PAD: A Robustness Enhancement Ensemble Method via Promoting Attention Diversity

Yuting Yang*, Pei Huang, Feifei Ma, Juan Cao and Jintao Li
Institute of Computing Technology, Chinese Academy of Sciences (CAS)
University of Chinese Academy of Sciences
Department of Computer Science, Stanford University
State Key Laboratory of Computer Science, Institute of Software, CAS
Laboratory of Parallel Software and Computational Science, Institute of Software, CAS
{yangyuting, caojuan, jtli}@ict.ac.cn, huangpei@stanford.edu, maff@ios.ac.cn

Abstract

Deep neural networks can be vulnerable to adversarial attacks, even for the mainstream Transformer-based models. Although several robustness enhancement approaches have been proposed, they usually focus on some certain type of perturbation. As the types of attack can be various and unpredictable in practical scenarios, a general and strong defense method is urgently in require. We notice that most well-trained models can be weakly robust in the perturbation space, i.e., only a small ratio of adversarial examples exist. Inspired by the weak robust property, this paper presents a novel ensemble method for enhancing robustness. We propose a lightweight framework PAD to save computational resources in realizing an ensemble. Instead of training multiple models, a plugin module is designed to perturb the parameters of a base model which can achieve the effect of multiple models. Then, to diversify adversarial example distributions among different models, we promote each model to have different attention patterns via optimizing a diversity measure we defined. Experiments on various widely-used datasets and target models show that PAD can consistently improve the defense ability against many types of adversarial attacks while maintaining accuracy on clean data. Besides, PAD also presents good interpretability via visualizing diverse attention patterns.

Keywords: adversarial robustness, pre-trained language models, attention

1. Introduction

Deep neural networks (DNNs) have been broadly applied in various natural language processing (NLP) tasks. However, they are vulnerable to adversarial examples that are intentionally crafted by attackers for misleading the predictions of models with few perturbations, ranging from character-level, word-level to sentence-level. Character-level attacks (Li et al., 2019; Gao et al., 2018; Pruthi et al., 2019) usually insert, replace or delete some characters in the inputs. Most word-level attacks (Jin et al., 2020; Zang et al., 2020; Ren et al., 2019a) search adversarial examples in the synonym spaces with different search algorithms. Recently, pre-trained language models (PLMs) are also utilized to generate candidate substitutions (Yang et al., 2022a; Li et al., 2020). Sentence-level attacks manipulate new paraphrases for sentences, e.g., Iyyer et al. (2018) transform sentences with the desired syntax. With the emergence of more and more adversarial attacks, general and efficient methods for defending against adversarial attacks and enhancing robustness are of critical importance for developing trustworthy AI systems.

As a countermeasure, a series of defense methods are proposed targeted at certain specific attacks. Character-level perturbations can be corrected via predicting the original words using context semantics (Pruthi et al., 2019). For word substitution-based attacks, Zhou et al. (2019) train a perturbation discriminator to identify malicious perturbations. Recently, few work begins to explore the feasibility of general defense methods. Although adversarial training (Wang et al., 2021; Liu et al., 2020; Alzantot et al., 2018) is a general defense method, it relies on the knowledge about target attacks and is time-consuming owing to the process of generating enough adversarial examples. Considering that many attacks rely on iterative search mechanisms, Le et al. (2022) proposes to confuse the attackers by automatically weighted ensembles of several classification layers. However, the frozen feature extraction module limits the diversity of models and the randomness of the ensemble brings instability to the outputs.

For well-trained DNNs, adversarial examples occupy a small ratio in most perturbation spaces (Yang et al., 2022b). In this case, the ensemble method can have a large probability of defending attack if the adversarial examples of different submodels distribute in different regions. As the left part of Fig. 1 shows, the majority of sub-models make correct predictions for the input x, then the vote result of ensemble is also correct. Inspired by this, we want to develop an ensemble method to enhance robustness for Transformer-based models, which are the most popular architectures recently

*Corresponding author.

20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

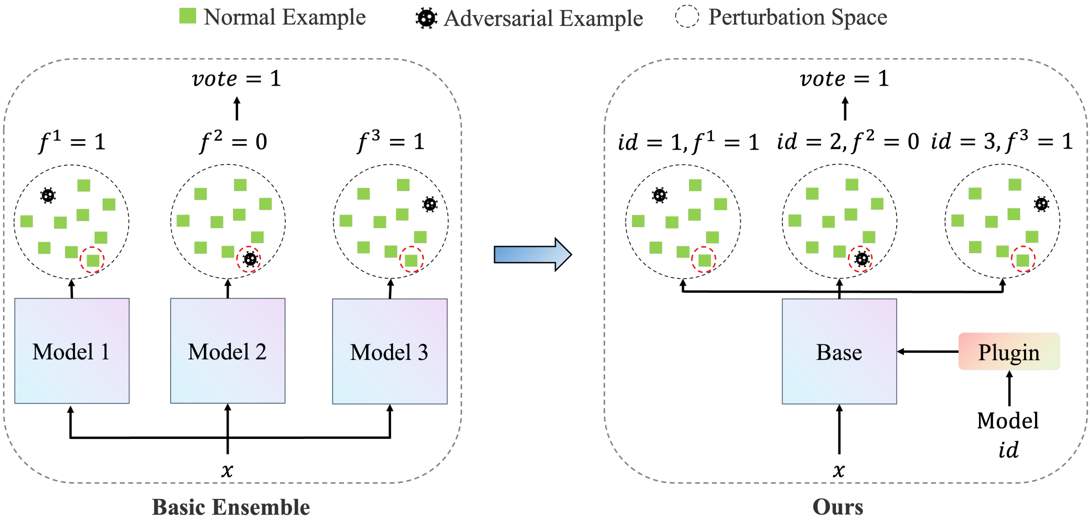

Figure 1: Illustration of ensemble method. The ensemble method can enhance robustness via diversifying the adversarial example distribution in the perturbation spaces. A lightweight plugin is designed to perturb the parameters of the base model.

due to their outstanding performances. Utilizing ensemble method to enhance robustness faces two main challenges: (1) How to train several sub-models with low computational resources, especially for large PLM-based models? (2) How to diversify the adversarial example distributions for different sub-models?

In order to diversify adversarial example distributions with as low computational resources as possible, we propose a novel ensemble method based on **p**romoting **a**ttention **d**iversity (PAD). To tackle the computational challenge, we design a lightweight plugin to learn the perturbation of parameters since sub-models generally share the same architectures but differ in the values of parameters as the right part of Fig. 1 shows. Instead of perturbing all parameters, PAD only perturbs some parameters of the first self-attention layer as attention module is one of the most important parts for feature extraction in Transformer-based models. Perturbing first-layer attention also provides good interpretability as they correspond to the attention on original input tokens.

Although different sub-models can be perturbed with different parameters at first, their parameters tend to be very similar while achieving coverage state as they share the same architectures and optimization goals. Intuitively, a true prediction can be made based on different attention patterns. Take a text (*"I like this movie, it is really nice!"*) in sentiment classification task as an example. If a model pays more attention to (*"like"*,*"movie"*) and the other two models pay more attention to (*"movie", "nice"*), all of them can output positive classification results. Thus, the perturbation on *"like"* will be alleviated in the ensemble. To diversify the attention of different models, we first define a measure for attention diversity based on the differences in attention score vectors and then optimize it in the training process.

We evaluate our method on experiments of three NLP tasks and two target models (BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)). Experimental results show that PAD significantly improves the robustness under different levels of attacks with an average increase of 9.2% robustness accuracy over the state-of-the-art defense baselines. Besides, PAD can maintain the performance on normal examples well. Via visualizing attention scores in different sub-models, our work also presents good interpretability which explains how the different sub-models make complementary decisions.

2. Preliminary

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 and output space $\mathcal{Y} = \{y_1, y_2, ..., y_c\}$ contains c possible predictions of an input.

(Adversarial Example) Consider a classifier $f(x)$. Given a sequence x with gold label y^* and x' is a text generated by perturbing x with semantic preservation, x' is an adversarial example if:

$$f(x') \neq y^*. \tag{1}$$

(Perturbation Space) A perturbation space $\Omega(x)$ of an input sequence x is a set containing all perturbations x' generated by perturbing the original input with semantic preservation.

(Weak Robustness) If the value of $\mathbb{PR} > 1/2$, f is said to be weak robust on the perturbation space $\Omega(x)$, where $\mathbb{PR}$ is the robustness metric:

$$\mathbb{PR} := \frac{|\{x' : x' \in \Omega(x) \wedge f(x') = y^*\}|}{|\Omega(x)|}. \qquad (2)$$

$(1 - \mathbb{PR})$ measures the proportion of adversarial examples in the perturbation space, i.e., the probability of being altered by a random perturbation. Existing work observes that most well-trained NLP neural models satisfy weak robustness, e.g., BERT trained on IMDB dataset achieves $\mathbb{PR}$ larger than 0.9 in 90.66% word substitution spaces (Yang et al., 2022b). As perturbation space $\Omega(x)$ is difficult to be formally defined for other types of perturbations, we can not analyze the value of $\mathbb{PR}$. In this paper, we assume that well-trained DNNs can satisfy weak robustness for other types of perturbation, which is the prerequisite of our method.

3. Methodology

Ensemble learning is a generic approach to aggregating weak models into strong models and is usually effective to improve generalization performance. In this paper, we extend it to enhance weak robustness. Assuming that we have M well-trained deep models$\{f^1, ..., f^M\}$, they are weakly robust on the perturbation space and their adversarial example distributions are different. A basic idea for reducing the number of adversarial examples is aggregating the predictions of individual M models, e.g., plurality voting. The prediction result $\tilde{y}$ of the ensemble is:

$$\tilde{y} := \arg\max_{y \in \mathcal{Y}} \sum_{m \in [M]} \mathbb{I}(f^m(x) = y), \qquad (3)$$

where f^m is m-th sub-model in the ensemble.

There still exists two issues that need to be addressed:

1 Modern popular Transformer-based models like BERT and RoBERTa are big with millions of parameters. Training several models can be very expensive which sometimes will be infeasible in terms of memory and time complexity.

2 Using the same training data and similar gradient-based training algorithms to train models with the same structure will result in homogeneous models. How to make the distribution of adversarial samples in the perturbation space of M models diverse?

To reduce computational resources, we propose to perturb the weights of the original network with an additional module, which is more like a plugin for a model. The models perturbed under different perturbations generated by the plugin can be seen as different models. To promote model diversity, we propose attention diversity which encourages different models to focus on different parts of an input sequence to make predictions.

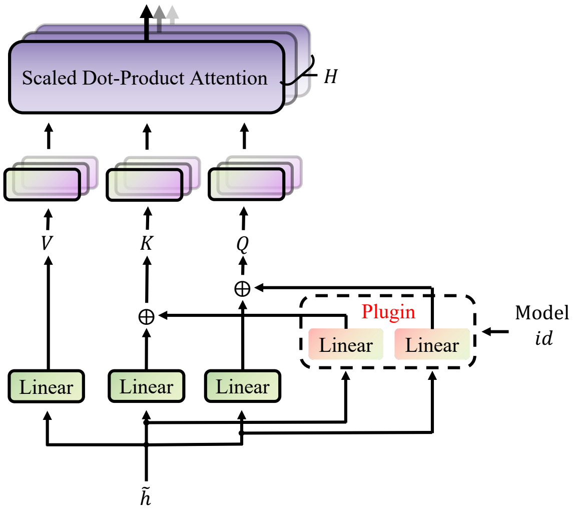

Figure 2: Diagram of the attention plugin module.

3.1. Plugin Module

In this section, we introduce how to construct the ensemble with a lightweight plugin module. Notice that if M models $\{f^1, ..., f^M\}$ in an ensemble have the same structure, then the main difference among these M members is their weights. So we can use small neural networks to generate perturbations for weights and produce different models. Formally, if f is a well-trained base model and θ is its parameters, then we can produce f^m via modifying parameters θ of f as

$$\theta^m = \theta + \Delta^m, \qquad (4)$$

where Δ^m can be generated by a small network.

In pursuit of efficiency and good interpretability, instead of perturbing all the parameters in the base model f, we only perturb some parameters of the first self-attention layer, since the attention module is one of the most important parts for feature extraction in modern NLP models. Besides, the first attention layer is close to the original input sequence and the attention score can be directly interpreted as the attention to the input sequence, which can provide better interpretability.

The query matrix Q and key matrix K of a self-attention module are usually transformed from a matrix $\tilde{h}$ (hidden state or embedding) with linear layers.

$$Q = Linear_Q(\tilde{h}) = W_Q\tilde{h} + b_Q, \qquad (5)$$

$$K = Linear_K(\tilde{h}) = W_K\tilde{h} + b_K. \qquad (6)$$

We can generate f^m via perturbing parameters W_Q, b_Q, W_K and b_K. The query matrix Q_m and key matrix K_m of m-th member f^m can be obtained under perturbation Δ^m:

$$\begin{aligned} Q^m &= (W_Q + \Delta_W^m)\tilde{h} + (b_Q + \Delta_b^m), \\ &= (W_Q\tilde{h} + b_Q) + (\Delta_W^m\tilde{h} + \Delta_b^m), \\ &= Linear_Q(\tilde{h}) + Linear_Q^m(\tilde{h}). \end{aligned} \tag{7}$$

Similarly, we have:

$$K^m = Linear_K(\tilde{h}) + Linear_K^m(\tilde{h}). \tag{8}$$

Thus, the parameters perturbation Δ^m for f^m can be realized with additional linear layers $Linear_Q^m(h)$ and $Linear_K^m(h)$. Fig. 2 is a diagram. For an input sequence x, it will be copied in M copies and concatenated with an index (id) indicating which model to feed into. For example, an input $< x, m >$ indicates that the plugin will generate perturbations Δ^m and constructs f^m to deal with the sequence x.

Suppose $\mathcal{L}_{CE}^m$ is the classification loss of m-th member (e.g., cross-entropy loss), then one of the optimization objectives of the ensemble can be defined as:

$$\mathcal{L}_{CE} = \frac{1}{M}\sum_{m\in[M]} \mathcal{L}_{CE}^m. \tag{9}$$

3.2. Attention Diversity

If we only use $\mathcal{L}_{CE}$ to train the ensemble, the parameters of different sub-models will tend to be very similar as they share the same architecture and optimization goal. In this section, we define a new loss to encourage the diversities among sub-models.

The self-attention mechanism allows the inputs to interact with each other ("self") and find out to who they should pay more attention ("attention"). Sometimes the same prediction result can be given based on different attention patterns. For example, for a sentence *"I like this movie, it is really nice!"*, if a model pay more attention to (*"like","movie"*) and the other one pay more attention to (*"movie","nice"*), They both output positive sentiment classification results. Another case is that if both models have the highest attention on (*"like", "movie"*) but with different values, they can also output the same classification result. Thus, promoting attention diversity can not only maintain accuracy but also improve the diversity of different sub-models. Further, it can affect the distributions of adversarial examples for each model.

Suppose $A^m \in R^{H\times L\times L}$ represent the attention scores outputted by a self-attention layer of m-th model. H is the number of attention heads and L is the sequence length. Then the sum score of all heads can be denoted as $\hat{A}^m \in R^{L\times L}$:

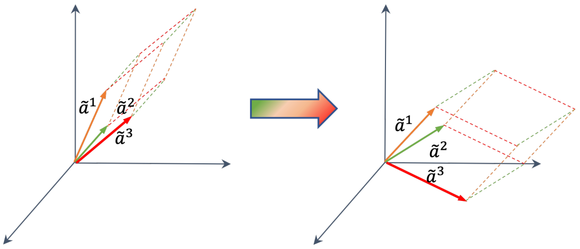

Figure 3: Diagram of promoting attention diversity.

$$\hat{A}^m = \sum_{h\in[H]} A_h^m, \tag{10}$$

which can be regarded as the attention scores between any two token pairs. $\hat{A}^m = (\hat{a}_1^m, ..., \hat{a}_L^m)^\top$ with each row $\hat{a}_l^m$ is the attention scores for l-th token position. We use $\tilde{A}^m = (\tilde{a}_1^m, ..., \tilde{a}_L^m)^\top$ to denote the matrix that each row $\tilde{a}_l^m$ is obtained by normalizing $\hat{a}_l^m$ in $\hat{A}^m$ under ℓ_2-$norm$, i.e., $\|\tilde{a}_l^m\|_2 = 1$.

Let D_l ($l \in [L]$) be the gather of the normalized attention scores of l-th token position for M models. It is denoted as:

$$D_l = (\tilde{a}_l^1, ..., \tilde{a}_l^m)^\top. \tag{11}$$

Based on the matrix theory (Bernstein, 2009), the determinant of matrix computes the volume spanned by the vectors. Then we use the determinant of matrix $D_lD_l^\top$ to measure the diversity among these attention vectors.

$$det(D_lD_l^\top) = Vol^2(\tilde{a}_l^1, ..., \tilde{a}_l^M), \tag{12}$$

where $Vol(\cdot)$ is the volume of the geometry formed by vectors $(\tilde{a}_l^1, ..., \tilde{a}_l^M)$. Intuitively, higher values of $det(D_lD_l^\top)$ indicate higher attention diversity as shown in Fig. 3.

In pursuit of diversities among models, we can minimize the attention diversity loss:

$$\mathcal{L}_{PAD} = -\frac{1}{L}\sum_{l\in[L]} det(D_lD_l^\top). \tag{13}$$

To maintain accuracy and promote attention diversity, the minimization goal in the training process can be denoted as:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{PAD}, \tag{14}$$

where λ is a hyper-parameter for regulating the effect of the impact of attention diversity. While training, the original parameters of base models are frozen.

			Character				Word				Sentence	
Dataset	Method		VIPER		DeepWordBug		TextFooler		BertAttack		SCPN	
		Acc	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob
MR	Base	89.97	38.56	52.50	32.24	61.80	63.82	33.00	71.05	26.40	37.49	53.10
	ADV	89.78	39.68	51.00	34.95	59.20	53.19	42.60	65.05	31.80	39.90	51.00
	ASCC	89.87	38.16	53.00	21.88	71.40	**21.01**	72.20	56.67	39.60	37.09	53.60
	DNE	88.66	38.13	52.50	29.13	65.20	41.30	54.00	68.15	28.60	37.43	53.00
	SHIELD	89.97	36.25	53.50	18.08	75.20	29.54	64.40	54.92	40.20	37.45	53.20
	PAD	**89.97**	**35.42**	**55.50**	**9.73**	**83.50**	21.62	**72.50**	**21.93**	**73.00**	**36.19**	**57.50**
IMDB	Base	88.81	26.48	66.00	35.16	59.00	86.81	12.00	90.11	9.00	60.44	36.00
	ADV	88.70	**22.22**	69.00	16.57	73.00	56.57	38.00	74.44	23.00	63.33	32.00
	ASCC	87.72	28.53	64.50	15.82	81.50	48.96	53.50	79.01	18.00	69.77	26.00
	DNE	86.84	28.89	63.00	16.49	81.00	55.67	43.00	84.54	15.00	68.16	26.00
	SHIELD	88.81	24.61	66.00	17.14	72.50	36.00	56.00	69.66	27.00	59.14	36.50
	PAD	**88.81**	23.19	**69.00**	**8.79**	**83.00**	**30.77**	**63.00**	**37.36**	**57.00**	**58.64**	**38.00**
SNLI	Base	88.36	38.16	51.20	45.52	47.40	65.52	30.00	78.16	19.00	19.89	68.40
	ADV	85.03	39.55	50.20	43.93	49.60	53.28	33.10	77.45	20.30	23.08	61.40
	ASCC	82.98	38.50	47.20	31.52	57.80	56.75	34.60	75.25	19.80	21.75	60.60
	DNE	87.55	40.00	51.60	41.15	51.20	55.40	38.80	77.24	19.80	19.89	67.40
	SHIELD	88.34	37.25	52.30	40.78	51.40	53.46	40.40	71.43	24.80	19.49	68.20
	PAD	**88.36**	**36.05**	**53.60**	**8.51**	**79.60**	**37.47**	**54.40**	**51.03**	**42.60**	**17.13**	**69.80**

Table 1: Robustness evaluation results (%) of BERT-based target models. 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.

4. Experimental Setup

4.1. Datasets and Models

We conduct experiments on two important NLP tasks: text classification and natural language inference. For text classification, MR (Pang and Lee, 2005) and IMDB (Maas et al., 2011) are sentence-level and document-level classification tasks with two classes (positive and negative) respectively. SNLI (Bowman et al., 2015) is the dataset for the natural language inference task: whether the second sentence (hypothesis) can be derived from the first sentence (premise) with entailment, contradiction, or neutral relationship.

Target models include two popular deep neural architectures based on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). BERT and RoBERTa are both the base versions with 12 layers, 768 hidden units, 12 heads, and 110M parameters.

4.2. Attacking Methods

We choose five adversarial methods for different levels of attacks. VIPER (Eger et al., 2019) modifies characters with visual-similar ones, e.g., replacing *"wiki"* with *"w!k!"*. DeepWordBug (Gao et al., 2018) performs small character-level perturbation targeting the important tokens in inputs. For word-level perturbation, TextFooler (Jin et al., 2020) is a greedy algorithm based on the word importance which is measured as the prediction change before and after deleting the word in a sentence. BertAttack (Li et al., 2020) uses the masked language model (BERT) as a perturbation generator and finds perturbations that maximize the risk of making wrong predictions, which extends the perturbation space beyond synonyms. For sentence-level perturbation, we utilize SCPN (Iyyer et al., 2018) which transforms sentences with the desired syntax. All attack algorithms are implemented in the open framework *OpenAttack* (Zeng et al., 2021)[1]. To keep the semantics consistent after attacking, we limit the ratio of perturbation to be less than 0.25 and the similarity of sentence embedding to be larger than 0.5.

4.3. Defense Baselines

Four strong baselines for robustness enhancement are selected: ADV (adversarial training) (Wang et al., 2021; Ren et al., 2019b) usually retrains the model with adversarial examples. ASCC (Dong et al., 2021) models the word substitution space as a convex hull and optimizes adversaries generated in the hull. DNE (Zhou et al., 2021) expands convex hulls to two-hop synonyms neighbors and performs prediction via weighted average of the softmax probability vectors of all the randomly sampled sentences in the convex hull. SHILED (Le et al., 2022) is a general defense method which modifies the classification layer of a trained NN and conducts a stochastic weighted ensemble of different prediction heads.

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

4.4. Metrics and Settings

Three metrics are used for evaluating enhancement methods: clean accuracy (**Acc**), successful rate (**Suc**) and robustness accuracy (**Rob**). Acc is the the prediction accuracy of the original test data. Suc is the attack success rate. Rob measures the accuracy of a model under attack, which is the ratio of inputs that are correctly classified and not successfully attacked.

The weight of attention diversity λ in Eq. 3.2 is set to 0.1 which can make two losses on the same magnitude and make ensemble perform the best results. The number of sub-models is set to 3. PAD has only 1.08% parameters compared with training 3 models for ensemble. [2]

5. Experimental Results

5.1. Robustness Enhancement

We evaluate the performances of different enhancement methods on 500 data randomly sampled from test set and the experimental results are presented in Table 1 and 2 for BERT-based and RoBERTa-based models respectively. As generating adversarial examples is time-consuming, we only generate adversarial examples for 25% randomly sampled training data. Besides, a large number of adversarial examples will shift the original data distribution. ADV in Table 1 and 2 is conducted with word-level adversarial examples as a reference.

Compared with all enhancement baselines, PAD achieves the highest robustness (Rob) for all adversarial attacks and target models. The improvements are significant on DeepWordBug, TextFooler and BertAttack (more than 6% in Rob over the state-of-the-art baselines). As sub-models of PAD pay different attention to input tokens, some perturbation on tokens can be ignored in the ensemble. For the character-level DeepWordBug, e.g., replace *"movie"* with *"moive"*, the embedding vector is split and paid attention to by different heads. Thus, such perturbation is hard to deceive all attention patterns of the sub-models.

PAD achieves relatively moderate improvements on the two hard cases (VIPER and SCPN). Since VIPER modifies original characters with visually-similar ones, e.g., replace *"1"* with *"l"*, it sometimes generates examples out of the original data distribution. SCPN is relatively hard to be defended as it changes the structures of input sentences largely. Thus, sub-models can not learn them from the original data distribution and the ensemble will be ineffective.

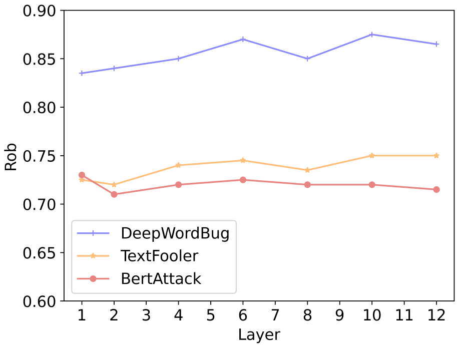

Figure 4: The robustness while performing perturbation on different attention layers of MR-BERT.

Compared with the defense methods designed for some target attacks, PAD performs consistent improvements under all types of attacks. ASCC and DNE aim at defending word-substitution perturbation and sometimes even decrease defense ability for other types of attacks, e.g., ASCC decreases 10% Rob of IMDB-BERT under the sentence-level attack SCPN. PAD also performs a good trade-off between clean accuracy and robustness, which always keeps the same Acc as the base model. Most existing methods improve robustness with the loss of accuracy including ADV, ASCC and DNE (even 5.38% decrease for ASCC on SNLI-BERT). Compared with the general defense method SHIELD, PAD diversifies the feature extraction process while SHIELD freezes it. Some adversarial examples can be caused by the feature extraction module which maps them to inappropriate embedding spaces. So PAD achieves better performance than SHIELD. The good performances of PAD indicate that the general robustness enhancement method is promising, especially under the trend of utilizing unified Transformer-based architectures in NLP domain.

Further, we compare PAD with ADV which retrains the base model with different levels of adversarial examples. Experiment results on MR-BERT are presented in Table 3. ADV$_c$, ADV$_w$ and ADV$_s$ means retraining the base model with character-level, word-level and sentence-level adversarial examples respectively. ADV$_a$ utilizes these three levels of adversarial examples. ADV with adversarial examples of target attack can enhance the robustness against the attack, but becomes ineffective while encountering some other levels of attacks and even decreases the robustness (ADV$_w$ decreases 1.5% Rob under VIPER, VP, compared with the base model). ADV$_a$ stimulates the situation that simultaneously uses methods that resist different types of attacks. We find that ADV$_a$ can improve robustness for all levels of attacks compared with the base model. However, the improvement is lower

[2] Code is available at https://github.com/YANG-Yuting/PAD

Dataset	Method		Character				Word				Sentence	
			VIPER		DeepWordBug		TextFooler		BertAttack		SCPN	
		Acc	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob	Suc↓	Rob
MR	Base	90.07	22.78	67.50	32.17	62.00	65.65	31.40	66.67	30.00	31.67	59.50
	ADV	89.03	27.48	62.60	30.73	62.50	49.71	44.00	48.36	47.80	32.47	58.30
	ASCC	90.01	25.00	65.50	29.44	63.50	30.56	62.50	33.33	60.00	31.11	60.00
	DNE	90.03	27.16	63.73	28.11	65.00	29.83	61.50	30.15	57.01	31.60	59.50
	SHIELD	90.07	21.74	68.00	20.42	70.50	29.11	62.50	31.14	60.50	30.31	60.50
	PAD	**90.07**	**21.67**	**68.50**	**11.67**	**79.50**	**27.22**	**65.50**	**31.01**	**62.00**	**29.44**	**61.50**
IMDB	Base	93.90	28.89	63.00	16.20	75.00	84.92	13.50	93.55	6.00	28.28	66.00
	ADV	92.07	28.26	62.10	15.83	76.00	58.75	35.20	78.65	21.30	27.20	66.30
	ASCC	92.83	26.11	65.40	13.92	77.50	55.36	38.50	75.37	24.70	25.03	67.00
	DNE	93.05	27.93	63.60	15.83	75.50	53.27	36.50	76.54	25.20	27.05	66.10
	SHIELD	93.88	23.42	69.50	10.73	80.00	56.23	38.50	63.73	32.60	24.87	67.20
	PAD	**93.90**	**20.75**	**73.00**	**5.56**	**85.00**	**33.33**	**60.00**	**54.44**	**41.00**	**23.33**	**68.50**
SNLI	Base	86.34	30.30	63.10	38.62	53.40	61.61	33.40	78.85	18.40	35.27	59.20
	ADV	85.75	31.32	62.40	36.88	56.70	49.73	38.90	75.19	21.00	35.30	58.60
	ASCC	85.33	29.37	64.00	35.83	57.80	48.52	38.60	70.36	27.30	34.29	60.20
	DNE	85.59	29.29	64.20	36.27	58.70	49.01	39.80	72.84	25.10	35.10	62.20
	SHIELD	86.34	27.29	66.70	18.74	76.50	50.41	40.70	68.18	28.20	32.24	62.10
	PAD	**86.34**	**25.02**	**68.50**	**5.99**	**81.60**	**40.32**	**51.80**	**55.53**	**38.60**	**28.39**	**66.30**

Table 2: Robustness evaluation results of RoBERTa-based target models.

Method		Character		Word		Sentence
		VP	DW	TF	BA	SC
	Acc	Rob	Rob	Rob	Rob	Rob
Base	89.97	52.50	61.80	33.00	26.40	53.10
ADV$_c$	89.78	53.50	68.00	41.50	30.50	49.50
ADV$_w$	89.78	51.00	59.20	42.60	31.80	51.00
ADV$_s$	89.69	52.00	59.00	30.00	29.50	56.00
ADV$_a$	89.69	53.00	67.00	42.50	33.50	54.00
PAD	89.97	55.50	83.50	72.50	73.00	57.50

Table 3: Robustness of MR-BERT with various adversarial training strategies.

than the ADV trained with the specific level of attack. For example, ADV$_a$ achieves 67.00% Rob under DW while ADV$_c$ achieves 68.00%. It indicates that there may exist conflicts among the fitting of different levels of attacks for ADV.

Number of Sub-models In order to explore the effect of more sub-models for ensemble, we also conduct experiments with five sub-models. The ensemble method improves Rob with an average value of 2%, implying the phenomenon of diminishing marginal returns. As DNNs are no longer weak classifiers, it is difficult to further improve the generalization to some examples out of the distribution.

Perturb Different Attention Layers Transformer-based models always contain multiple self-attention layers. Base versions utilized in this paper have $N = 12$ layers. Some researches observe that BERT (Jawahar et al., 2019) captures surface features at the bottom layer, syntactic features in the middle and semantic features at the top, following the classical tree-like learning structures. We explore the difference in perturbing attention of different layers. Fig. 4 shows the robustness of enhanced MR-BERT for several attacks (DeepWordBug, TextFooler and BertAttack). PAD presents consistent robustness performance on different layers: $85.50\% \pm 1.41\%$ under DeepWordBug, $73.78\% \pm 1.09\%$ under TextFooler and $72.00\% \pm 0.59\%$ under BertAttack.

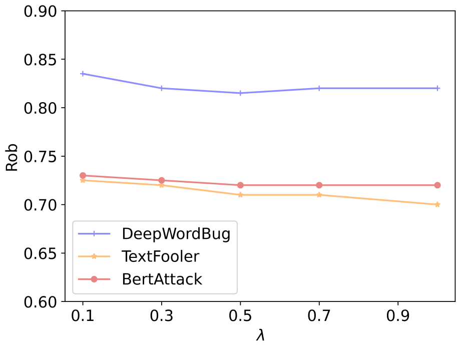

Figure 5: The robustness while performing perturbation with different values of λ of MR-BERT.

Weight of Attention Diversity We observe the effect while setting the weight of attention diversity λ with different values in $\{0.1, 0.3, 0.5, 0.7, 1\}$. Experiment results of MR-BERT are shown in Fig. 5. We find that the Rob decreases when the λ is too large which can not balance the effect of attention diversity in training process. $\lambda = 0.1$ achieves the best performances for most attacks.

5.2. Attention Diversity Visualization

We use *BertViz* (Vig, 2019) to observe the attention matrix. Fig. 6 provides a birds-eye view of attention across all of the sub-models first layer and heads for a randomly sampled input (*"I like this movie, it is really nice!"*). Its rows represent models and columns represent heads. *BertViz* visualizes attention as lines connecting the word being updated (left) with the word being attended to (right) and color intensity reflects the attention weight (darker one means more attention paid).

Sub-models (f^1, f^2, f^3) learned by PAD can keep the general attention patterns similar to that of f with partial changes. The bottom of Fig. 6 shows the attention of head 6. f^1, f^2 and f^3 present diverse attention patterns. The phenomenon is consistent with the learning goal expressed in Eq. 3.2: clean accuracy is kept via maintaining broadly similar attention patterns compared with the base model and minor attention changes diversify the adversarial example distribution of sub-models.

6. Related Work

The ensemble method is originally designed to improve generalization performance (Kuncheva and Whitaker, 2003; Wen et al., 2020; Sinha et al., 2021), which can boost multiple weak classifiers to a strong classifier. With the growing interest of the deep learning community in robustness issues, the possibility of improving the robustness of the model with an ensemble method was recently explored. For image, Pang et al. (2019) try to achieve the ensemble's robustness improvement via promoting the diversity among non-maximal predictions of individual members.

However, the work of He et al. (2017) implies that ensemble of weak defenses is not sufficient to provide a strong defense against adversarial examples. In recent, Yang et al. (2022c) try to establish the relationship between ensemble and gradient diversity for image classifiers based on the model smoothness assumption. For NLP, ensemble method for robustness improvement is still under-explored especially due to that NLP models face various types of attacks. In 2022, Le et al. (2022) modifies and retrains the last layer of a well-trained NN and utilizes a stochastic weighted ensemble of sub-models for prediction. Since the parameters of the feature extraction module are frozen, this method may not be able to eliminate the presence of adversarial samples in the feature extraction step. Our method utilizes the attention diversity to realize a robust ensemble. It can diversify the sub-models with good interpretability.

7. Conclusion

In this paper, we propose an ensemble method PAD which utilizes attention diversity to enhance robustness against different attacks. Compared to previous efforts that focus on enhancing defense ability against a specific type of attack, we provide a novel approach to improve general defense. In order to save memory and computation resources, PAD applies a scheme of adding plugins to the self-attention layer which can dynamically generate multiple sub-models for training and inference. Experiments on three NLP tasks and two target models show that PAD significantly improves robustness under three levels of attacks including five attack methods.

Limitations

The approach to promoting attention diversity in this paper is only applicable to Transformer-based neural networks. For other neural networks like CNNs and LSTMs, our plugin is also applicable, however, it may not be interpretable like attention. Besides, since our ensemble method does not introduce data outside the training set, the sub-models may not be able to generalize well to out-of-distribution adversarial samples.

Ethics Statement

With the emergence of more and more adversarial scenarios, few perturbations, e.g., modifying some words with their synonyms, can mislead a well-trained DNN's prediction. It arises society's concern about the safety and applicability of DNNs in piratical scenarios. PAD has an important significance and role in building a trustworthy AI system. The general defense ability under different levels of attacks indicates that general defense for different adversarial scenarios is promising. The visualization of attention can also assist the society in understanding the inner mechanism of the ensemble. Our work does not arise ethical issues directly and all used datasets are publicly available with no privacy violation.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (NSFC) under grant No.61972384.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei

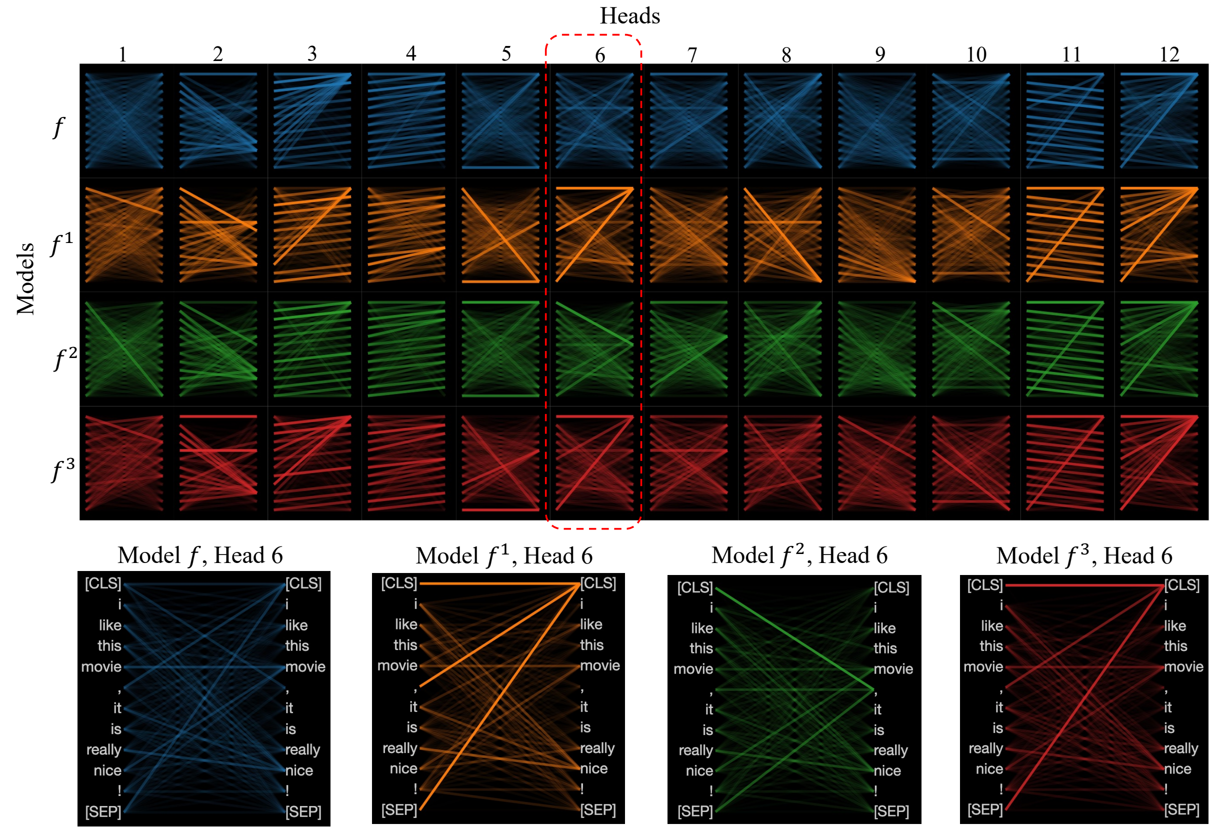

Figure 6: Visualization of attention. Sub-models (f^1, f^2, f^3) in ensemble and the base model (f) present different attention patterns.

Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing EMNLP 18*, pages 2890–2896. Association for Computational Linguistics.

Dennis S Bernstein. 2009. Matrix mathematics. In *Matrix Mathematics*. Princeton university press.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 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*, pages 632–642.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Steffen Eger, Gözde Gül Sahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1634–1647. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 50–56. IEEE Computer Society.

Warren He, James Wei, Xinyun Chen, Nicholas

Carlini, and Dawn Song. 2017. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies, WOOT 2017, Vancouver, BC, Canada, August 14-15, 2017*. USENIX Association.

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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1875–1885. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207.

Thai Le, Noseong Park, and Dongwon Lee. 2022. SHIELD: defending textual neural networks against multiple black-box adversarial attacks with stochastic multi-expert patcher. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6661–6674. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

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 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics.

Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. 2020. A robust adversarial training approach to machine reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8392–8400. AAAI Press.

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. *CoRR*, abs/1907.11692.

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 *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150.

Bo Pang and Lillian Lee. 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*, pages 115–124.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4970–4979. PMLR.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5582–5591. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019a. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1085–1097. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019b. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1085–1097. Association for Computational Linguistics.

Samarth Sinha, Homanga Bharadhwaj, Anirudh Goyal, Hugo Larochelle, Animesh Garg, and Florian Shkurti. 2021. DIBS: diversity inducing information bottleneck in model ensembles. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 9666–9674. AAAI Press.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 13997–14005. AAAI Press.

Yeming Wen, Dustin Tran, and Jimmy Ba. 2020. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuting Yang, Pei Huang, Juan Cao, Jintao Li, Yun Lin, Jin Song Dong, Feifei Ma, and Jian Zhang. 2022a. A prompting-based approach for adversarial example generation and robustness enhancement. *CoRR*, abs/2203.10714.

Yuting Yang, Pei Huang, Feifei Ma, Juan Cao, Meishan Zhang, Jian Zhang, and Jintao Li. 2022b. Quantifying robustness to adversarial word substitutions. *CoRR*, abs/2201.03829.

Zhuolin Yang, Linyi Li, Xiaojun Xu, Bhavya Kailkhura, Tao Xie, and Bo Li. 2022c. On the certified robustness for ensemble models and beyond. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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, ACL 2020, Online, July 5-10, 2020*, pages 6066–6080. Association for Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. Openattack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5482–5492. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4903–4912. Association for Computational Linguistics.