; Yogatama, D.; Gowal, S.; Dvijotham, K.; and Kohli, P. 2019. Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *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*, 4081–4091. Association for Computational Linguistics.

Islam, A.; and Inkpen, D. 2009. Real-Word Spelling Correction using Google Web 1T 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, 1241–1249. ACL.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, 2021–2031. Association for Computational Linguistics.

Jia, R.; Raghunathan, A.; Göksel, K.; and Liang, P. 2019. Certified Robustness to Adversarial Word Substitutions. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *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*, 4127–4140. Association for Computational Linguistics.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 8018–8025.

Li, L.; Ma, R.; Guo, Q.; Xue, X.; and Qiu, X. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, 6193–6202.

Liang, B.; Li, H.; Su, M.; Bian, P.; Li, X.; and Shi, W. 2018. Deep Text Classification Can be Fooled. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, 4208–4215. ijcai.org.

Loshchilov, I.; and Hutter, F. 2018. Fixing weight decay regularization in adam.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 142–150.

Mozes, M.; Stenetorp, P.; Kleinberg, B.; and Griffin, L. D. 2021. Frequency-Guided Word Substitutions for Detecting Textual Adversarial Examples. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*, 171–186. Association for Computational Linguistics.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics*, 115–124.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 1532–1543.

Pruthi, D.; Dhingra, B.; and Lipton, Z. C. 2019. Combating Adversarial Misspellings with Robust Word Recognition. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, 5582–5591. Association for Computational Linguistics.

Raghunathan, A.; Xie, S. M.; Yang, F.; Duchi, J. C.; and Liang, P. 2020. Understanding and Mitigating the Tradeoff between Robustness and Accuracy. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, 7909–7919. PMLR.

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *Conference on empirical methods in natural language processing*.

Ren, S.; Deng, Y.; He, K.; and Che, W. 2019. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, 1085–1097. Association for Computational Linguistics.

Sakaguchi, K.; Post, M.; and Durme, B. V. 2017. Grammatical Error Correction with Neural Reinforcement Learning. In Kondrak, G.; and Watanabe, T., eds., *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017*, 366–372. Asian Federation of Natural Language Processing.

Samanta, S.; and Mehta, S. 2017. Towards Crafting Text Adversarial Samples. *CoRR*, abs/1707.02812.

Shi, Z.; Zhang, H.; Chang, K.; Huang, M.; and Hsieh, C. 2020. Robustness Verification for Transformers. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Wang, X.; Yang, Y.; Deng, Y.; and He, K. 2021. Adversarial Training with Fast Gradient Projection Method against Synonym Substitution Based Text Attacks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, 13997–14005. AAAI Press.

Wolsey, L. A.; and Nemhauser, G. L. 1999. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons.

Xu, K.; Shi, Z.; Zhang, H.; Wang, Y.; Chang, K.; Huang, M.; Kailkhura, B.; Lin, X.; and Hsieh, C. 2020. Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Ye, M.; Gong, C.; and Liu, Q. 2020. SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 3465–3475. Association for Computational Linguistics.

Zang, Y.; Qi, F.; Yang, C.; Liu, Z.; Zhang, M.; Liu, Q.; and Sun, M. 2020. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6066–6080.

Zhao, Z.; Dua, D.; and Singh, S. 2018. Generating Natural Adversarial Examples. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net.

Zheng, X.; Zeng, J.; Zhou, Y.; Hsieh, C.; Cheng, M.; and Huang, X. 2020. Evaluating and Enhancing the Robustness of Neural Network-based Dependency Parsing Models with Adversarial Examples. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, 6600–6610. Association for Computational Linguistics.

Zhou, Y.; Jiang, J.; Chang, K.; and Wang, W. 2019. Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *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*, 4903–4912.